Division of Mathematical, Physical and Life Sciences
MSc in Computer Science 2021

Report of the examiners

Part I

A: Statistics

1. Numbers and percentages in each class (Merit grade was introduced in the 2018-19 academic year):

Year	Distinction	Merit	Pass	Fail	(Total)
2021	22 (49%)	10 (22%)	12 (27%)	1 (2%)	45
2020	35 (60%)	11 (19%)	3 (5%)	1 (2%)	58
2019	20 (45%)	9 (20%)	14 (32%)	1 (2%)	44
2018	24 (41%)		31 (53%)	1 (2%)	59
2017	18 (38%)		26 (54%)	0	48
2016	22 (39%)		32 (57%)	2(4%)	56
2015	19 (37%)		32 (62%)	1 (2%)	52
2014	14 (22%)		46 (73%)	3 (5%)	63

2. Number of vivas

There were no viva voce examinations, with all candidates dispensed from this requirement.

3. Marking of scripts

All candidates took a wide variety of options.

Computer Science courses that were examined as take-home assignments were completed during the vacation succeeding the term in which the course was taught, with the exception of Requirements which was completed as a take-home assignment during Trinity term.

Databases and Concurrent Programming were examined by a 3-hour written paper in Hilary Term.

Artificial Intelligence, Bayesian Statistical Probabilistic Programming, Computational Complexity, Computational Game Theory, Computer Security, Concurrency, Concurrent Programming, Knowledge Representation and Reasoning, Machine Learning, Principles of Programming Languages, and Probabilistic Model Checking were examined by a 3-hour open-book exam in Trinity Term.

The scripts for Law and Computer Science, Computers in Society, and Requirements were double-marked. All remaining scripts for Computer Science were single marked according the agreed mark scheme.

Due to the unusual online, open-book format of the papers it was not possible to conduct the usual checks on examination scripts to ensure that all material had been seen by the examiners, or to double-check marks on scripts against recorded marks. However, assessors were given additional time to check that they had recorded their marks correctly on the results spreadsheets.

A per-paper data-entry check was performed after the recorded marks were transferred to the database programs used for processing marks.

All papers were reviewed in the Examiners' preliminary meeting. For each paper, USMs were produced by applying a piecewise-linear mapping to the total marks, with the mapping chosen to produce a mean USM of approximately 65 for all candidates taking that paper. For papers that were already close to that distribution, or with a very small number of candidates, the raw marks were used. The examiners additionally checked that the class borderlines for each paper fell in reasonable places, but did not further adjust the marks as a result. In considering the classification of each candidate whose mean USM fell close to a borderline, the examiners verified that the candidate's mark had not been unduly affected by the scaling applied.

4. Dissertations

The project dissertation is compulsory for candidates in the MSc Computer Science. All project dissertations were double marked, by one internal examiner and by one assessor independently, using a revised set of qualitative descriptors. Both markers agreed a mark between them.
Examiners received a report from the project supervisor. Where the project had involved the production of software, supervisors were expected to see a demonstration of the software in action and to comment on it in their reports.

5. Practicals

Practicals for all courses were assessed together as a single paper but did not contribute to the classification. The same method for assigning credit for practical work was used as for other examinations conducted in the Department of Computer Science.

B: New Examining Methods and Procedures

Exceptionally, all exam were held in open-book format. The examination duration was increased by 30 minutes to allow time for downloading, uploading and technical issues. Assessors marked pdf scripts and entered marks onto a spreadsheet for uploading to the database programme.

The viva component of Advanced Topics in Machine Learning was held on Teams.

C: Suggested Changes in Examining Methods and Procedures

The examiners recommend to return to handwritten or electronic closed-book exams instead of open-book. The Inspera system created numerous difficulties for academics and admin staff and the examiners do not recommend its use in its current form.

D: Examination Conventions

The Examination Conventions were uploaded onto the departmental website. This is explained in the Course Handbook. In addition, the candidates received a notice to make them aware of this fact. Updated conventions were published with information on the safety net.

The Course Handbook can be found at http://www.cs.ox.ac.uk/teaching/handbooks.html.

Examination Conventions and Notices to Candidates can be found at: http://www.cs.ox.ac.uk/teaching/examinations/.

Part II

A: General Comments on the Examination

University Standardised Marks (USMs) were used. A marking scheme indicating the approximate number of marks allocated to each part of a question was included on the question paper. Markers could, however, be flexible in their interpretation of this scheme in the light of the responses to the question. A limit was imposed on the number of questions that a candidate could attempt. The details are described in the Notices to Candidates which can be found here: http://www.cs.ox.ac.uk/teaching/examinations/

All practical exercises were assessed together as a single paper. The same method for assigning credit for practical work, based on the mark scheme used for practicals within the Department of Computer Science, was used to assign marks.

The examiners agreed that late submission in open book exams should not be penalized, because the system of submitting through a service request form made it impossible to know when the students had finished working on their script, and when they would have submitted their responses had they been able to submit within the system. Moreover, all students who had submitted late had submitted an MCE, in most cases the late submission was within 5 minutes, and the examiners had no way of knowing if their claims to have had technical difficulties were true or not. In addition, the University's penalty table only allowed them to either impose no penalty or fail the script, which seemed inappropriately harsh.

B: Equality and Diversity Issues and Breakdown of the Results by Gender

The number of female candidates was so low that the breakdown of results by gender will be omitted to maintain confidentiality.

C: Detailed Numbers on Candidates' Performance in each Part of the Examination

MSc Candidates

Paper AI	Raw	USM
No. of attempts	11	11
Average mark	51	51
Standard deviation	12.39	12.39

Paper ALG	Raw	USM
No. of attempts	4	4
Average mark	87.5	85.83
Standard deviation	16.39	17.42

Paper ASEC	Q1	Q2	Q3	Q4
No. of attempts	4	4	4	4
Average mark	18.25	19.5	16.75	15
Standard deviation	3.34	2.69	1.48	1.87

Paper ATML	Raw	USM
No. of attempts	33	33

Average mark	77.54	77.54
Standard deviation	5.61	5.61

Paper BSPP	Raw	USM
No. of attempts	14	14
Average mark	58.71	61.45
Standard deviation	12.13	14.37

Paper CADS	Raw	USM
No. of attempts	7	7
Average mark	67.57	67.57
Standard deviation	17.2	17.2

Paper CAFV	Raw	USM
No. of attempts	6	6
Average mark	74.08	74.08
Standard deviation	8.73	8.73

Paper CGT	Raw	USM
No. of attempts	22	22
Average mark	84.55	77.57
Standard deviation	11.56	11.82

Paper CIS	Raw	USM
No. of attempts	19	19
Average mark	63.53	63.53
Standard deviation	7.07	7.07

Paper CLT	Q1	Q2	Q3	Q4
No. of attempts	8	8	8	8
Average mark	19	15.25	14.88	17.38
Standard deviation	6.06	2.33	5.35	7.65

Paper CONC	Raw	USM
No. of attempts	7	7
Average mark	55.86	55.86
Standard deviation	21.19	21.19

Paper CP	Q1	Q2	Q3	Q4
No. of attempts	3	3	3	3
Average mark	18.33	11.83	13.5	10
Standard deviation	1.93	1.31	3.74	5.52

Paper CPLX	Raw	USM
No. of attempts	5	5
Average mark	50.8	50.8
Standard deviation	27.35	27.35

Paper CPP	Raw	USM
No. of attempts	5	5
Average mark	87	87
Standard deviation	3.63	3.63

Paper CQM	Raw	USM
No. of attempts	3	3
Average mark	95	93.54
Standard deviation	7.07	9.13

Paper CSEC	Q1	Q2	Q3	Q4
No. of attempts	3	3	3	3
Average mark	15.83	12	11.67	14
Standard deviation	1.65	4.9	4.5	4.32

Paper DB	Raw	USM
No. of attempts	10	10
Average mark	64.4	67.95
Standard deviation	15.32	14.99

Paper DBSI	Raw	USM
No. of attempts	6	6
Average mark	75.5	75.5
Standard deviation	7.68	7.68

Paper FOCS	Raw	USM
No. of attempts	8	8
Average mark	59.88	62.6
Standard deviation	26.56	26.74

Paper FP	Raw	USM
No. of attempts	8	8
Average mark	54.62	57.44
Standard deviation	16.34	16.02

Paper KRR	Raw	USM
No. of attempts	6	6
Average mark	50.5	58.61
Standard deviation	13.02	10.99

Paper LCS	Raw	USM
No. of attempts	8	8
Average mark	66.25	66.46
Standard deviation	3.38	3.7

Paper LCT	Raw	USM
No. of attempts	4	4
Average mark	74.25	74.25

Standard deviation	9.96	9.96

Paper ML	Raw	USM
No. of attempts	29	29
Average mark	61.86	61.86
Standard deviation	16.92	16.92

Paper PAC	Raw	USM
No. of attempts	15	15
Average mark	81.53	81.53
Standard deviation	15.89	15.89

Paper PMC	Raw	USM
No. of attempts	12	12
Average mark	73.75	71.76
Standard deviation	16.47	15.83

Paper PPL	Raw	USM
No. of attempts	3	3
Average mark	49	49
Standard deviation	23.15	23.15

Paper Prj	USM
No. of attempts	44
Average mark	72.77
Standard deviation	8.77

Paper QPC	Raw	USM
No. of attempts	16	16
Average mark	79.25	76.11
Standard deviation	16.9	16.74

Paper REQ	EXAM	GRP	Total	USM
No. of attempts	17	17	34	17
Average mark	68.53	8.38	76.91	73.07
Standard deviation	5.82	2.32	6.62	6.29

Paper SC8	Raw	USM
No. of attempts	15	15
Average mark	75.93	75.8
Standard deviation	4.97	4.92

D: Comments on Papers and individual Questions

Advanced Topics in Machine Learning

This course is organised in two themes, and this year's themes are relational representation learning and Bayesian machine learning, and both of the themes have attracted significant interest. The evaluation of this course is based on a reproducibility challenge, where the students form teams to reproduce the results of a machine learning research paper assigned to them based on their preferences. Overall, 16 teams are formed, exactly 8 teams for each of the two themes. The overall performance was very good. There was no significant difference between the average marks for each theme, and so no team/student was at disadvantage due to their choice of theme. Most of the teams met the reproducibility challenge despite the limited computational resources they had access to. Almost all teams went well beyond the assignment paper, by including additional experiments to gain and deliver further insights.

The examiners agreed not to scale this paper.

Advanced Security

Unfortunately, students were not provided with the exam paper written for HT2021, and instead provided with the exam that had been set last year (2020). After this was discovered during the marking process, the examiners did the following: notify academic administrators, checked whether any exam questions were present in this year's problem sheets (they were not), and whether the tutors gave answers of the past exam papers verbally or discussed these with the students (we believe this to not be the case). We asked the administrative team to assess the scripts for evidence of any plagiarism with previous exams (and especially 2020 exam papers) using TurnItIn, and they have advised that there are no issues to report.

Question 1 was concerned with attack vectors, the vulnerabilities they exploit and the harms that can arise. In part (a) students had to consider two specific vulnerabilities associated with the Stuxnet attack, explain their nature, and how they differ from each other in terms of exploitability and impact. They then had to consider the CVSS scores associated with each, how they differ and consider whether they agreed with the scores. In general, this question was answered well, however all students failed to comment on the availability of Metasploit modules in the context of exploitability (which we will need to consider enhancing the discussion of in next year's lectures). No students managed to obtain full marks in part (a), however 13 students managed to drop only a single mark. The standard of answers was high.

In part (b) students had to research the 2015 attacks on the Ukrainian power grid and produce an attack tree or graph describing the attack, including elements which described initial system compromise, lateral movement and at least one example of a human-vulnerability being exploited. The tree had to be explained. Then a further reflection on the nature of the cyber-harms resulting was required, and further details of two specific harms provided and ideas on how to measure the harm proposed. Answers to part (b) were more variable, with only 4 students achieved full marks.

Question 2 was concerned with attack detection and comprised of two parts. Part (a) required students to consider human vulnerabilities in systems, providing examples and explanation of example types of attack surface associated with human vulnerability, and then considering how appropriate an anomaly-based detection system might be in detecting attacks which use such vectors. 4 students obtained full-marks and there was a wide variety of answers given. In general, students appeared to have understood what a human vulnerability is, but not all were able to clearly explain the issues, nor how anomaly-based threat detection should be applied. There were some excellent answers however. In part (b) students were required to design threat-detection rules for use in the Ukrainian power grid attack (from Q1). Each had to be fully explained, and then for three examples a discussion of the potential for false-positives and false-negatives

provided. 6 students achieved full-marks. All students managed to present an answer addressing each aspect, but the quality of answers was variable with some rules being simple (but valid) where others had chosen to challenge themselves further. In general, the quality of the consideration of false-positives and false-negatives mirrored the quality of the rules designed.
Question 3 was concerned with achieving situational awareness at a small logistics company. Each student took different approaches to the dependency modelling aspect of the assignment, most of which were a variant of graph-based modelling or BPMN modelling. All submissions were of good quality.

Question 4 was concerned with attacks and response decisions. The students had to create attack graphs, design visualizations and make recommendations for type of SOC to make use of. It was good to see a variety of answers here. One student did not quite understand the visualization design exercise, and instead focused on stating the requirements for such a design. Overall though, all students here performed really well again.

From a marking perspective, we saw no evidence of students copying material from past exams or online resources, and no evidence of collusion (students answers were suitably distinct from each other). We regard this year's students as a strong cohort, so these results are not surprising.

Overall, student average was high at a 76% average with a st.dev of 8.9%, which is regarded as high, but not cause for suspicion. From a marking perspective, the assessors saw no evidence of students copying material from past exams or online resources, and no evidence of collusion (students answers were suitably distinct from each other). This year's students as a strong cohort, so these results are not surprising. Interestingly, the Part C students outperformed the MSc students in Q1 and Q2.

The Board agreed that the only fair outcome was to allow the marks to stand, and that this was in accord with the advice from the Proctors.

Artificial Intelligence

Generally, the marks in this paper where quite low. The highest mark obtained was only 69, and most students scored within in the range 50-65. A significant proportion of the students scored below 50. From these results, it may appear that the paper was probably too hard. However, the Assessor reported that that this was not the case, and that the paper was of a suitable level of difficulty.

The Board followed the Assessors’ recommendation and did not apply any scaling.

Automata Logic and Games

In Question 1, all candidates provided a correct construction of a Büchi automaton with O(nm) states which is equivalent to an initial Rabin automaton. Almost everybody, apart from a few exceptions, gave a detailed proof of the equivalence of the Büchi and Rabin automata.

In Question 2(a), many candidates ignored the deadlock states in an LTS. For example, many gave the following answer to (a): it is the set of states s from which on any path ¬ P holds until P∩[a]A. The correct answer is the set of states s from which on any path ¬ P holds until P∩[a]A or a deadlock is reached. To obtain full marks, the candidates were expected to take into account the deadlock states. Part 2(c) was solved correctly by most students, and parts 2(b) and 2(d) usually contained very detailed proofs.

Some students used game graphs to solve Question 2(e). However, this requires more computation, and a more succinct solution would be using approximations.

To prove determinacy in Question 3(a), several candidates tried to adapt the proof of the determinacy of standard parity games from the lecture note. In doing so, many made some incorrect statements. A typical mistake was to claim that (mW_R)‾ is a trap for R or that G-mW_R is a subgame, where mW_R is the maximal set of vertices from which R has a memoryless winning strategy. This is true for standard parity games but not for simple ones. It was enough to notice that determinacy follows from the correctness of the algorithm that computes the winning regions W_V and W_R.

When describing the algorithm for computing W_V and W_R, several students used an algorithm from the lecture notes for standard parity games. Such an answer was marked as incorrect because simple parity games have different winning conditions, and hence different winning regions, than standard parity games. The goal of this question was to find a polynomial time algorithm for computing W_V and W_R, which was successfully done by the majority of the candidates.

The assessor recommended scaling the paper, as the average was well over 80. The examiners decided to map 100-100, 80-72, and 50-50 in a piecewise linear function, which brought the average down to 79.2, with a standard deviation of 19.8.

Bayesian Statistical Probabilistic Programming

Question 1.
This is a question about Markov chain Monte Carlo and the Metropolis-Hastings algorithm. Parts (a) and (b) are straightforward: they compare the Metropolis-Hastings algorithm with rejection sampling, from the perspective of estimating an expectation, and relate the M-H sampler to the Simple Monte Carlo estimator. Parts (c) and (d) test the candidate's understanding of detailed balance. Most students answered part (c) correctly, but some failed to construct counterexamples. Parts (e) and (f) are about using the M-H algorithm to simulate the Poisson distribution, and proving that the simulation is correct. (In the lectures, the candidates have seen a simulation of a discrete distribution using the M-H algorithm, and a correctness proof.) For part

(f), there are various ways to prove that the induced Markov chain has Poisson distribution as its stationary distribution. The intended method is direct calculation from the definition of stationarity, which the question should perhaps have specified. It is, of course, possible to---and many candidates did---argue that the detailed balance guarantees stationarity, and Markov chains induced by the M-H algorithm satisfy detailed balance.

Question 2.

This is a question about variational inference. Parts (a) and (b) are a straightforward test of the candidate's understanding of the basics, including ELBO and KL-divergence. For part (c), the candidates have seen a similar (but somewhat simpler) derivation of the gradient of ELBO. Part (d): score function and reparameterisable gradient estimators were discussed at length in the lectures. Some candidates failed to notice that the second and third summands in the RHS of the equation of part (c) constitute the reparametrisable gradient estimator.

Question 3.

This question is about using probabilistic programming to construct probabilistic models. It tests the candidate's understanding of the meanings of probabilistic programs, and their ability to express them using Lebesgue integration. Candidates were free to use any probabilistic programming language (including pseudocode), and most explained their programming intentions using comments. Example probabilistic programs were covered in lectures but this particular scenario (probit regression) was not considered. Parts (a) and (b) were generally done well: most candidates were able to construct programs containing the correct components (priors on mean and variance, score for each late and early observation, sampling for 5 days, and checking whether Alice was late on all 5 days), and worked out the probability space, outcome space and observations. For part (c), some candidates already constructed low-dimensional programs as their answer to part (a). Part (d) was the most difficult part; and part (e) the easiest: most candidates scored all 3 marks for (e).

The average was low at 62.6. The board followed the assessor’s recommendation to scale up by mapping 50-50, 66-70, and 80-88. This resulted in an average of 66.

Computational Complexity

Generally, the exam was done very well, with students showing proper understanding of concepts introduced during the course. q.1: As expected, part (a) was done well (best done by building up solution from empty set than by eliminating undesired vertices from entire set). The NP-completeness in part (b) also done well, errors mostly just in details. In part (c), incorrect conclusions often made. Part (d) mostly done well, but often solutions went wrong in details. q.2, reasonably well done by most students. As expected, part (e) was hardest, most answers not using the padding argument that is needed (although most got partial credit). For (b) most solutions had good conceptual idea but didn't present it well. 2(d) - not many made the "random restriction" argument that is needed. q.3 slightly easier than the others, in marking it I was more fussy about precision and presentation. Part (a) was fairly easy. q.3(b) mostly right general idea but mostly without enough detail on selecting uniformly at random from n items using a fair coin, with sufficiently low failure probability. (c): some but not most got that when you can return to s, you may still have inverse-exponential probability to find t.

The average of this paper was a little high at 71.8. The assessor judged that the students had performed well and shown good understanding. The examiners discussed scaling and agreed on mapping 47(raw marks) - 85(USM), 40-70, 30-60. The resulting average was 66.3.

Computer Security

The paper was only attempted by 3 MSc students. Overall, the MSc candidates appeared substantially weaker than the undergraduates, and this could not wholly be attributed to the difficulty of Question 4 (the additional question they were required to attempt) – although the extra time pressure of having to answer all four questions was almost certainly a contributory factor. Given that there were only 3 candidates, the Board concluded that no scaling should be applied.

Concurrency

Overall the paper proved to be a good test.

Q1. This was a question on modelling and state explosion. As the least standard question is was also the least popular, though those who tried it did quite well. The only real disappointment was that a good proportion of people believed that a process that holds M distinct values from a type of size M has about N.M states rather than N^M.

Q2. The first part of this question was about deadlock and UFP. Given the number of marks available in (B), candidates were expected to justify the required properties of the Pi and Qi, except what was to be proved in (c). To get full marks in (c), it was better to define a vector of N processes based on the Qi than to use a lot of …..s. The second part, on reasoning about infinite behaviour was done quite well but proved a good test.

Q3. The first half of this question was about determinism. Almost everyone correctly asserted that the throw operator preserves determinism. To get full marks, candidates had to understand and demonstrate that each trace is attributable in only one way to P and Q. (The failure of this is the explanation of why ||| fails to preserve determinism.)

The second half, on trace specification, was done well.

The examiners were happy with this paper’s average of 68.1 and the distribution of marks and decided not apply any scaling.

Concurrent Programming

The overall standard in the Concurrent Programming exam was extremely good.

Question 1 concerned the implementation of a lock-based elimination stack, a concept that students hadn't seen before. Part (a) asked students to implement a lock using a monitor: quite a lot of students used "notifyAll", when "notify" would have been more efficient; some students used a CSO monitor, when a JVM monitor would have been more efficient; a few students had unnecessary signals from the "acquire" method to the "release" method, which seemed very odd. Part (b) asked students to implement a concurrent stack using a lock for protection: most answers were good, but a few tried to return before releasing the lock. Part (c) asked students why throughput in a system using this stack would not scale well: most answers were reasonable, if a little vague. Part (d) asked students to add a "tryLock" to the lock: a surprising number of students forgot to use a "synchronized" block, even they had done so in part (a). Part (e) asked students to consider overlapping push(x) and pop invocations, and to explain why, for linearization, it is sound for the pop to return x; again, most answers were rather vague; must saw that the linearization point for the push could be taken to be immediately before that of pop; but most failed to consider the effect on other operations. Part (f) asked students to implement an exchanger, to allow threads to exchange values, or to timeout of no other thread was willing to exchange. This part proved difficult, as expected. Most attempted solutions would go wrong if a third thread tried to exchange at an inopportune moment, leading to two threads receiving the same value and/or a value getting lost; in particular, a surprising number of students thought that an existing thread that receives a signal would have priority over that third thread, even though lectures clearly stated that the threads compete for the lock. Another common (rather surprising) mistake was for students to place the timed wait inside a while loop, with the effect that the timeout was disabled. Part (g) asked students to describe how to use these ideas to give better throughput. Most students saw the idea of trying to get push and pop operations to exchange via an exchanger, although some explanations were rather vague. Not all students saw the possibility of two pushes (or two pops) exchanging (in which case it's necessary to re-try). Few students spotted that using multiple exchangers would improve throughput further.

Question 2 concerned implementing an object like a monitor using other concurrency primitives. Part (a) asked students to implement a monitor by encapsulating a server process. This was mostly well done. Some students neglected to capture that when a thread performs an "await", it releases the lock. Some students used more channel communications than necessary, for example, using two channel communications to tell a waiting thread that it has received a signal and obtained the lock, when a single communication would be preferable. Part (b) asked students to implement the monitor using semaphores. This was mostly done well. A fairly common error allowed signals to be lost, if a thread called "signal" twice, with the second call before the first signal was received. Part (c) asked students to discuss fairness of their implementation from part (a). This was again mostly well done, although some answers were a little vague. A fairly common mistake was to confuse fairness of a "serve" construct (i.e. fairness between different branches, which is provided by the standard implementation), with fairness of a channel within a serve (i.e. fairness between different threads trying to send or receive on the channel, which is not provided).

Question 3 concerned a mechanism to allow three threads of different types to synchronise and exchange identities. Part (a) asked students to implement this mechanism by encapsulating a server process. As is often the case with such examples, many students over-complicated the problem: the simplest solution has the server receive from three clients in a fixed order, and then to send to them in a fixed order. A surprising number of students used OneOne channels, when ManyOne and OneMany channels were needed. Some students used more communications per round than were necessary. Part (b) asked students to implement the mechanism using semaphores. Most students completed this very well, with a well thought-out signalling strategy. Some students failed to achieve mutual exclusion: in some cases, this was fairly sloppy; in other cases, it was rather subtle, allowing a thread in one role to start the following round before threads in other roles had completed the previous round. Some students used rather more signals than necessary: it is possible to solve the problem with five signals per round; some students used as many as fifteen per round; this was lightly penalised. Part (c) asked students to implement the mechanism using a monitor. This part proved harder, although some students again produced very good answers. The best answers again had a clear signalling strategy: a regimented scheme tended to work best. A common error failed to block threads from one round of the exchange before the previous round had progressed sufficiently. Some students used a JVM monitor (as opposed to a CSO monitor), which worked less well, because being able to target signals is beneficial.

Despite the concurrent programming aspects of the exam being done very well, some students' understanding of standard sequential aspects of Scala was poor. Common mistakes were not understanding basic scoping rules, or not understanding the correct way to return results from functions. Added to this were the perennial shortcomings of poor commenting and failure to keep private variables private. The overall standard of basic programming seemed worse than in recent years. It is possible that this is because students have had fewer practicals than normal this year, because of Covid, and so have done less programming.

The assessor commented that the overall standard was extremely good, but recommended to scale this paper to achieve a better distribution of marks; the examiners agreed that the marks were a little high at the top end and followed the recommendation. They mapped 50(raw mark)-90(USM), and 8-30, which resulted in an average of 70.8.

Computational Game Theory

A total of 51 students sat the exam, of which 22 MSc students and 19 fourth year students in Part C. Generally, the students did extremely well, with aggregate marks ranging from 57 to 98 (out of a maximum of 100).

Question 1a (15 marks) This question tested the candidates' mastery of the von Neumann-Morgenstern axioms when comparing lotteries in a preference order. Gennerally, the students did very well. A common mistake was not mention some of the axioms used, in particular, the equivalence axiom. I adopted the following guidelines for common mistakes: – not mentioning first axiom: −3 marks – not mentioning second axiom: −2 marks – insignificant computation error: −1 mark – wrong conclusion: −2 marks Apart from using the axioms directly, the question can also be addressed by invoking the von Neumann-Morgenstern Theorem and then reason about expected utility. Full marks were given for such a solution if correct and if it was mentioned that all axioms are used in the von Neumann Morgenstern Theorem.

Question 1b (10 marks) This question was about invoking the von Neumann-Morgenstern Theorem and reasoning about expected utility. The question can also be answered correctly by invoking the independence theorem directly. Both solutions could give full marks if correct. All students drew the right conclusion that the preferences over `1, `2, `3, and `4 as stated cannot satisfy the von Neumann-Morgenstern axioms.

Question 2 Question 2 was about zero sum-games and the mini-max theorem, and appeared to present no major problems to the students.

Question 2a (5 marks) The candidates were expected to write down four equations. Virtually all of them did so impeccably.

Question 2b (5 marks) The candidates were expected to draw a graph on basis of their solution for 2a. Virtually all students did so correctly. Where some annotations, which were not asked for, were missing, these were usually provided in part 2d.

Question 2c (5 marks) As for Question 2b. 3

Question 2d (10 marks) The students were expected to give an analysis of the game that was the subject of this question and transfer its findings to zero-sum games in general. In particular, they were expected, but not explicitly asked to, to make a connection with the minimax theorem by von Neumann, which was dealt with in the lectures. Most students did very well giving an analysis of the Nash equilibria in this game, where the indifference principle was regularly invoked. The indifference principle only applies indirectly when inspecting the graphs in parts b and c, and a couple of marks were deducted if that was not spotted. The students were also expected to spot the connection with the minimax theorem, and some marks were deducted if they did not.

Question 3

Question 3a (15 marks) In Question 3a, the students were expected to invoke the Nash Folk Theorem for machine strategies, compute the "security values" (minmax values) in a given strategic form stage game, and determine which of its pure strategy outcomes are sustained by a Nash equilibrium in the infinitely repeated game. The students were moreover asked to design machine strategies for the equilibria thus found. The most straightforward way to do the latter was by providing so-called GRIM strategies. Again the students generally did very well. Virtually all followed the reasoning underlying the Nash Folk Theorem and the security values. Yet not all mentioned the Nash Folk Theorem by name. The identification of the pure strategy profiles sustained by a Nash equilibrium in the repeated game on basis if the Nash Folk Theorems also presented no problems. In some cases, students attempted more complex machine strategies, which in some cases also worked but in some they did not (this was generally relatively hard to check). GRIM strategies, however, were a clear favourite when answering this question. I adopted the following as a rough marking scheme: – Nash Folk Theorem + security value argument: maximally 6 marks – Correctly identifying the three "pure" equilibria: maximally 3 marks – Maximally 2 marks for each correct machine strategy.

Question 3b (10 marks) In this question, the candidates were asked to make a connection between a once repeated strategic form game and an extensive form game. More specifically, they were expected the former as the latter. Apart from Question 4a, this question arguably presented the students with most trouble. Some students complained in their solutions that the question was not clear as to how the repetition of the stage game should be understood, but I cannot agree. The students were expected to give a game tree, identify the information sets, compute the number of strategies available to each player, give an example of a strategy profile and compute its outcome, and comment on the difference mixed and behavioural strategies in this game, mentioning perfect recall and Kuhn's theorem. Drawing the game tree generally did not present major problems, although sometimes the payoffs were omitted. When providing and 4 identifying the information sets and counting the number of strategies available, the students struggled considerably more, which is surprising as this is extensively dealt with in the tutorial classes and problem sheets. For the question on behavioural strategies the students' answers displayed a greater variance. I adopted the following as a rough marking scheme: – correct game tree: maximally 3 marks – information sets: maximally 1 mark – correct number of strategies: maximally 2 marks – correct strategy profile: maximally 1 mark – behavioural strategies mentioning Kuhn's Theorem and perfect recall: maximally 3 marks.

Question 4

4a (10 marks) In the lectures and on the exercise sheets, coalitional games are presented in a setting where coalition structures are taken into account. Question 4a, by contrast, tacitly assumed superadditivity, which makes that coalitional structures can be abstracted away from. The question asked the candidates to prove that, if the core of a game with a given property is not empty, then it also contains an outcome of a particular form. In the general setting with coalition structures and without superadditivity, however, counterexamples to this claim are possible. The students were therefore asked to prove something for which no conclusive proof is possible. In some cases this resulted in rather strained arguments that are hard to judge on their merits. Other students, however, tacitly or overtly made the superadditivity assumption and gave a correct proof. Full marks are awarded in such a case, provided that the assumption of superadditivity was made explicitly. Otherwise, one mark was deducted. A couple of students provided counter-examples, which were awarded according the quality of the counter-example, and full marks could very well be awarded (and were in some cases). This question was of a slightly different character than the other questions, in that it expects the students to write a mathematical proof. This is generally considered to be relatively hard. Most students struggled with this question, although this is presumably due to its unfortunate phrasing. Yet, a handful of students excelled, some of them even daring to give a counterexample.

4b (5) This question concerned the concise representation of characteristic form games by induced subgraphs. This question generally did not present the students with any difficulty, most of them obtaining full marks. Even the hardest part, where the candidates were asked to specify the Shapley values for the players, mostly earned them full marks. As marking scheme I adopted the following: – correctly identifying first game cannot be represented: maximally 1 mark – correctly identifying second game can be represented: maximally 1 mark – correct induced subgraph representation fo the second game: maximally 1 mark – correct Shapley values: maximally 2 marks.

4c (10 marks) In this question two characteristic form games were presented and the candidates were asked to determine whether their cores are empty, and if not, characterise the solutions in it. From the set solutions provided it transpires that the question was phrased for the setting without coalition structures, whereas in the lectures and exercise sheets coalition structures are taken into account. Most students answered the question for the setting with coalition structures, which renders the questions also 5 slightly harder (and harder to mark). This question was not entirely straightforward and should differentiate among the students. This considering the students did somewhat better than could be expected, but not exceedingly so. I adopted the following rough marking scheme: – correctly identifying the core is not empty in first game: maximally 2 marks – correctly characterising the core + justification: maximally 3 marks – correctly identifying the core is empty in the second game: maximally 2 marks – correct argumentation why core is empty in the second game: maximally 3 marks

The examiners decided to adopt the marker's suggestion to use the students' raw marks without question 4a (so that the paper would be marked out of 90), and in addition to apply a mathematical formula to make sure that students who had made a valid attempt at the question would gain marks, but not be penalized if they did less well on it. Again, the average with question 4a discounted was high at 84.4, and 89.2 after the formula had been applied. This prompted the examiners to scale the paper by mapping 50-50, 80-70, and 100-95. The average is now at 81.9.

Computational Learning Theory

As was the case in previous years, students were required to typeset their submissions and submit electronically. In order to ensure that students were spending a reasonable amount of time on the exam and not writing unnecessarily long-winded solutions, strict guidelines on the number of pages permitted, font size and margin width were set.

Exam Summary

For the purpose of this report, I've included students from all cohorts together. There were a total of 5 Part C scripts, 8 MSc in CS (+1 missing) scripts, and 2 MFoCS (+2 missing) scripts. Removing the missing scripts from consideration, the average was 71.6 with a standard deviation of 15.7.

Question 1. Overall the answers to this question were satisfactory. Some students had difficulty with Part (b) of the question. Some students designed unnecessarily complex constructions in Part (a), but were not penalized significantly for that, except when they had inaccuracies.

Question 2. This was intended to be an easier question and that indeed proved to be the case. Most students scored well on this question.

Question 3. This question was intended to be harder than it turned out to be. Students scored well on this question.

Question 4. This proved to be the hardest question on the exam. The intention in Part (b) was that students make only minor modifications to the approach outlined in Part (a). However, some students came up with completely different answers. As the wording was sufficiently ambiguous: allowing students to modify the loss function, no marks were deducted for answers that were correct but strayed significantly from the intended answer.

The average for the five Part C candidates was high at 82.2 (overall 71) and the examiners decided to leave this paper untouched

Computers in Society

We were impressed with the level of sophistication and creativity with which students imagined the possible socioethical impacts of the technologies they discussed. Further, almost every student drew on extensive research from a wide range of sources. The range of marks is typical of those for an essay assignment: virtually all of the answers had merit and some of them were excellent.

The very best exams exhibited a few key qualities. First, the “interventions” (design, organisational, governance) they suggested were detailed and spoke directly and clearly to the potential impacts at hand. Second, they drew on some sort of theory (either from class sources or others) in a meaningful way: they both correctly applied the theory and comprehensively explained how the theory applied in the case under discussion. Third, they were especially insightful in either the possible socioethical impacts or their suggested interventions. This manifested in several ways: for example, discussing impacts or interventions at various “levels” (individual vs structural or near-term vs long term effects, design vs. organisational vs. policy interventions), highlighting less obvious or less discussed ideas, or going very granular in their explanations. The exams that scored less well tended to suffer in certain ways. (Be aware that our summary here is in generalities: some exams exhibited some of the best qualities while also exhibiting some of the less good qualities.) These certain ways included being particularly lacking in the “good” dimensions noted just above. Some interventions were sometimes only briefly discussed or were at too high a level of generality. In some cases, theory was either not discussed at all, discussed only in passing, or misapplied.

The exams that scored less well also tended to be disorganised or harder to understand. This was sometimes because possible effects or interventions were presented more as a running list than in a narrative; there was significant overlap between the sections; or the writing was hard to follow on a sentence-bysentence level. A number of exams lost marks because they relied too heavily on a small number of references - in particular sources supplied in the exam question and/or class materials. Finally, a number of students didn’t read the exam question fully so lost marks for not completing all of the elements required

The average was healthy at 66.3, and the paper had been double-marked. The assessors had supplied a detailed report, commenting favourably on the overall performance. The examiners agreed not to apply any scaling.

Categories Proofs and Processes

The CPP exam this year required the students to understand and use rather advanced material about ends and coends, both of which are concepts that have not been covered during term time. Most of the students did quite well, and they demonstrated the ability to generalise known ideas about limits and colimits to more abstract settings, hence I do not believe any rescaling of the marks will be necessary. Surprisingly enough, marks were lost on conceptually easier questions - for example, the one asking them to prove that the stagewise cartesian product is the categorical product in the category of presheaves. Several students showed that it satisfies a universal property, but more than half failed to check that the universal maps were actually natural transformations, which is the proper notion of morphism in presheaf categories. The essays ranged from incomplete to extraordinary, and there are at least 2 which I think would deserve to be published online somewhere, as literature reviews, as they do a far better job explanationwise than the original papers (the essay about the local state monad and how coends first appeared in computer science being my absolute favourite).

The paper's median was high at 76.3 (counting all submissions), with a considerable deviation of 18.9. However, the examiners noted that the marking was granular, and were therefore not in favour of scaling the paper.

Categorical Quantum Mechanics

This year's cohort performed consistently strong, almost uniformly at a distinction level. It is possible that there was an issue with exam difficulty calibration: due to remote teaching conditions, I might have underestimated the average level of the class. From the quality of many submissions, however, I am more inclined to think that this was—by chance or as byproduct of current circumstance—a rather dedicated class, that by and large took this assignment very seriously.

Detailed question commentary

Question 1: a. Some candidates lost marks because of ill-typed or incomplete definitions. b. Some candidates lost marks because of incorrect usage of the graphical calculus for ribbon categories. c. Some candidates lost marks because of incorrect steps, incorrect usage of the graphical calculus, or because the final scalar was not in the form required by the question.

Question 2: a. Almost all candidates got full marks. b. Some candidates lost marks because they did not define/check the action of the tensor product on morphisms. c. Almost all candidates got full marks. d. Several candidates lost marks because they incorrectly interpreted the snake equations (which followed from those of the base category). e. Almost all candidates got full marks. f. Some candidates lost marks for incorrect/incomplete proofs. g. Some candidates lost marks for incorrect/incomplete proofs.

Given the mark distribution it was decided to scale the marks: 50 -> 50, 90 -> 85, and 100 -> 100.

Concurrent Algorithms and Data Structures

Twenty four fourth year students and seven MSc students took this take-home-exam. The exam consisted of two questions, one asking for the design of a lockfree priority queue in Scala using ideas of a given paper, the other one a bit more theoretical on how to construct a MRMW register from an array of SRSW registers. Overall, the performance was quite good, with lot of candidates showing a good understanding what the given tasks ask them to do.

Question 1:

This question was about designing a lockfree concurrent priority queue in Scala using ideas of a paper by Linden and Jonsson building on the lockfree skiplist discussed in lectures. The average mark achieved was 34.4 out of 50 with a spread between 16 and 48.

- Quality of explanation including rationale for the design:

A number of students just translated the pseudo-code from the paper or the implementation of the authors in C found online to Scala without showing much critical analysis. Others adapted the lock-free skiplist from the lectures with varying degree of analysis of which ideas of the paper are required when memory management is left to the jvm. The quality of the write-up varied from publishable to rarely a sketch or mainly snippets from the given paper. A few aspects were rarely discussed: the need for the "inserting" flag, a requirement for unique priorities or not, and reason for the flag "skew" in locatePreds.

- Correctness and quality of code: Overall the quality of the coding was quite good, and most programs behaved as expected. Eight candidates used uniformly distributed node heights which removes the efficiency advantage that a skiplist has over a singly linked list. One candidate implemented a partial priority queue that by definition can never be lockfree as a removeMin blocks if the queue is empty. Of the submissions with full code bases, a few occasionally deadlocked or produced non-linearizable histories in the linearizablity tester.

- Abstraction and datatype invariants: Abstraction functions were quite good, only occasionally missing some precision or showing some inconsistencies. Regarding dataype invariants the spectrum was more varied. Some candidates copied some or all invariants from the paper without showing any insight. All attempts included the main invariant underlying the idea of the paper, namely that the logically removed nodes always form a prefix of the whole skiplist. Many didn't mention what that means for the usual invariants of the skiplist, i.e., how these invariants need to be adapted.

- Linearization points: The successful lin.-point for removeMin and the lin.-point for add (always successful) were mostly put correctly at the CAS-operation on level 0 that corresponds to the change of the abstract state. The lin.-point for unsuccessful removeMin was sometimes not mentioned at all or put at the test against tail, rather than when the node (that is equal to tail) is read.

- Lock freedom/wait freedom: The arguments provided were mostly good, but sometimes taken as a proof for wait-freedom when it was the correct reasoning for lock freedom, but not wait freedom, "the operation will always terminate unless ...".

- Explanation how the implementation reduces the memory contention caused by the removeMin operations: This was done very well.

Question 2:

This question was about possible constructions of a MRMW register from an array of SRSW registers. The average mark was 34.1 out of 50 with a spread between 10 and 50.

Part (a) asked to show that the given code for a first possible construction is incorrect by providing a non-linearizable history. This was done very well. Most attempts had a detailed explanation how the history can arise, a few being very sketchy or incorrect. The explanation for not being linearizable was sound in most cases.

Part (b) presented a first variation considering readColumn and writeRow calls atomic. In many cases the example from (a) could be used here as well. In fact any non-linearizable history for (b) could have been used for (a) as well. A few candidates tried to show that this is a linearizable implementation.

Part (c) asked about the case when write-calls never overlap. This leads to a linearizable implementation of the sequential register, so required a corresponding proof. A lot of good answers, some good sketches and some attempts showing lin.-points for some possible histories without making it into a general argument.

Part (d) used a variation of the idea of the bakery algorithm, i.e., resolving conflicts of stamps by pairing the stamp with the thread-id, that was studied in lectures. A few candidates misinterpreted the question and showed non-linearizability again. Some candidates tried to prove linearizability by identifying lin.-points in the code, which is a bit tricky because of the "helping" needed for waitfreedom. The attempts ranged from highlighting a few important facts, to complete proofs. Some consisted just of a one sentence proof idea. Some used notations from the course book without explanation. Construction of registers was not a topic presented in detail in lectures. That's why this question was phrased as analysing the linearizability of certain implementations with respect to the given sequential register specification.

The mean for this paper was 69.2, with a standard deviation of 17.7, and the examiners were pleased with the spread of marks and the detailed assessor's report. They decided not to scale the paper.

Database Systems Implementation

Overall, solutions were largely good, following mostly the trend from previous years. This is not completely by chance as the exam design choice for the 50+ and 70+ mark bands is similar to previous years. 50+ and 70+ marks. The most relevant aspect is that the exam gives candidates a clear way to obtain either a 50+ or a 70+ mark, mainly by choosing Q1b and Q2b (as well as Q3b). Most candidates (16 out of 21) decided to show proficiency at the 70+ bands – all of them succeeded, 9 of them at the top end (80+).

Overall, solutions were largely good, following mostly the trend from previous years. The exam design choice for the 50+ and 70+ mark bands is similar to previous years. There was a good distribution of marks, and the Assessor and Board agreed that no scaling was necessary.

Foundations of Computer Science

The FoCS exam paper consisted of 8 questions of various difficulty: questions 1 and 7 were considered relatively simple, questions 2, 3 and 4 medium, and 5, 6 and 8 difficult. Generally, these expectations confirmed: maybe, only questions 5 turned out to be easier. Overall, I believe the exam was of appropriate difficulty and very good in distributing the students: there are good, medium and weak works.

Next I give more detailed feedback for each question of the exam.

Question 1: Generally straightforward, only very few students had problems.

Question 2: Most of the students correctly figured out that the new class of languages are incomparable with regular and context-free languages; only a couple of them wrongly claimed that the all languages in the new class are regular. However, many students could not prove, with an appropriate level of formality, that there are new languages that are not context free.

Question 3: Most of the students correctly understood that the right-flooding tape is essentially a stack, however, less figured out that the new machines are more expressive than pushdown automata. Surprisingly, very few considered both deterministic and non-deterministic versions of flowing machines, which was explicitly asked in the question.

Question 4: The main challenge of this question was to correctly formulate the (formal) language to study. Some students could not do this, and essentially considered a set of infinite words, which is not a language by definition. Others, while formulating the language correctly, could not give a correct reduction of a known non-semi-decidable language, or even claimed that the problem is not decidable but semi-decidable.

Question 5: I am pleased to notice that most of the students understood the idea of NP-completeness correctly and managed to come up to a non-trivial reduction in this question.

Question 6: Most of the students managed most of the question well. Most common problem was that some did not figured out that class NP does not change by itself with respect to new reductions (its definition does not depend on reductions), but is not closed under these reductions, contrary to all usual cases.

Question 7: This question is quite straightforward and just checks for basic understanding of first-order logic. Most students managed it well, only few had technical problems writing a correct formula.

Question 8: As expected, this was the most challenging question in the exam. However, it is still disappointing that none of the students came up to an optimal EXPTIME algorithm, which is in fact a standard materialisation algorithm for Datalog (the students were not supposed to know this language, but could come to it by themselves in the literature, or, more likely, to come up to this quite natural algorithm by themselves). Anyway, some of the students did not understand that propositional logic satisfiability algorithms do not work for first-order logic without a major preprocessing; others gave an essentially correct algorithm, but wrongly assumed that the interpretation domain (or the set of constants in the signature) are finite and given as input. A couple of students gave correct algorithms (e.g., based on the algorithm for the Bernays–Schönfinkel class), but these algorithms are not optimal.

Knowledge Representation and Reasoning

The exam worked well. Apart from one outlier with 28%, the marks ranged from 44% to 92% with a good spread in between and a median of 69%. Q1 was a little easier and was attempted by all but one of the students with a median mark of 76% and no student getting less than 48%. Q2 was more challenging, with one student getting only 8%; however, this student attempted only a small fraction of the question (I assume they ran out of time or had some other problem). Moreover, several students performed very well on this question, with two getting 96%. Q3 was somewhere between the other two, with marks ranging from 24% to 96%, with a median of 68% As the assessor commented that the exam worked well, and there was no obvious need for scaling. The average was 66.2. The examiners were pleased with the quality of the exam paper and the distribution of marks, and decided not to apply any scaling.

Lambda Calculus and Types

Overall the candidates performed very well and seemed confident with all important concepts.

41 students attempted Question 1. Part (a) was usually answered correctly, though with various levels of clarity. Part (b) was almost always correct. Almost all students understood Part (c) in a different way than I intended and therefore I marked it generously; many students made impressive attempts at a challenging question. For Part (e)(ii), which was just a programming exercise, students often had a correct idea but the details and presentation were lacking.

Question 2 was the second most popular, with 31 attempts, and students generally did very well, though a few seem to have run out of time. Parts (a) and (b) were broken down into many simple steps, and most students got full marks. Given (a) and (b), (c) was basically free marks, although I expected a proof that m\infty is unsolvable, which many students omitted. (d) was easier than I intended, partly because the topic was very close to the well-known result that H* is maximally consistent.

Only 12 students attempted Question 3, which was about simple types and unification; I think that these topics are generally less popular. Mostly students did well, although in part (c) and (d) there were some misunderstandings about what could and couldn't be proved by induction.

The average was high at 74.2, with a particularly large number of very high marks. The examiners scrutinised the borderline scripts and were confident that the distinction borderline was in the right place.

They agreed to bring the top marks down by scaling 48 (raw marks)-85(USM), and 35-70. This brought the average down to 70.8.

Law and Computer Science

This is the second year that we have run this course jointly between the Faculties of Law and Computer Science, where it is open to students on the BCL/MJur/MLF and 4th year/MSc courses respectively. As last year, the course contains two summative components; the written paper which is the subject of this report and a practical project which required students to work in interdisciplinary groups of 6 (three from each discipline) to produce a legal product based either on blockchain technology or NLP. The practical project is marked simply on a three-mark scale: satisfactory, satisfactory – or satisfactory+. Such was the quality of this year's projects that all students received an S+ mark. In law this does not have any implications for the candidates' overall degree classification, though such a mark can have that effect in Computer Science.

Like last year, the best scripts in the theoretical paper were those where the candidates had engaged in detailed analysis of particular, specific examples considered from both a law and a computer science perspective in order to analyse how both disciplines can contribute to the solution of particular problems. This is the key goal of the course and one that was emphasised to students in all sessions. Weaker scripts tended to focus on their own discipline without engaging fully with the other, thereby failing to achieve this central goal.

As last year, all papers were marked by both examiners in order to ensure that tey had been marked from a similarly interdisciplinary perspective. Notably, and reassuringly, like last year there was not usually a significant difference between the marks awarded by each examiner when these were compared at the end of the marking process. Where there were minor discrepancies these were discussed and a final mark was agreed upon.

The questions in Part A were based on the first half of the course, covered in Michaelmas Term, and examined the interrelationship between the two disciplines and the ways in which technology might affect the process of law and/or legal practice. We were pleased to see that this year answers were more evenly distributed across the questions, though questions 5 and 7 were the most popular. Our detailed comments on each question are as follows:

Part A

1. This question required consideration of both the theoretical differences between the two disciplines, in particular the need for vagueness in law and why this arises. Strong answers were able to include consideration of the specific circumstances in which code and law interact, such as the DAO experience in smart contracts, machine readable legislation, and some of the examples discussed in Lessig's work.

2. This question asked candidates to consider the advantages and disadvantages of automation and augmentation in both technical and ethical/sociological terms, examining materials from across the first term of the course. As before, strong candidates were able to give specific examples of both automation and augmentation and advanced a normative approach to weighing the pros and cons and making a policy decision in each case.
3. Successful answers to this question were able to give specifics on issues such as risk assessment and treatment, vagueness and professional duties and to explain how those factors might carry through into interactions between the two disciplines. Again, stronger candidates were able to present a normative thesis about their interaction.

4. The first half of this question required an examination of what it is that a lawyer does, both in terms of Susskind's own work and that of others such as Hildebrandt and Howarth, and whether or not it is only the outcome of a case or legal process that is important, drawing on the work of Mulcahy, Tidball and others. The second half required an examination of what legal technology can replace or alter, and what it should not, in the light of the initial discussion.

5. This question required a good analysis and understanding of the technical techniques for achieving explainability, both in terms of the differences in inherent explainability between different kinds of systems, and the techniques for achieving explainability ex post in less transparent systems. It also required a normative thesis of what level of

explainability is necessary for deployment of technology in different circumstances. As before, strong candidates were able to refer to specific examples.

Part B

6. This question required an understanding of the specific challenges arising from algorithmic decision making (ADM) including, but not limited to, bias in data or algorithms themselves, transparency, population but not individual level accuracy, correlation as opposed to causation, scaling, rigidity etc. It then required candidates to demonstrate an understanding of areas of law such as public law and the law against discrimination and in particular the extent to which those areas already contain tools capable of addressing these challenges and the extent to which further development is necessary before they can do so.

7. This question required an understanding of different 'metrics' for assessing the performance of a system. Stronger candidates were able to address the question of 'success' in a nuanced way, outlining both technical metrics such as sensitivity, specificity, precision etc and 'softer' metrics such as efficiency, equality and fairness in relation to different populations. The question also required attention to the legal rules which might require the application of such metrics before a system is considered 'reasonable', within public law for example, or unlikely to attract a claim in tort law.

8. This question asked candidates to examine the areas of law dealing with harms, including particularly tort and criminal law, but potentially also competition and public law, and the extent to which those areas of law, designed for application to human beings, are capable of applying to digital technology. In the contexts where further modifications are needed, stronger candidates were able to outline where these modifications should be technical (in terms of gathering evidence or increasing transparency and auditability) or legal (such as the alteration of the definitions of particular offences).
9. This question asked candidates to consider the law relating to property as we had studied it in relation to both cryptocurrencies and intellectual property and the extent to which those areas are in need of adaptation before they can apply successfully in the digital context. In class we had spent time examining and discussing the *Oracle v Google* litigation concerning APIs, and stronger candidates were able to draw on this successfully in their answers, though it was not a strict requirement that they do so.

10. This question was a slightly broader question which could be answered successfully using material from across the second half of the course. Stronger candidates were able to give specific examples of technology-led solutions (such as automated compliance) and legal ones (in terms of the incentive structures of particular areas of law such as tort law).

The examiners followed the proposed scaling mapping 70-70 and 73-78 to increase the average slightly to 68.9.

Machine Learning

In general, the overall level of answers and the knowledge of most of the candidates were excellent. The answers to the questions were effective in differentiating the candidates. Question 1 was answered by all candidates but one; Question 2 was answered by 21 candidates; and Question 3 was answered by 40 candidates.

Question 1 Most candidates answered parts (a) to (c) well, with several outstanding answers throughout; answers were differentiated by accuracy, completeness, focus and rigour. At part (d), many candidates stated that feature independence implies that the features are class-conditionally independent; others gave vague statements about better classifiers, with either insufficient reference to the question, or with no specific examples and/or classifier that would perform better, and why. At part (e), most candidates answered well, with the weaker answers only referring to the Gaussian radial basis function, or vaguely to SVM regularisation. There were only a few rigorous and complete answers to part (f), with many answers incorrect, or just mentioning the impact of test points infinitely far away from the training points on the kernel values, with no further discussion of, or detail on, the impact of the decision function. Parts (d) and (f) were the key differentiating parts of this question.

Question 2 Most candidates answered parts (a) to (b) well, showing a good understanding of theory and their ability to apply it to simple examples, and to compare and contrast different methods. At part (c), most candidates gave accurate time complexity answers, but there were few fully accurate space complexity answers, with most of them not considering both pre-activation and activation vectors. Consequently, most answers did not identify the constant reduction in memory for invertible non-linearities. Most candidates answered part (d) well, with the best answers also fully discussing the limitations of their proposed approach. Most of the answers to part (e) were incomplete, with a few good attempts to simplify the notation and to spot useful patterns in the specification of the ResNet architecture. The memory efficiency part of the question was either not answered, or answered mainly qualitatively, rather than based on the relationship between the forward and backward equations. Parts (c) and (e) were the key differentiating parts of this question.

Question 3 Parts (a) to (c) were effective in testing the candidates' understanding of key concepts and methods, and their ability to reason about them. Parts (d) and (e)(i) were mostly answered correctly, with weaker answers being less rigorous. There were few complete and fully rigorous answers at part (e)(ii), although many candidates correctly specified the neural network optimisation problem and stated its link to the PCA objective function. At part (e)(iii), many candidates gave correct, either mathematical or qualitative, answers. Part (e)(ii) was the main differentiating part of this question, with some candidates possibly running out of time on the final parts of this question

The exam paper was well set. The assessors commented that "the overall level of answers and the knowledge of most of the candidates were excellent. The answers to the questions were effective in differentiating the candidates." The average was healthy at 67.2, and the distribution of marks satisfactory. The examiners agreed there was no need for scaling.

Principles of Programming Languages

Questions 3 had mostly very good answers, but not very many students attempted it. This suggests to me that only the most competent students took this question.

In Question 1, almost all students demonstrated basic competence in the topics of the question. But I was a disappointed that so few students correctly answered part 1a. It is a little more complex than continuation passing style questions from the past. Only the students who correctly answered 1a should be regarded as excelling in this topic.

In Question 2, almost all students demonstrated basic competence in the topics of the question. For example, Question 2c was perfectly answered by almost all the students. The poorest answers were in parts 2a and 2b. In 2a, some students neglecting to show that the limit of the chain is in P_f. Part 2b requires some imagination so I was expecting that to be the tough part and it was expected that the answers to 2b are quite varied. In marking the question I realized that 2d(ii) admits a shorter answer than the one in the marking notes, but most students who were successful in this part used the longer proof that is in the marking notes, so I do not propose to change the mark for this question.

Several students submitted pages in an apparently random order, with different questions mixed up together, for example part 2c, then 3a, then 2b. This made marking very fiddly and time consuming. I have been careful to look carefully through all the submitted pages. But I suggest in future to require that pages are uploaded in a sensible order.

Overall, the assessor judged that "that the marks accurately reflect the qualitative descriptions of the mark boundaries, and I do not think any rescaling is necessary." The average was 65.9, and the examiners agreed not to scale the paper.

Probability and Computing

Question 1 has been mastered by almost all students perfectly. Only few submissions presented a wrong coupling in Question 1 c).

The statement of Question 2 a) contained a rounding error, which was discovered early by the students; the correct statement of the question was broadcasted to all candidates taking the exam.

The answers to 2 a) have been of high quality in general. However, many students did not get full marks due to missing/incomplete arguments in the proof or due to missing rigour. Question 2 b) was successfully solved by almost all of the students.

Question 3 a) was also solved by most of the students completely. The answers to 3 b) have been of high quality, but almost no student got full marks: Some students presented a flawed coupling, others did not rigorously bound the expectation or missed cases in the analysis, and most students did not argue properly about the change of the distance after one step in the coupling.

In contrast, the answers to Question 4 could mostly be split into perfect solutions and non-attempts: Almost everyone who had the correct idea – which is true for most of the candidates – was also able to properly present the solution. The only exception to this were submissions in which it was erroneously claimed that BFS can solve the uncoloured k-path problem, which essentially resulted in a flawed solution for Question 4 b).

The candidates performed very well on this paper. The mean is 78.2, and the median is 89, which is significantly higher than last year.

Physically Based Rendering

Despite the remote teaching circumstances and the course not being taught, all candidates performed remarkably well at Question 1 and Question 2.

In Q1 Minor differences from the model implementation resulted in some discernible quantitative differences in the renders, which were however qualitatively hard to distinguish from the model ones. Code was well-documented and discussion of techniques and result was competent throughout.

In Q2 each student demonstrated their ability to skilfully apply image assessment and user study knowledge to well-thought through experiments, all very different from each other. All students demonstrated excellent independent thinking, and all answers were very well justified throughout.

Probabilistic Model Checking

Question 1 comprised a DTMC modelling part, a second part dealing with semantics of probabilistic temporal logic, and a final part dealing with the verification of a given PCTL formula. The answers varied, with up and downs on the 2nd and 3rd parts. Comments have been added for single students.

Question 2 initially dealt with the building of a CTMC model, then it focussed on checking a steady-state temporal property, and concluded with encoding two reward-based properties. Again the last two parts were addressed in-homogeneously, whereas the modelling part was broadly ok. Comments have been added for single students.

Question 3 dealt with non-determinism and probabilism over MDPs. I expected it to be the hardest among questions, however students took it quite well, in fact most of them nailed it via examples provided. Comments have been added wherever students have failed to address some of these parts.

The paper had a high average of 86 which was out of line with the students' performance in their other papers. The examiners decided to follow the assessor's recommendation to scale down by mapping 67-57, 80-70, and 96-90, creating an average of 78.2

Quantum Processes and Computation

All in all, the exam was very well done by the students, with the majority of people showing a good handle both on graphical reasoning and the fundamentals of quantum theory. I will highlight a few places where students lost points.

Some points were lost due to a lack of clarity in proofs. For example, when proving an equation G = H by diagrammatic reasoning, it's often most clear to right this as a chain G = G2 = G3 = … = H, where each step is small and justified (e.g. by an equation number). Similarly, some points were lost for omitted or sketched proofs. Giving the idea of how to do the proof is not enough: it should be written out explicitly and for all cases.

Generally students did well on Q1, with the exception of occasionally losing a point or two for overly convoluted derivations in the last 2 parts.

A mistake made by several students on Q2(b) was to give a protocol where the two parties ended up with 2 classical cups. This corresponds to having 2 correlated classical bits, which is not the same as sharing an entangled quantum state. The most common mistake was giving a protocol that relied on non-causal processes (e.g. having parties "do" a quantum white dot effect). Only causal processes are deterministically physically realisable, so a protocol should only consist of these (or at least say what happens when a non-deterministic process fails to produce the desired effect).

For Q2c(i), again some people lost points for allowing Alice and Bob to do non-causal processes, or do things that don't make sense physically (e.g. plugging in Alice's states to the outputs of Bob's process). Some also lost points for just drawing a picture without giving some labels or an explanation of how the protocol works Which parts are done by Alice? Which parts by Bob?

For Q2c(ii), Bob can get some information about which state Alice prepares by measuring some or all of the qubits Alice sends. Some students wrote this, but gave an explanation which showed some lack of understanding of what information this gives to Bob (namely: one of several outcomes with some probability).

Many people got the right idea for Q2c(iii): that Alice should choose randomly between white-0 and white-pi to send a white spider and between gray-0 and gray-pi to send a gray spider. However, they only gave a justification that Bob's strategy for getting information they gave in Q2c(ii) wouldn't work any more, whereas they needed to show there was no possible way for Bob to get any information about Alice's state. This holds because Bob sees a mixture, either white-0 + white-pi or gray-0 + gray-pi. Both of these are equal to the maximally mixed state, so cannot be distinguished by any quantum measurement.

Q3a was generally well done, and I noticed no major issues among the students that attempted it.

For Q3b, some students applied essential uniqueness of purification to processes which are not pure (and hence are not purifications). This was usually to create a process from B to B' (e.g. by discarding B and preparing a state of type B'), in order to make the discarded types match on the left and the right of the equation. The way the types can be matched while keeping everything pure is to introduce a new pure state of type B' on the left and a new pure state of type B on the right. Many students ignored this mismatch, either by implicitly assuming that B = B' or that a pure process from B to B' exists (which is only the case of dim(B') >= dim(B)). However, I decided not to deduct points for this minor mistake.

The examiners noted that the mean of this paper was very high at 83.6. They decided to map 100-100, 90-85, and 52-50 and scale in a piecewise linear function. The average after scaling was 80.7, with a standard deviation of 16

Requirements

Overall this was a good set of mini projects, with the vast majority of students performing well in the task given. They demonstrated that they had learn a great deal across the course relating to the methods of requirements elicitation and the practices of communicating requirements.
The students who performed in the most well did some, or all, of the following:

- consider how different data sources could complement each other – for instance to triangulate findings or cover overcome limitations/omissions of the alternative data source
- specified how much data had been collected, from where and the precise process of analysis
- referred to academic sources when describing the process of analysis
- paid careful attention to the specifics of presenting requirements in SRS, Volere templates etc.
- included critical reflections of where they could have improved their work in the reflections
- set out next steps that related directly to the artefact chosen and the lessons learnt from the reflections

The students who performed less well often did some, or all, of the following:

- relied over heavily on a single data source
- did not specify how much data had been collected and where from etc. and where vague about how it had been analysed
- had errors in the communication of the requirements
- presented generic next steps – for instance the description of a focus group, questionnaire etc as it might appear in a text book rather than relating it the elicitation of requirements for their chosen artefact and previous reflections
- under referenced their report
- ignored/neglected the font size and line spacing criteria

Practical Work

All candidates passed the practical requirement.

9) Names of Members of the Board of Examiners

Prof. Santhanam (Chair)
Prof. Creese
Prof. Kay
Prof. Gottlob
Prof. Koutsoupias
Prof. Kalvala (External, University of Warwick)