PGT Exam Performance Feedback
Semester 1

COMP60021  Grid Computing and eScience  John Brooke

Comments:
The questions were generally well answered with the standard of written descriptions and diagrams being good overall.

The main reason for dropping marks was that most students gave answers to the compulsory question 1 that were too long. Thus they ran out of time on the optional questions where they generally did better, perhaps because they could choose their strongest topics. Each section of Question 1 (a to e) required no more than 6-8 lines of writing to give all the information for full marks. Some students wrote over a page or in some cases two pages on each section. There is no point in doing this since each section carries a maximum of four marks.

Students did better than in the previous year in terms of actually reading the questions and answering the questions rather than just writing what they knew about a particular topic without considering what the question was actually asking. Those student who did answer after carefully reading the questions generally got higher marks for less writing.

There were a few students with very low scores, this was mainly because they only answered parts of each question. This suggests they either did not understand the course material or that they did insufficient revision. If students do not understand the course material they should ask in the lectures or in the labs.

COMP60041  Low-Power System Design  Steve Furber

Comments:
This year's cohort was small - just 7 students - so the results are statistically unreliable.

All students attempted question 1 and achieved marks over 65%, showing that they had at least a basic working knowledge of the ARM instruction set architecture.

All students attempted question 2, with most achieving good marks. This question was closely based on one of the coursework projects, so anyone who had made a serious attempt at the project should find the question straightforward.

All students but one attempted question 3, generally achieving good marks. This question tested knowledge of cache architecture and low-power issues; the results show that low-power memory hierarchy issues have been absorbed by those taking the course.

No student attempted question 4, which was on processor pipelines. This suggests the students have mainly picked up system and software issues from the course, and not the hardware architecture issues.

One student attempted question 5, which was on the system development process. The question requires detailed knowledge and understanding of several aspects of the course bookwork, and it may be quite hard to get high marks.
Q1 Visualization Critic

Majority of marks were lost due to either not considering the correct formal definitions; and by only mentioning a sub-set of the graphical integrity items when requested. It should be noted if ‘n’ marks are being awarded for a set of point, it is likely that ‘n’ sub-answers are often expected and not just a repeated description of the same point.

Visualization critics can be improved with practice and discussion; including the ability to make good simple estimates of specific ratios.

Q2: Perception Issues

A few students described a far more complex scenario than was asked – as the more complex scenario was given during lectures. If a simple scenario can be used, but described in detail then this is potentially both faster and creates a cleaner answer.

Secondly marks were lost for drawing accurate figures but then not describing each part.

Q3: Marching Cubes Algorithms

Marks were lost for poor description of pseudo-code – which in its best form should be convertible to code without complex ‘hidden’ descriptions. Carelessness also meant that straight line segments became ‘artistic’ curves as this potentially was easier to draw. When considering presenting results from an algorithm it is best to think like a computer and carry out each stage systematically.

Q4: Parallel Strategies

Questions on parallelism can appear complex, but given the right terms are often more straightforward than expected. The key options are often consistent in all parallel issues; but practice previous questions will aid in future exams.

Q5: Flow Visualization

Main areas were marks were lost were firstly in not considering each part of the question in detail; but also in presenting two solutions as different, which were effectively identical in principle.

It is always worth reviewing your answers to the questions before handing in as this will stop careless errors. And always re-read the question after answering.
(1a) Some students parsed a formula of the form $A \land B \to C$ as $A \land (B \to C)$, which resulted in wrong answers. The answers were not great in general: some wrong interpretations were given with no or incorrect explanations.

(1b) Generally good
(1c) Good
(1d) The question was just to formulate a couple of definitions - only one answer was correct (probably lack of culture/background in math).

(2a) Only one of the reversal programs was correct. The concatenation program was generally OK. Several mistakes based on mixing a single-element list with the element.
(2c) Two correct answers based on applying resolution. Some mistakes based using an interpretation $I$ with $I \models p(x)$ - this cannot be done since $p(x)$ has free variables.
(2d) Good

General mistakes (infrequent)
- giving insufficient explanations or wrong explanations
- not saying what the answer is
- not reading/understanding question properly
- not using/understanding (?) definitions in appendix.

Question 3
********

a) i) posed no problems ii) common mistake: not to say what the answer is and/or applying inference steps, which is wrong iii) bookwork, poorly answered
b) i) generally well answered ii) answers mixed, but mostly good iii) answers not so good; most gave a correct argument why positive factoring is not needed, but not for negative factoring

Question 4
********

a) bookwork, answers good.

b) bookwork, answers ok.

c) common mistake: not to use the calculus from Part II of the course, and using negative factoring.
Question 1

a) mixed - some good, but too many didn't know what 'satisfiable' or 'valid' means, or didn't use how to use a (counter)model to explain why a formula is satisfiable/not valid.

b) mostly ok, but too many didn't know that 'valid' means 'true in *all* interpretations and 'satisfiable' means 'true in *some* interpretation'. Of those, many answered (ii) wrongly (i.e., they said it was false, but it is true).

c) i) most answers correct  ii) common mistake: not to continue the application of the theory rules in the branch that could not be closed.

Question 2

a) Mostly ok - sometimes little errors like wrong syntax for number restrictions.

b) Mostly good. If something was wrong, then mostly in the reading-off of the formula.

c) Same as (b)

d) was not answered by some students. Of those who did, mostly ok.

e) again not answered by several students. Only few answered correctly that a:D and a:not D are both consistent with input because we have 2 extensions of this default theory (and we have those 2 extensions because T and A entail a:C1 and a: C2): some computed a single extension containing both.

Question 3

a) i) bookwork, poor answers  ii) bookwork, poor answer, but slightly better than in i)

b) common mistakes: insufficient justification, not to give 'formal derivation', wrong expansion of quantification, unclear bracketing

c) common mistakes: not to specify an world and model, to give a model and world in which the formula is true - this is wrong, the questions asks for a falsifying model

d) i) answers predominantly correct  ii) common mistake: not to be able to work out the interpretation of exists R.C and exists S.C
Q1: Joint most popular question and generally very well answered. Most people knew how compositing worked although some of the answers to the 8-bit matte question were a little vague (often no description of antialiasing). The arithmetic for the Over operator was generally correct although some didn’t spot the optimization of using only one of the rgb components because they were identical. Marks were usually dropped in the final part where a mix operator had to be designed. A lot of people went for a simple average of the two input colours. Instead a linear interpolation between the two colours, using weights x and (1-x) was the solution.

Q2: Joint most popular question. Most people knew how Euler angles are represented but many failed to mention that rotations about specified axes are independent and that an order in which the axes are considered should be chosen and used consistently. Marks were lost in the description of interpolation with Euler angles. Here we were looking for use of motion graphs to interpolate each angle (x,y,z for example) independently. Several people wanted to interpolate the components of the rotation matrices from one matrix to another - not correct. The basics of quaternions were generally known but the description of combining Euler angles and quaternions in a user interface was often poor. We were looking for the use of Eulers in the UI and converting to quaternions internally in the software when interpolating orientation. Rendering can convert the quaternion to a matrix.

Q3: A popular question. Generally well answered but a few marks lost due to not knowing what is stored in nodes and arcs of a tree representation. This was thought of as an easy four marks but many people only gave object geometry and a list of nodes as the answer. The tree traversal algorithm was often a little unclear but this was perhaps a difficult 5 marks although some good answers were given. Pseudo-code or a general description were accepted. Most people knew a stack should be used but didn’t quite explain why and were in the algorithm it should be used. Many people didn’t know what the Jacobian was used for (relating joint angle velocities to end effector velocity; the Jacobian is inverted to solve the equation). There were some good answers to the final part asking for an animation tool but often little detail was given with simply a high level technique being mentioned without more explanation as to how it would be used or combined with other techniques.

Q4: This question was answered thoroughly by most candidates who had understood the concepts from the notes as well as implementation issues; English and quality of presentation was very good. Lacking by many of the candidates was to consider order of complexity of the proposed systems and also interaction with other rendering systems; for example adding collision detection (allowing computer generated sparks, created by a particle system, to bounce along a surface).

Q5: Least popular question (Q5 usually is). This wasn’t answered particularly well but that may have been due to time. Many people didn't state that dynamics considers forces. The extension from particles to rigid bodies was not well answered with many overlooking the need to consider distributed mass, centre of gravity, rotational motion. The third part was asking for the update cycle of a dynamics simulator although it wasn’t asked so explicitly. That should have been a straight forward 5 marks. The accuracy of the technique was referring to the integration scheme used (Euler integration). Most people knew the difference between the two motion capture techniques and what problems may arise with optical tracking. However the post-processing method was often lacking detail - examples were available in the lecture notes (offset curves, motion warping).
The exam was taken by 37 students, of which 36 were serious attempts. The percentage mean was 59.8 with a standard deviation of 18.4. This compares favourably with past year's exam's percentage mean of 53.95 and would have been higher if the second most popular question had not proved the hardest of all.

Question 1

This was taken by 34/37 of the students, the most popular question. It was the second easiest, with a percentage mean of 62.9. No systematic errors worth commenting on were identified in marking.

Question 2

This was taken by 27/37 of the students, the second most popular questions. It was, however, the hardest question of all, with a percentage mean of 44.7. Many students had very poor recall of the impact of hash join on a pipelined evaluation strategy. Many students addressed the issue of partitioning strategies with respect to distribution (which was not asked) as opposed to parallelisation (which was asked). Thus, horizontal v. vertical partitioning are (data) distribution strategies, whereas round-robin, range and hash-based partitioning are (data) parallelisation strategies. Many students also confused the size of the output with the size of the inputs (in the item about buffer sizes required for the optimiser to select nested-loop join). Finally, too many students simply did not address the question being asked.

Question 3

This was taken by 24/27 of the student, the third most popular question. It was the easiest one, with a percentage mean of 70.7. No systematic errors worth commenting on were identified in marking.

Question 4

This was taken by 11/37 students, the second least popular question. It was the second easiest, with a percentage mean of 61.1. Many students failed to provide specific details on how this specific query translates to a map-reduce computation. The simplest way to do this is, as in the lecture notes, via relational algebraic translation. Some students chose to show the corresponding Java for the mapper and the reducer functions. The problem with this is that, in terms of queries, it leaves open the question of taking the union of the fragments.

Question 5

This was taken by 10/37 students, the least popular question. It was the second hardest, with a percentage mean of 58.0. Many students had poor recall of the need for time synchronisation and of the synchronisation strategy used in TinyDB.

In total, 64 students sit in the exam.

Question 1: taken by 32 students, and the average mark is 56.5%;
Question 2: taken by 56 students, and the average mark is 63.7%;
Question 3: taken by 46 students, and the average mark is 68.3%;
Question 4: taken by 53 students, and the average mark is 63.5%;
Question 5: taken by 5 students, and the average mark is 53%.

Overall speaking, I think the questions are set at the right level.
Q1:

Most did well on this, with "textbook" definitions of the simpler aspects of the SVM. Few people (only 3) gave full descriptions of the whole process including lagrange, QP, and the correct meanings of the objective/penalty parts of the optimization problem.

Q2:

(b) Most people attempted this but almost all failed to write correct definitions of wrappers versus filters, many confusing them the definitions of forward versus backward search. Both forward/backward can be used *after* the choice of a wrapper/filter.

(c) Few gave full descriptions of cross-validation and experimental procedures, instead choosing to give convoluted definitions of mutual information.

Q3. Parts attempted were answered quite well but there were some common mistakes,

b) The additional parameters was p(j), the probability of a component, or possibly K the number of components. Some people specified the mean and variance of p(xj) but that is specific to a Gaussian mixture model and the question did not specify that this is a Gaussian mixture model. Also, these parameters would be denoted by the w.

c) EM is guaranteed to not decrease the likelihood - some people defined the maximum likelihood, but that is the likelihood after maximisation (the maximum likelihood value is what the EM algorithm converges too eventually, it is the likelihood that increases each iteration not the maximum likelihood).

Q4. Again, parts attempted were answered quite well but there were some common mistakes,

c) p(C1C1) = 0.2*0.1*0.5*0.1*0.2 = 0.0002

The results were quite satisfactory. Most students got 50 per cent. Only one did badly (10/60). In general the students showed a good grasp of the math concepts taught, which is good for non-mathematicians, but a common fault is lack of accuracy often on trivial aspects e.g. consistency in naming objects etc. Overall slightly better than previous years.

Please see attached pdf.
A1. Few students chose this question, but the (few) answers are good, missing only detailed requirements stock control for the shop, e.g. for recording sales, in the example.

A2. The answers are good overall. The weaker answers show a lack of understanding of requirements engineering in general.

B1: This question was very popular with the students and the average mark was 12. Most of part a) was bookwork, but many students struggled with focusing their answers on delete operations (many describing general integrity constraints). Part b) was mainly ok, with many students failing to explain the process of EER to relational mapping. Surprisingly, explaining the concept of overlapping sub-classes proved to be difficult.

B2: This question was taken by all students, but the average mark was less then 10. Again, some students struggled with bookwork in part a). SQL queries that needed ‘group by’ were very problematic, as well as update and create statements. Some answers didn’t have join constraints in their queries.

B3: This was not a popular question, but it seems it was well done by those who have taken it - the average mark was almost 14. Bookwork (part a) proved again to be an issue, as well as providing the examples. Part b) was generally ok. OQL statements didn’t take into account that some relations are sets (and that one can’t refer to an individual attribute in that case).