MA/CS 109 Practice Final Exam Solutions (CS Part only!)

Fall 2017

The CS part of the Final Exam will consist of True/False, Computation (as in the homeworks), Short Answer, and an Essay (which will be given to you ahead of time). You will have some choice in which questions to answer.

This practice exam will cover the first three kinds of questions; please answer the following questions, and then we will discuss the solutions.

Problem One (Manditory) True/False

Write True or False to the left of each statement.

1. The ASCII code can represent over a million different symbols. FALSE: ASCII is an 8 bit code, so can only represent $2^8 = 256$ symbols.

2. Run-Length Encoding is an example of a lossy compression method. FALSE: It records the exact sequence in a shorter sequence of symbols by counting repeated symbols.

3. The thesis of "Strong AI" is that an appropriately programmed computer can exhibit life-like qualities such as growth and evolution. FALSE: The Strong AI thesis *per se* says nothing about growth and evolution.

4. If we add 1111 + 101 we get a binary number whose decimal equivalent is 22. FALSE: You get 10100 = 20 in decimal. Alternately, you can see that the original problem is adding 15 + 5.

5. Huffman Encoding is an example of a lossless compression method. TRUE: Each letter is given a distinct binary code, so no information is lost.

6. The ratio between successive Fibonacci Numbers approaches the Golden Ratio (1 : 1.618...). TRUE

7. Kurt Godel proved that some mathematical statements are true but can never be proved so by mathematics. TRUE

8. In the Turing Test, a person attempts to figure out which of two hidden people is a woman and which is a man. FALSE: The Turing Test (called by Turing the Imitation Game) attempts to figure out which is a human and which a machine.

9. To specify 32 different possibilities, we would need at least 6 bits. FALSE: You only need 5 bits, since $2^5 = 32$.

10. This statement is true. ☺ Just for fun, I won't ask you such a trick question on the exam! It could be true or false! If it is true, then, well, it is true; if it is false, then the statement is not true, hence the statement is false. Either works!

Problem Two (CS -- Calculation)

This question concerns Error-Detecting and Error-Correcting codes.

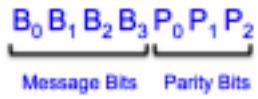
(A) If you get a message "11101011" using the first error-detecting ("even parity") coding technique we studied, can you absolutely claim that no error occurred? Why or why not?

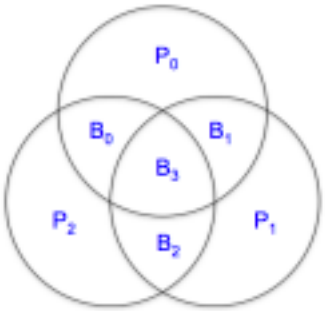
Solution: No! The string may have had 2 errors and still be in even parity. In general, an even number of errors will not be detected.

(B) If you get a message "11101010" using the first error-detecting ("even parity") coding technique we studied, can you absolutely claim that an error occurred? Why or why not?

Solution: Yes! The string has odd party, which can only happen if there has been an error (or an odd number of errors).

(C) Suppose you get a message "1001101" encoded using the Error-Correcting Code that we studied (diagram to right). Has there been an error? If so, correct the error and show the corrected message.





Solution: All the parity circles have odd parity, so the error must have been B_3. So the correct message is 1000101.

(D) Suppose you use this Error-Correcting Code to send the message 1000101, but along the way **every** bit gets flipped, that is, there are 7 errors! What will happen when the receiver decodes the message? Will he understand that the message has errors? Explain carefully!

Solution: All the parity circles still have even parity, so an error would not be reported!

Problem Three (CS – Short Answer)

(A) Suppose you are presented with an "investment opportunity" for a new start- up, the QuickTravelingSalesman company, whose main product is a software program that they claim can solve the Traveling Salesman problem for **any** N cities in N^2 seconds. Should you invest? Explain why or why not.

NO! The Traveling Salesman Problem is a well-known hard problem for which only exponential-time solutions are known. It would take centuries to solve for relatively small N. There are no known N^2 and therefore this company is either lying or confused. In either case, not a good investment!

(B) Describe briefly, using words and diagrams, the idea of Public Key Cryptography.

Public Key Cryptography is based on the existence of problems (like the Partition Problem, or the Traveling Salesman Problem) which are exponentially hard to solve, but it is fast and easy to check a solution for correctness. If a bank uses PKC to protect your account, you and the bank (and anyone else) have an instance of the problem, but to gain access to the account, you must provide a solution, which the bank can check. The point is that if someone hacks into the bank's computer, there is no password to steal!

(C) If we decrease the "bit depth" (the number of bits used to store a pixel) of a picture, what effect might this have on the picture?

The effect would be to reduce the number of possible colors that can be represented, so in addition to having a picture with less variety of colors, you will get "blotchy" patches where a formerly smooth transition between closely-related colors becomes a visible jump between two different colors.

(D) In which situation would a Huffman encoding do a much better job of compressing a text file, compared with Run Length Encoding? Explain your answer carefully.

The RLE does a good job when there are sequences of exactly repeated numbers (or colors, or letters, etc.), and the Huffman encoding does a good job when the probability of some letters is not evenly distributed, and some numbers occur with much higher frequency than others. RLE tends to do a good job with media files (sound, pictures, video) but not so good for text files; the Huffman code can do a good job with any kind of file depending on the probability of the letters or numbers.