UG Exam Performance Feedback
First Year - Semester 2

COMP 11120  Mathematical Techniques for Computer Science

Graham Gough/Aravind
Vijayaraghavan
Mark Kambites (Maths)

Comments: Please see attached PDF.
The following only concerns Section A of this exam.

General remarks: 182 students answered questions from this part.

Comments on exam technique: Firstly I am rather surprised at how few students answered Question 3. This makes me wonder whether many students just do the first two questions for no better reason that they appear first on the script, or whether they do not prepare for questions on context-free grammars. Students should always read all the questions and then decide which ones they're best equipped to answer. While working on past papers it would be a good idea to keep track of what kind of questions you do particularly well on and which ones you tend to struggle with. That will help making decisions when you are sitting the actual exam.

Again it was good to see students starting to use sample words to test their solutions, and I noticed a few cases where students managed to correct their answer based on such tests. Other students, however, were not very creative in coming up with sample words, so this technique didn't help them.

Question 1. 178 students attempted this question. The average mark was 5.1 out of 10. 35 students received 2 marks or fewer (14 had a mark of 0), 53 students got 8 marks or higher. Compared with last year the average is lower, and the proportion of very low marks has increased, but so has the proportion of very high marks. If students who got 0 marks for this question are excluded the average increases to 5.5. Quite a few students had fairly long calculations that didn’t go anywhere, and then suddenly seemed to realize that this wasn’t so difficult and came up with something quite short and at least partially correct.

a) Many students did not seem to know what `precede' means, namely `comes before' (not `comes after', which many students seemed to assume). As a consequence there were a lot of wrong answers here. Also, many students had automata where as had to come in pairs immediately one after the other, which is not general enough. The smallest automaton that works has four states and it can be generated as the product of two automata with two states each.

b) Typical mistakes were not to allow any words not containing a at all, or not allowing for an arbitrary number of bs and bcs freely mixed before, between, and after any as. Some students tried to use Algorithm 2 to do this, but they typically made mistakes and started from an incorrect automaton.

c) A few students did not even try this part, making me wonder why they chose to answer Question 1 rather than one of the other two. Where students were making obviously the same mistake as in part b) (typically using cb instead of bc) I did not count it against them again. However, those who used an answer to a previous part to help with this did not seem to spend any time in checking whether that answer was correct!

Question 2. This question was attempted by 160 students. The average mark was 5.4 out of 10, very similar to last year. Twelve students received a mark of 2 or lower, while 41 students got a mark of eight or more. Only two students got a mark of 0 so removing them does not change the average by much.

a) Most students found the three words of length four accepted by the automaton and got all three marks. Quite a few missed out one of the words, and a handful wrote down what seemed a random collection of four letter words.

b) The vast majority of students who tried to read off a regular expression from the automaton did not succeed, missing out substantial parts. Many
of those who did use Algorithm 2 made mistakes with the indices and got a non-sensical answer. Those who were able to use the algorithm correctly but made minor mistakes along the way typically received at least five out of the seven marks available (it did not seem to matter too much which indices had been chosen for which state). Most common mistake: The number of the state that appears in the cycle was set to the highest number currently available, when it has to be one more than that.

Question 3. This question was attempted by only 32 students. It had an average mark of 4.2 out of 10. Nine students received a mark of at most 2, while four managed 8 or more. If we again remove the students who got 0 we have an average of 4.9.

a) Many students wrote something correct here. Popular choices where that every such words contains the substring ab (or bab), or that every such string ends in b. Some students gave trivial properties or incorrect ones and received 0.

b) There were a fair number of incorrect answers. The language is regular, and this is most easily demonstrated by producing a regular expression for it. For some reason quite a few people could give the right answer but could not give a valid reason. One student gave an NFA instead and that was okay too.

C) I was a bit surprised that almost no one could answer this correctly. Most students started with a rule that said $S \mid TaT$ and then had rules for $T$, but this won't work because those rules can then be used to have the first $T$ turn into a larger word than the second, and the $a$ generated in the first step will then no longer be the middle symbol. There are two fairly easy grammars that do this. The first of these has one non-terminal symbol and five clauses, the second has two non-terminal symbols, and two clauses for the start symbols and two for the other non-terminal.

D) The language cannot be regular since we would have to count how many symbols we have seen so far which is not possible using, say, a DFA. Again there were a lot of wrong answers or wrong reasons.
Q1 and Q2

General Feedback:
• Among 180 students, 86 percent students answered question 1 and 14 percent students answered question 2.
• The average mark for Part 1 is 58.1% (i.e., little less 12 marks out of 20), in which the average mark for question 1 is 59% whereas the average mark for question 2 is 53%. Therefore question 2 is more difficult for the students, as less students selected it and the average mark is lower;
• 32 or 17.8% students received a mark of less than 40% (i.e., 7.5 marks or less). These students seem either having done little to prepare for the exam or having poor understanding about the teaching materials;
• 66 or 36% students received a mark of 70% or better (i.e., 14 marks or more).
• General speaking, the students’ performance in Part 1 is satisfactory from the teaching outcome point of view. In particular, In particular, it is very encouraged that 36% students achieved the 1st class marks.

Detailed Feedbacks for Question 1:
• Question a). The question is about what are the three sub-problems in the robot localization problem. As the robot localization problem is to decide the (probability) location of the robot, the three sub-problems are 1) How to determine the initial probabilities of the robot locating at each pose? 2) When a new observation is obtained by sensors, how to update the probabilities of the robot locating at each pose? 3) When a new action is attempted by the robot, how to update the probabilities of the robot locating at each pose. More than half students answered the three uncertainties in the robot location, which is a question asked in the exam paper of the last year.
• Question b). The question is that, Use the definition of probability distribution to prove that if events . More than half students failed to answer this question correctly or completely. In fact, this is one of 2010 exam questions and obviously many students did not test themselves with 2010 exam paper when they prepared for the exam. The key step in the proof is that ) and , then apply K2 condition to have.
• Question c). Most students answered this question correctly. The common mistake from a small number of students are: firstly some forgot to take into account that there is an obstacle in the arena and led to incorrect probability; secondly some assumed that the robot is equally likely to be in each possible position and orientation whereas the only assumption in the question is that robot is equally likely to be in each possible pose. In other words, these students got their results using some assumption which is not given.
• Question d). There are two sub-questions i) and ii) in this question:
  o Most students answered d.i) correctly. The common mistake by the small number of students are mis-calculation;
  o More than half students answered d.ii) correctly. The common mistake by the remaining students is using the wrong formula. The correct formula which should be used is the extended total probability formula but Bayes theorem was used instead. Another mistake by a small number of students are mis-calculation.
  o As both formulas needed in answering the above question have been used in Lab exercise, this may be the reason why this question was answered well.
• Question e). This is a proof question and was designed to be more difficult. Most students got one or two marks from this question, which showed that they had some idea to deal with the question. But only a small number of students receive the full mark in this question. To answer this question correctly, you need to use condition K2 in the definition of probability distribution first to show that and substitute in the above formula.

Detailed Feedbacks for Question 2:
• Question a). The question is to ask what is Turing test and what is the purpose of the test. Most students answered this question correctly. The common mistake by a small number of students is that they did not answer what purpose of the test is.
• Question b). This question is to prove the extended total probability formula. In fact, this is a question tried by students in an example class and the intention for this question is to check the outcome of the example classes. Unfortunately more than half students failed to answer this question. With this question in the exam paper, it is hoped that the students will take the example classes more seriously in future.
• Question c). There are two sub-questions c.i) and c.ii). Most students answered c.i) correctly but most failed to answer question c.ii) which is more difficult. The correct answer is $\text{P}(E|F)$ but most answers given is $\text{P}(E \cap F)$. This shows the conditional probability is not well understood by some students.

• Question d). $2/3$ students answered this question correctly and completely by using Bayes theorem. The other $1/3$ students showed no clues how to answer the question. The difference suggests the different level of preparation for the exam: the well prepared students did very well and the poor prepared students did poorly.

• Question e). Less than half students answered this question correctly and completely. Most students know to use Bayes theorem but failed to calculate the condition probability correctly, where the event that the car is behind Door 1 and is the event that Monty Hall opens Doors 2 & 3. The correct answer for is $1/3$, as Monty Hall can and only can open doors 2&3, 2&4, and 3&4. As a result, a small number of students get the full 7 marks from this question but most get 2-4 marks.

Section B
Question 3
The question was reasonably well answered, mean = 56.5 ± 18.4 %.
Part a was answered well, it required a little thought, and those that did this achieved a good mark.
Part b required regurgitation from the notes.
Part c required students to estimate probabilities from a truth table and substitute values into Bayes rule. Most managed this, although some students failed the substitution.
Part d was an expansion of part a, whilst some answered were a repletion of part a’s material, and therefore did not attract marks, many successfully expanded on the first answer.

Question 4
Another mostly well-answered question, mean = 61.5 ± 26.5%: most answers achieved high marks although there was a significant tail of marks towards 0 from students who failed in one or more portions of the question.
There was a query from the exam room by a student who failed to realise that the transition probabilities that usually appear on the Markov chain model were instead listed in the question.
Part a required students to list the correct sequence of probability parameters. Most did this, some instead wrote them as $p(txy)$ which received some credit, others invented numbers – wrong.
Part b necessitated students counting the numbers of transitions between all pairs of symbols to complete the table. Then realise that the probability parameters were computed by making each row sum to 1. Apart from adding errors, the typical mistake was to have the table normalised to 1.
Part c was variously interpreted as transitions from states a, b and c to STOP or from START, a, b, c to STOP. Since the question could be interpreted either way, either answer was accepted.
Part d was answered correctly in the majority of cases.
Feedback will be provided from Arcade in the usual way.
Q1 (a) This was very badly done, and there were some very strange answers. Almost nobody answered part (ii) correctly.
(b) Mixed response: some answers were too long - unnecessarily so.
(c) Mixed response. Remember jitter is variation in one-way (not round trip) delay. It is not delay but delay variation.
(d) Quite well done on the whole. TCP is not unsuitable because it it is "too slow", as many people answered. Retransmissions will mostly be too late to be useful, as some answered correctly. The acknowledgements and retransmissions will increase congestion to no purpose for VoIP.
(e) Reasonable answers. RTP is not considered a 'reliable' protocol.

Q2

This question was, on the whole, well answered.

(a) Both parts of this question (a definition of statelessness, and a description of its implications in terms of stability of distributed systems) were generally well answered. Some candidates confused statelessness with some kind of 'security' (i.e. not sending credentials), and lost marks on that basis.

(b) This too was generally well answered. In some cases fairly vague statements about 'web sessions' were made, for which some but not full marks were given (unless a mechanism for creating a session was described).

(c) Curiously the definition of an 'atomic' transaction was answered least well of all the parts of this question, in spite of being essentially a piece of bookwork, and in spite of the fact that many candidates then went on to describe two phase commit quite well. Most candidates made a sensible attempt at describing atomicity, however.

(d) Two Phase Commit was explained well; students were not penalised for creating their own variations of the names of the messages sent during the operation of the algorithm. There was a little confusion about 2PC and deadlock avoidance/detection.

(e) The bully algorithm was generally explained well.

(f) The role and effects of a 'cache' were typically well explained; although the question explicitly asks for examples in the context of a distributed system

(g) some candidates lost a small number of marks by describing CPU-type caches.

Q3

The question was taken by 52 students, the mean was 48.5% but the distribution of marks is not ideal in that about 60% of the students got around 50% or lower.

The question had five parts. Part (a) was answered by the overwhelming majority of students and most did reasonably well. In Part (b) a few students didn't realize that the contrast between functional and non-functional motivations is a technical one, described in the course. This led some students to use the informal sense in which a system is "functional" (meaning that it basically works), which caused marks to be lost. In Part (c) many students did reasonably well. Those that didn't were mostly unable to recall what transparencies were discussed in the course and/or to apply the underlying notions to the scenario given. There weren’t serious, systematic misunderstandings in Part (c), though. Quite a few students failed to answer Part (d), those that did were able to provide mostly good answers. One reason for losing marks was a lack of precise, concrete application of the notion that if the workload is not balanced, the overall time taken will be that of the slowest process to complete. Overall, Part (e) was the one in which more clearly there was a systematic misunderstanding, as follows. Most students that provided an answer were able to spot that without transactional semantics one could have, e.g., a table booked but no commission paid. However, having rightly concluded that a middleware component was needed (viz., a transaction manager), far too many students failed to conclude that this, however, does not imply that a message-oriented middleware is
needed. As a result, many students wrongly concluded that in order to enable transactional semantics one would need to move from direct to mediated message exchange.
Comments: Over the lecture period [as usual] one coursework components was undertaken. The results from this piece of coursework showed the students had a moderately good [to good] understanding of a subset of concepts presented during the lectures. The exam results enforced this view – with a good spread of results. The exam results show the module’s learning outcomes were sufficiently exercised. Both the courses and the lecture learning outcomes (LOs) were tested and the students [as a whole] showed good comprehension of the LOs. The exam and coursework validated the pedagogical approach.
Question 2

2.1 Write a class called Calculator which has an int instance variable to represent the current stored value. The constructor should set that value to zero. It should have a method void add(int other) which adds another value to the one stored, and a method void display() which prints the current stored value.

```java
public class Calculator {
    private int _value;

    public Calculator() {
        _value = 0;
    }

    public void add(int other) {
        _value = _value + other;
    }

    public void display() {
        System.out.println(_value);
    }
}
```

Only 4 people attempted this question. Three got it almost all correct, the other person should have tried a different question!

2.2 Now we will change this into a slightly unusual calculator in that it will store several values at once. Replace the int instance variable with an array of ints. Have the constructor take the required size of the array as its parameter, and create the array. Hint: when you create an array of ints, the values in it are all zero by default, so you don't need to set them. Change the add() and display() methods so that each takes another int parameter which says which value in the array to add to or display. You don't need to check that the parameter is valid.

```java
public class Calculator {
    private int[] _values;

    public Calculator(int size) {
        _values = new int[size];
    }

    public void add(int other, int index) {
        _values[index] = _values[index] + other;
    }

    public void display(int index) {
        System.out.println(_values[index]);
    }
}
```

Only two people attempted this part. So I don't understand what the other three were doing. One got it almost spot on, the other tried to use a loop in add() where only simple indexing is required.

Write a method int maximum() which will return the maximum value stored in the array. You can assume that this value will not be negative.
public int maximum() {
    int max = 0;
    int i = 0;
    while (i < _values.length) {
        if (_values[i] > max) {
            max = _values[i];
        } else {
            // Do nothing
        }
        i++;
    }
    return max;
}

or, with a for-each loop:

public int maximum() {
    int max = 0;
    for (int value : _values) {
        if (value > max) {
            max = value;
        } else {
            // Do nothing
        }
    }
    return max;
}

This is similar to an example in the lecture notes and also one of the methods in the inventory class, although since it just involves ints, it’s simpler.

Both serious attempts were near enough to get full marks.

Question 3

This was the most popular question with 27/30 people attempting it.

3.1 Explain how inheritance in programming languages is related to the way we manage complexity in the real world, and what its main advantages are.

Bookwork (see week 7 lecture notes and slides).

Answers to this were generally good. Most people talked about concept hierarchies and gave examples (Plant -> Tree -> Oak tree was very popular). Generally where people lost marks it was on the second part. A lot of people mentioned avoiding code duplication, but the other, more important advantage - that it allows us to structure applications in a logical easy to understand way, was missing from many answers.

3.2 Draw a UML class diagram which shows an abstract class TimeDisplayDevice, with subclasses Clock and Watch.

About half the answers were fully correct, like the one shown below.

Most of the others had the same shape but incorrect notation (having the arrows the wrong way round or the wrong kind of arrows). These got 2/3.

3.3 Show how the three classes would be declared in Java.

Public abstract class TimeDisplayDevice () { … }
public class Clock extends TimeDisplayDevice { … }
public class Watch extends TimeDisplayDevice { … }

(The {…} bit is optional).

Most people got this right, or very nearly right, which was pleasing. One person gave a
difference three classes, and another wrote a random bit of unrelated code. There’s no point doing this, if you don’t answer the question asked you don’t get any marks.

3.3 Write the TimeDisplayDevice class, assuming that:
- It represents the time with two ints, for the hours and minutes. Initial values for these are provided as parameters to the constructor
- There are public methods to get the hours and minutes
- There is a method to display the time as a String. This method will be implemented differently in each subclass.

```java
public abstract class TimeDisplayDevice {
    private int _hours, _minutes;
    public TimeDisplayDevice(int hours, int minutes) {
        _hours = hours;
        _minutes = minutes;
    }
    public String getHours() {
        return _hours;
    }
    public int getMinutes() {
        return _minutes;
    }
    public abstract String displayTime();
}
```

I was very pleased with the answers to this. Several people got close enough for full marks or at least 4/5.

Question 4

Only 5 people attempted this question

4.1 State TWO ways in which an ArrayList is similar to an array
Both are ordered sequences of values
Both allow the value at an index to be retrieved directly and efficiently
(Both are indexed from 0 was also accepted).

4.2 State FOUR ways in which an ArrayList is different from an array
ArrayLists can change their size after they're created
Arrays can contain primitive types as well as objects
Arraylist is a library class, arrays are built in
Arrays have special syntax, Arraylists just use message sends
(other differences are also acceptable answers).

4.3 Assume you have a Train class, and each Train has a unique ID and a current position, accessed by the methods String getID() and Position getCurrentPosition(). Briefly explain what the following method does.
```java
public Position find(Train[] trains, String trainID) {
    Position result = null;
    int i = 0;
    while(i < trains.length) {
        if (trains[i].getID().equals(trainID)) {
            result = trains[i].getCurrentPosition();
        }
        else {
            // Not found yet, do nothing
            i++;
        }
    }
    return result;
}
```

It finds the train with the given ID, if any, by running through an array of trains, and comparing the ID of each with the required ID. If a match is found the variable result is
the position of the train in question, which is returned at the end.

A more detailed, step by step explanation was also ok, provided that it included the information above.

4.4 If instead, the condition was written
if (trains[i].getID() == trainID)
it would not work correctly. Explain why not.

This compares for identity - whether the two ID strings are the same object, not equality - whether they contain the same characters, and so won't work unless the happen by coincidence to be identical.

This was a tricky one - you'd really have to be paying attention at the right point in one of the early lectures - and nobody got it right. Only one mark, though.

4.5 Explain how, by using a suitable collection, we could avoid using a loop altogether. (It's not necessary to write the actual code so long as the explanation is clear). Hint: an ArrayList is not a suitable collection for this job.

By using a Map<String, Train>
where the String is the trainID. Then we could use containsKey() to check that the ID matches and actual train, and get() to get the corresponding Train object so we can ask it its position. This would be far more efficient if we had lots of trains.

One person got this completely right, and a couple of others got the idea of using a Map.

Question 5
This was, unsurprisingly, a popular question done by 24 people.

5.1 Give a brief outline of how, in a Java application, you would store data about trains, such as owner, capacity, position etc. using each of the following formats.

5.1a Java serialised objects
See week 10 lecture notes, p3, and also the mention of object input and output streams on at the top of p6.

A lot of people failed to say two things. A number of people mentioned that serialization is not appropriate for long-term storage, which strictly speaking doesn't answer the question but I allowed it.

5.1b Comma-separated value (CSV) files.

See week 10 lecture notes p6, and the Student class example at the end of those notes.

Most people got this right. A couple confused CSV with serialization.

5.1c XML files

See week 10 lecture notes pp3-5
Answers to this were a mixed bag. It was disappointing that a lot of people seemed to only have a vague idea of what XML is.

5.1d A relational database

See week 10 lecture notes, p2
I was fairly strict on this one as you already knew about relational databases and the question is specifically about Java. So I gave both marks only to answers which mentioned JDBC (or in a couple of cases a slightly different acronym but it was clear what was meant.

5.2 Suppose the reason you are storing this information is that you are building an information system for Network Rail. Which TWO formats would you seriously consider using, and why

Since this is a large and complex system, only a relational database or XML are worth considering.

A database would enable us to store the data securely, and query it in lots of different ways, as well as supporting transactions etc. XML would make it easy to present the information in different ways, and may help with dealing with data in different formats (held by different train companies for instance). If I was implementing this for real I'd probably use a database as the main means of storage, but XML might well also have a role

Almost everybody chose and justified a RDMS. Quite a number of people picked
serialization, which I might have allowed if a specific plausible role for it was given, but it wasn’t.

5.3 Suppose the reason you are storing this information is that the trains are not real ones (phew!) but models owned by enthusiasts, each of whom will have at most a few tens of them, collected over a period of years. Which TWO formats would you seriously consider using, and why?

In this case a database would be overkill, requiring the enthusiasts to have access to a database server. Serialisation isn’t really appropriate for long-term data storage. XML is a contender, as it is supported directly by Java, and is very flexible. CSV could also be considered, as even XML might well be overkill! The choice between XML and CSV would be made on the basis of how complicated the information to be stored was (e.g. if it included a detailed breakdown of the components of each train, XML would cope better than CSV). For this simple application, it’s unlikely that more than one form of data storage would be necessary.

A number of people suggested serialization, sometimes in combination with CSV. As mentioned above having two storage formats would be overcomplicating things. The problem with serialization is that over a period of years the data format almost certainly would change (as a result of requests for enhancements to the software from the users).