Division of Mathematical and Physical Sciences
MSc in Computer Science: Long Vacation 2008

Report of the examiners

Part 1

A: Statistics

1: Results

	2008	2007	2006	2005	2004	2003	2002	2001	2000	1999	1998	1997	1996	1995
Distinction	16	11	10	5	5	6	5	3	6	7	5	7	5	3
Pass	48	44	40	35	28	25	29	24	31	25	22	28	22	25
Fail	2	8	5	3	6	3	4	5	2	2	0	2	2	2
Candidates	53	53	46	38	34	28	33	29	33	27	22	30	24	27

One candidate has an extension on his dissertation. Two candidates results are incomplete.

2: Number of vivas

There were two vivas.

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 two dissertations.

4: Numbers taking each optional subject

Automata, Logic and Games	0
Advanced Topics in Language Processing	8
Bioinformatics and Computational Biology	2
Categories, Proofs and Processes	1
Compilers	8
Computational Complexity	3
Computational Linguistics	12
Computer Aided Formal Verification	9
Computer Security	28
Computers in Society	8
Concurrency	11
Concurrent & Distributed Programming	17
Databases	19
Executable Biology	2
Formal Program Design	3
Functional Programming	9
Game Semantics	0
Information Retrieval	26
Intelligent Systems I	27
Intelligent Systems II	12
Introduction to Specification	21
Lambda Calculus and Types	2
Machine Learning	22
Object Oriented Design	9
Object Oriented Programming I	17
Object Oriented Programming II	10
Program Analysis	1
Programming Languages	4
Quantum Computer Science	2
Randomised Algorithms	7
Theory of Data and Knowledge Bases	8

B: Changes in examining methods and procedures this year

University Standardised Marks were used for the first time this year. The transition has been smooth and unproblematic.

Turnitin was first used to screen dissertations for evidence of plagiarism last year. The process worked reasonably well overall. In one case, Turnitin did not find any evidence of plagiarism initially when the entire dissertation was fed into it; however, when processing individual chapters separately, it revealed convincing evidence of plagiarism. The Examiners consider the use of Turnitin to have been beneficial in providing assurance of the integrity of the examination.

Holding the vivas and examiners' meeting in week minus one worked much better than in previous years (when it was held in 0^{th} week). The examiners welcome the change to Regulations which will make this the norm.

C: Changes in examining methods and procedures envisaged

To be consistent with the pass mark for course work assignments and the dissertation, the examiners recommend that the pass mark for practicals should be raised from 30 to 50.

Part 2

The overall standard was high, with a good number of distinction candidates and only two failures (from this year's cohort).

Two candidates were found to have plagiarised parts of a question of a take-home assignment. They were awarded no mark for the entire question, in accordance with the Proctors' instruction. In addition a candidate was identified as having plagiarised significant parts of the dissertation. This candidate was referred to the Proctors; this examination remained suspended at the final meeting of the examiners.

The examiners awarded the Hoare Prizes for best overall performance and the best project to Jiewen Huang .

Appendix

Automata, Logic and Games – No candidates took this option

Advanced Topics in Language Processing
Number of candidates: 8 *Mean: 63.1* *Standard Deviation: 9.6*
Questions 1 and 2 were on the whole answered very well, although some candidates showed a tendency to recycle lecture material. A few candidates displayed evidence of considerable reading beyond what was required for the lectures.

On Questions 3 and 4 there were a few good scripts, in which the students showed a good understanding of the course and some evidence of having thought deeply about the questions. The rest contained rather formulaic answers, with a heavy reliance on Wikipedia and the course notes. The tendency to immediately use Wikipedia to answer an essay type question, rather than more authoritative, and original, sources is a little depressing.

Categories, Proofs and Processes
The majority of candidates did very well indeed, with clear understanding and good proofs. All parts of the assignment were completed. There were two weaker candidates: one who showed reasonable understanding, though provided an incomplete submission, and a second who only made a minimal attempt at answering the questions, and showed very little understanding.

Compilers
Number of candidates: 8 *Mean: 65.5* *Standard Deviation: 9.0*
The course was updated compared to previous years, and it was the first time the assessor had set a take-home examination. The paper consisted of four compulsory questions, spread across the material. The students did not appear to find the exam difficult, and were able to answer all questions. The answers were at times unnecessarily long, and there was some evidence (for some bookwork questions) of reliance on textbooks and lecture notes, and (for the question on optimisation) of guessing based on familiarity of the programming language.

Question 1: Students had no difficulty with simple grammar topics such as parse trees, abstract syntax trees, removal of ambiguity, etc. Most were able to discuss the construction of an LL parser and all could show the action of an LR parser on a given string.

Question 2: Bookwork question on attributes was generally not answered well, with too much copying from textbooks. The questions about typing rules were too easy. Syntax-directed translation, understandably, presented the most difficulty; all but two students answered it.

Question 3: This question concerned IR trees, a notation used in practicals, so the material was generally understood, with mistakes due to sloppiness or misunderstanding.

Question 4: This question was answered well by most students, and a few even attempted to do more than was required (to calculate the live variables).

Q1,3 and 4 had very similar averages. Q2 average was slightly lower.

In summary, the exam could have included more challenging and open-ended questions to separate the good students, a change that is intended for next year.

Computational Complexity
Except for two weak submissions, most candidates demonstrated a sound grasp of ideas and techniques studied in the course. The most difficult question seems to have been Question 1 where students where asked to define a reduction to show NP hardness. Here some reductions were flawed whereas others were very nice and interesting. Question 2 was mostly ok, only in Part 3 some students failed to get the point that already a single pointer to the tape consumes too much space. Question 3 was generally well answered. The solutions to Question 4 part 1 showed a high standard whereas in part 2 some students failed to see that the language B needs to be fixed and independent of a particular problem. The optional Question 5 did receive some very good answers.

The presentation of solutions was generally of high, sometimes even of impressive, standard.

Computational Linguistics
Number of candidates: 12 *Mean: 62.5* *Standard Deviation: 9.8*
Q1 was answered at varying levels of detail. Some candidates merely rehashed lecture notes, but some showed evidence of wider reading and were correspondingly rewarded for it.

Q2, with a few exceptions, was answered well, showing that students had benefitted from the lectures and the exercises and developed the necessary skills to tackle this question.

Q3 was again answered at varying levels of detail. Many students seemed not to understand the requirement that the method should be described in sufficient detail that someone could implement it without reference to the paper, and gave sketchy outlines. Others gave pseudo-code at the right level of abstraction.

Computer Aided Formal Verification
Number of candidates: 9 *Mean: 60.9* *Standard Deviation: 7.2*
Overall, this paper was very well done by many candidates.

Q1. Parts 1 and 2 were very easy and virtually all answers were technically correct. There were varying degrees of quality in the explanation provided for 2b. Several candidates got the wrong (polynomial) answer to Part 3, usually by allowing variables to appear more than once in the ordering. Some candidates provided only a superficial investigation of a few special cases. The correct solutions were accompanied by analyses of varying thoroughness and rigour, including some that were excellent.

Q2. Mostly very good, but a handful of candidates had incoherent or incorrect proofs.

Q3. Parts 1 and 2 almost always correct. Part 3 usually correct, but some candidates did not provide "fully detailed" derivations.

Q4. Varying degrees of explanation/derivation and elegance of notation, but virtually all answers very good.

Q5. Many candidates seem to have adapted a complicated proof from Seger and Hazelhurst - or at least taken the same basic approach - but a few noticed that there is a much easier and direct proof. The presentations had varying degrees of quality.

Computer Security
Number of candidates: 28 *Mean: 65.4* *Standard Deviation: 7.8*
There were 28 candidates in all, and most did a respectable job of answering the questions. There were 4 questions of equal weight and these were each marked out of 25.

Question 1 was mainly an "essay" question on issues relating to account and file security, particularly of data on mobile devices. Most people were able to write intelligently about the issues raised.

Question 2 was about protocols. The first two parts were reasonably well done and, unsurprisingly, almost everyone found successful attacks against the quoted protocol. Almost everyone in the third part could avoid the basic authentication attacks, but proving to B that A had been present successfully was rarely done wholly successfully.

Question 3 was about public key cryptography and was largely done well. I found it strange that no-one remarked that in parts b and c the same decrypting power was the inverse of 11, but of course this was not necessary to get the question right.

In Question 4 I was a little disappointed with the answers to parts b and c. For the former, remarkably few people came forward with the obvious remark that the function e(k,.) must clearly be subjective, showing that any 256-bit block can be decrypted under k. To get part c right it was necessary to use an appropriate combination of signing and encryption with enough information to avoid the standard signing-after-encryption (adn vice-versa) attacks. It was also (just about) necessary to include a hash of the message in the signature, and necessary to use something like CBC to split up the message into 256-bit blocks, and to include the sender's key certificate. I do not think anyone did all of these things.

There were a number of candidates who I would describe as very good, but no-one who stood out as truely outstanding.

Computers in Society

Number of candidates: 8 *Mean: 57.9* *Standard Deviation: 5.5*

The Computers in Society assignment was divided into three sections, the first two offering a choice of seven overall, and the last section offering a choice of four case studies. Students were asked to write an essay on one option in each section, and thus three essays in total. In the first section, students had to write an argumentational essay opposing or accepting a contentious statement; in the second, students had to write a descriptive or expository essay setting out the relevant information for a specific topic. The last section required the analysis of a case study.

Students taking this course are challenged by their relative unfamiliarity with writing essays. Students sometimes rely too much on the information available on the Internet or on very superficial research, and do not explore different perspectives sufficiently. Despite this, in most cases students coped adequately and sometimes very well with the essays.

Concurrency

Number of candidates: 11 *Mean: 64.5* *Standard Deviation: 10.2*

Question 1: This question asked students to identify all distinct divergence-free processes with trace set {<>, <a>, <b>, <c>] (there are 19) and to give their stable failures. This question was generally answered well, with most students identifying most of the processes, with 4 finding all of them, and several more missing just one.

Question 2: This question asked students to model and analyse a traffic light system, with sensors to detect cars waiting, and a controller to send commands to the lights. There were several common shortcomings with models. Some models of sensors did not distinguish between the sensors detecting cars and them sending messages to the controllers; others had sensors that seemed to always detect cars. Some controllers kept insufficient information about the state of the lights, so had to repeatedly query them; other controllers perhaps had too much information (I think it's best for the controller to send a single signal telling the lights to change from, e.g., green to amber and then to red, rather than two separate signals). Some controllers were ridiculously complicated (several pages long, whereas the model answer was 5 lines long). The safety analysis (that the lights in one direction turn red at least four time units before the lights in the other direction turn green) was generally done well. The liveness analysis (that once a car is detected, the lights in that direction turn green within 15 time units) was harder and less well done; several students tried to use the trace model, which can't be correct for a liveness condition. The analysis that the model treats time sensibly (i.e. doesn't cause time-stops) was particularly difficult, and few students answered it correctly, Question 3: This question assessed the students knowledge of CSP semantics via some true or false questions. Most students struggle to make the proof in the first part formal, but the latter parts were generally done well.

Question 4: This question tested the students' understanding of algebraic laws for parallel composition and hiding, and the UFP law. A few students didn't understand the step law for hiding properly. A few students produced recursive equations that were not (uniformly) guarded, and so miss-applied UFP. A few students tried to combine too many steps into one, and often made slips as a result.

Question 5: This was deliberately a very hard question, designed to identify truly outstanding students. Slightly over half the students attempted it; one student scored two marks, the rest zero.

Concurrent & Distributed Computing

Number of candidates: 17 *Mean: 58.8* *Standard Deviation: 17.0*

Q1: Binodal distribution of marks – one made around close to perfect answers: very gratifying to see. The other made (around (10-14) disappointingly naïve attempts to avoid answering the question as posed. Some truly poor attempts at solutions showed a misunderstanding of Scola, of CSO, and of these relatively simple problems.

Q2: (a) answered well by….
(b) the best students used an overflowburger from Q1. The least good students invented new forms of buffer without calling them that; and without hearing the call from Q1. overflowburger.

Bad students are bad in a whole host of ways…

(1) They can't program very well: eg they use "spinning" code to detect closing.
(2) They can't understand the idea of a generic component.
(3) They leave spurious debugging code splattered throughout their answers.

Q3: Uniform distribution of marks. The best answers were very good indeed, and dealt appropriately with the variety of traffic that had to be sent on the shaved channel. Part C was done well only by a few. The worst answers were almost as hard to show incorrect as they would be to show correct. Too complicated; too badly presented and explained; not general enough; lack representations of ack messages.

A forceful reminder of the dictum that testing can show the presence of bugs but not their absence.

Databases

Number of candidates: 19 *Mean: 60.5* *Standard Deviation: 5.6*

Students did well on questions that were similar to prior years and also covered in the text (1, 5). But on these two questions I have strong suspicions of collaboration, since several answers that were either wrong or unusual occurred on multiple papers. However, there were no two papers that I saw where the evidence was conclusive. Next year, I would like to get advice from the department on how to deal with this problem, which the markers said was also very prevalent on homework. It would be nice if we could review the homework and MSc exams together for evidence of collaboration.

The students performed quite badly on questions that were either not covered or covered superficially in the text. Clearly, most students did not understand that in the BCNF decomposition algorithm for normalization one needs to first take the closure of the dependencies before projecting; this led to low scores on problems 2 and 3. Many students did not understand how the BCNF decomposition rule applies to dependencies with multiple tables on the right hand side. These two subtleties of BCNF decomposition are not explained clearly in the text, although they were covered in lecture.

Problem 4 was related to a topic (safety of relational calculus formulas) that was covered in lecture but not at all in the text; no student got full credit on this, and it is clear that many do not understand issues of quantification well.

Executable Biology

Question 1: This question tested students' understanding of broad issues in modelling of biological systems based on recommended literature. It was focused on visual modelling formalisms. Answers were detailed about the nature of case studies but the analysis of advantages and disadvantages of such formalisms was rather superficial.

Question 2 & 3: These questions were testing the students' understanding of (i) molecular signalling and its representation in a concurrent modelling language, (ii) the PRISM modelling notation and how it can be used to describe signalling pathways, and (iii) the CSL temporal logic notation for properties, for simplicity all based on the PRISM model checker. Students were asked to adapt/modify existing model files, specify reward structures and properties, and produce plots for certain properties. Answers indicated a sound understanding of modelling issues and an ability to work independently with supporting software tools.

Overall, the results indicated that the students have mastered how to use modelling in a biological context and how to analyse models using modelling and verification tools.

Formal Program Design

All candidates demonstrated a sound grasp of the ideas and techniques studied and how to apply them (except for one who made an empty submission). Question A on weakest preconditions and refinement was generally well answered. The design in Question B(2) was well executed except by those who did not use the invariant suggested by B(1). The designs in Question C(2) ranged from competent to very good although several could profitably have been simplified.

Functional Programming

Number of candidates: 9 *Mean: 65.9* *Standard Deviation: 16.4*

Exercise A – An extended programming question about stacking shelves, and worth 50 marks. Some very good answers, but other solutions were marred by a very poor coding style in Haskell. Difficult to mark and a lot of reliance was placed on the worked example asked for at the end.

Exercise B – An exercise in scans and proofs. There was little scope for candidates to deviate from good programming style, but some managed it.

Exercise C – Straightforward question on evaluating expressions in polish or reverse polish notation. Most good answers.

Game Semantics – No candidates took this option

Information Retrieval

Number of candidates: 26 *Mean: 66.2* *Standard Deviation: 6.1*

There were around 4 very good scripts in total, in which the students showed a good understanding of the course and some evidence of having thought deeply about the questions. The rest contained rather formulaic answers, with a heavy reliance on Wikipedia and the course notes. The tendency to immediately use Wikipedia to answer an essay type question, rather than more authoritative, and original, sources is a little depressing. All were considered good enough to pass.

Intelligent Systems I

Number of candidates: 27 *Mean: 67.2* *Standard Deviation: 8.2*

The assignment was done competently in practically all cases; in some cases much more detail was given than was necessary. The questions with numerical answers were computed well with few errors in the results given, and the candidates were distinguished more by the clarity and quality of the explanations given. Question 1 was a straightforward question on the distinction between planning and searching, and the use of the STRIPS planner; several candidates did not spot that only solutions of optimal length should be found. Question 2 dealt with the filtering of sensor information. The second part could be greatly simplified by spotting that the matrices would always be multiples of the identity matrix, but few saw this. Question 3 dealt with concepts and calculations on a Bayesian network; many marks were lost here by students not making their working clear.

Intelligent Systems II
Number of candidates: 12 *Mean: 70.2* *Standard Deviation: 8.2*
The assignment was well completed by basically all students, with no really weak submission, and some outstanding ones. Question 1 required solid background on inference in first-order logic and logic programming; it was probably the most involved question. The students proved a sound understanding of the inference procedures, in particular resolution, though some of them showed some uncertainties on the role of Herbrand interpretations: in particular, some submissions were a bit awkward on part (a), which required a nontrivial proof, and on part (b), which required a solid knowledge of the notion of correct answer to a logic program. Question 2 was solved correctly by all students, with some minor imprecision. Question 3 required the students to write a small answer-set program; most students provided a good solution; only a few showed uncertainties in the comprehension of weak negation in ASP. Question 4 received generally very good answers, with some students showing very clear understanding of commonsense reasoning in AI.

Introduction to Specification
Number of candidates: 21 *Mean: 57.1* *Standard Deviation: 16.5*
Q1: (raw marks from 27)

This was done adequately (>=14/27) or well (>=20) by all but 7 people. Several people found part (a) a little too hard for them. In retrospect this is not surprising since they saw only a couple of derivations of "derived proof rules", and these were in the classes.

Some people really didn't seem to want to present their proofs in either of the conventional forms they had been shown in the classes and model answers to proofs (namely Sequent trees, and boxed "natural deduction" proofs). I did not penalise people for this, since I had invited proofs in any style they wished. I DID, however, penalise lack of evidence of the proof rules that were used to justify inference steps.

Lambda Calculus & Types
Between MSc in CS and MFoCS there were 5 scripts. Two were of distinction quality, and three worthy of middling passes. It was disappointing that more candidates did not structure the induction in question 2 correctly: it is identical to the corresponding result in the lecture notes. As intended, even the strongest candidates found the last two questions more difficult: only two candidates completed 4a correctly, and only one completed 5b correctly. The assignment seems to have discriminated well between the good and the excellent, as well as testing basic ability.

Machine Learning
Number of candidates: 22 *Mean: 67.0* *Standard Deviation: 6.6*
In general, the students have answered well or very well despite the fact that they have faced elaborate and some difficult questions, which required a deep understanding of the problems. The first question was the easiest one and almost all students have perfectly answered it. The most difficult question was Question 4, with only a few students providing complete answers. The lowest mark (USM) was 55, the highest mark was 97. Most of the students obtained marks between 70 and 90 which is what I would expect given the form of examination (take-home assignment, and assuming that we have reasonable MSc students). In conclusion, I am personally satisfied by the level of understanding reached by the students."

Object Oriented Design
Number of candidates: 9 *Mean: 58.8* *Standard Deviation: 13.7*

Object Oriented Programming I
Number of candidates: 17 *Mean: 61.2* *Standard Deviation: 8.1*
Questions 1 and 2
These were straightforward exam-style questions on specifying and implementing abstract data types. They were done reasonably well by most candidates, but the weakest candidates showed a marked tendency to write random formulas in the hope that they would attract some marks.

Question 3
This was based on one of the practical exercises for the course. I identified seven features of an ideal solution, and marked the question out of 7, with occasional half marks for partially correct features. The resulting marks have been converted to marks out of 25.

Question 4
This more open-ended question was designed to allow better candidates to show their strength. Sadly, there were no really good answers, and all the answers I received were similar (though not to any suspicious extent). All candidates made undisciplined changes to the program, and concentrated on details of the implementation, rather than describing a design in more abstract terms. No candidate succeeded in introducing any new class that had significant methods or represented a significant abstraction. No candidate gave any discussion of invariant relationships that would be maintained in the program, or of the protocols for executing commands or updating the display in the extended editor.

Some candidates have done significant implementation work for this question, but that is not what was asked for, and I have not given any extra credit for it. I have read all the essays carefully, but cannot distinguish between them, and so I have given each candidate a mark of 12/25 for the question.

Object Oriented Programming II
Number of candidates: 10 *Mean: 68.1* *Standard Deviation: 10.8*
Q1: A range of qualities of answers, including a few really good ones.

Q2: Gratifyingly well done by everyone, and brilliantly well done by 3 candidates. Several people took the opportunity for misunderstanding offered in b(i) by my bad phrasing of the question, but still got conditionals correct.

A few people used visitor rather than active abstract syntax – but it seemed churlish to penalise them for that since the invitation to use AAS was low key, and almost hidden beneath fig 2 (and it's much more tedious to do it that way).

Program Analysis
All candidates achieved an acceptable standard, with USMs of 59 and above.

The assignment consisted of three questions. The first one, on closure analysis and strictness analysis for lambda calculus, was very well answered by all three candidates. The second question on dataflow analysis was well-answered in the first, straightforward part, but the application of Park's theorem proved too hard. The third question on type inference got good answers, but none of the candidates was able to generalise the algorithm to deal with multiple inheritance.

Programming Languages

Quantum Computer Science
The results this year were substantially better than the previous years. I attribute this to the increase in the number of lectures I gave, namely from 16 to 24. I also covered less material and gave more examples. This seems to have made the material far more accessible. Next year I will need to put up some more challenging exercises to obtain a higher resolution for identifying outstanding students. But it is pleasing to see that a substantial group were able to correctly solve a problem which was actually a research result from 2007 (Question 3).

Randomised Algorithms
Number of candidates: 7 *Mean: 60.8* *Standard Deviation: 10.1*

Theory of Data and Knowledge Bases
Number of candidates: 8 *Mean: 65.8* *Standard Deviation: 10.0*
The candidates generally demonstrated a good understanding of the ideas and techniques studied in the course. The questions about hypergraph acyclicity and about Ehrenfeucht-Fraisse games were excellently answered by most of the students. The questions involving formal proofs were answered well by about half

of the students only. This shows that students lacking a background in Math or Theoretical Computer Science have a hard time with formal proofs.

Generally, students coming from more Engineering oriented undergraduate studies have a hard time with my course and need to make a huge effort to lift themselves to the level of abstraction required to understand certain concepts of logic and complexity theory that are fundamental to this course. I expect this to improve next year, when a MSc-level course on fundamentals of TCS will be taught (by M. Benedikt). However, I was pleased to see that most students actually tried their best to catch up with the theoretical foundations.

The presentation of solutions was generally of high, sometimes even of impressive, standard.

List of successful dissertations

- Implementing and Analysing Cryptographic Digests
- Information Retrieval
- An Investigation into Machine Learning for Image Steganalysis
- Multiple Ranked Outputs for A Natural Language Parser
- Tracking Objects in 3D Using Videos from Multiple Cameras
- Blog Quality Analysis
- Robot Sheepdog
- Object-Oriented Splines Library
- Vector-Space Models of Semantic Similarity
- Data Extraction Using Semantic Annotation
- Artificial Intelligence
- Models of Grammar Evolution: Evolving English
- Distributed, Functional, High-Level 3D
- Validating Uncertain XML Data
- Fusion Haskell Unicode Strings
- Scalable Confidence Computation in Probabilistic Databases
- Working with Free Theorems
- Machine Learning techniques for Astrophysical Modelling
- Bilingual Corpus Exploratory Workbench
- Virtual Birds
- Turning FDP Debugger Traces into Sequence Diagrams
- Practical Implementation of ECC on Lightweight Devices
- Building Support Vector Machine for Alternative Splicing
- Spatial Domain Syepanalysis using Statistical Learning Techniques
- Modelling with Prism (Intelligent Systems or HIP Protocol)
- Source-level Control-flow Analysis for Java
- Tracking Objects in 3D Space in Real-Time
- Trusted Infrastructure for the Campus Grid
- Machine Learning Algorithms Applied to Financial Forecasting
- Querying Data Under Access Limitations
- Simplifying Web Application Programming
- Querying Blogs
- Vector-space Models of Semantic Similarity Applied to the Output of a Question Answering System
- Web Data Extraction Tools and Applications
- Pervasive and Ubiquitous Computing
- Quantitative Funds
- Swarm Intelligence for Vehicle Routing Problems
- Querying Data Under Access Limitations
- Pervasive Computing
- Inductive Chart Parsing
- Efficient Algorithms for Verifying Timed Automata Research Proposal

- Multi-Agent Based Network Systems
- Building a Resolution-Based Ontology Reasoner
- 3D Reconstruction of Kidney Cancer Images
- Human Centered Computer Security
- Free Software: Ethical Issues Behind Reliability
- Modules for AspectJ