Institut for Matematik og Datalogi — August 31, 2010
Syddansk Universitet — JFB

Introduction to Computer Science E10 – Lectures 1 and 2

Textbook

Computer Science: An Overview, 10th Edition, by J. Glenn Brookshear, 2009. Note that there are two versions of the 10th edition of this textbook. It appears to me that the only difference is that in one the real textbook starts on page 1, while in the other it starts on page 19. The page numbers I will give on these notes will refer to the version that starts on page 19. If you have the other, please subtract 18 from every page number mentioned. The first time this happens, I specify both page numbers, so you can tell which version you have.

The textbook will be supplemented with notes.

Format

The course is being taught by Joan Boyar You will meet some of the other faculty members in the department during this course, since they will also give some lectures.

The weekly notes and other information about the course are available through the Worldwide Web. Use the URL:

http://www.imada.sdu.dk/∼joan/intro/

You are responsible for finding all weekly notes there (or in Blackboard) yourself. Please read the appropriate sections in the textbook or notes before coming to class and bring your textbook with you. Preparing for discussion sections (and labs) is important.

The weekly notes will generally contain information about upcoming lectures, problems for the discussion sections, and the assignments.

The discussion sections will sometimes be problems sessions and sometimes labs. Labs will be held in IMADA's Terminal Room. The "instruktor" for the first seven weeks (the first quarter) will be Arun Vadiveal, and the "instruktor" for the remainder (the second quarter) will be Uffe Thorsen. They will be teaching the discussion sections (problem sessions and labs). The first problem session will be Thursday, September 2, at 14:15 in U28. The first lab will be on Monday, September 6.

The course will be graded on a Pass/Fail basis, and satisfactory completion of all 10 assignments is required to pass. "Satisfactory completion" means that the answers are correct, with only very minor errors, and that they have been turned in on time.

You are allowed to retry (at most once) on at most 3 assignments which were not approved. If you turn in an assignment late (regardless of the reason), it will not be approved, and you will have to use a retry on it. These 10 assignments count as the exam in the course, so cheating on these assignments is viewed as cheating on an exam. You are allowed to talk about course material with your fellow students, but working together on assignments is cheating. If you have questions about the assignment, come to me (Joan Boyar) or your "instruktor".

You should turn in your assignments via Blackboard. Note that in the upper left hand corner of the screen, there is an icon which you can click on to expand the the menu for the course. It is just to the left of the code for and name of the course. The assignment hand-in is under "Tools". Keep the receipt it gives you proving that you turned your assignment in on time. Blackboard will not allow you to turn in an assignment late. Note that retries must be turned into me directly and must include both the old version which was not approved (including comments from being graded) and two copies of the retry.

I have office hours on Wednesdays and Fridays from 9:00 to 9:45, but am also available at other times.

Lecture, September 1, 10:15–12, U26

We will begin with an introduction to the course, covering chapter 0 in the textbook, but skipping section 0.2. We will also begin on chapter 1.

Lecture, September 1, 12:15–14, U28

There will be a brief introduction to LaTeX. We will also cover more of chapter 1 in the textbook.

The textbook's interpretation of the mantissa in floating-point representations is not the same as the IEEE-standard and hence somewhat outdated: The book says that the mantissa 1010 means 0.1010 and that the first bit is always 1 in normalized numbers. IEEE-standard says that 1010 means 1.1010, meaning that the fixed normalization bit is a "hidden bit" or "implicit bit" before the radix point. In calculating the value represented by the mantissa, an extra 1 is added. This way the first bit in the mantissa may be 0. Notes about the IEEE standard can be found at http://steve.hollasch.net/cgindex/coding/ieeefloat.html. (For problems in this course, we will use the format described in the textbook, using the same number of bits, but the mantissa will have this IEEE-standard form, with the implicit bit.)

Lecture, September 6, 8:15–10, U37

We will finish chapters 1 and 2.

Discussion section: September 2, 14:15–16, U28

Discussion in groups:

1. Divide into groups of four (or three) people. One person will choose five cards to give to the first "performer", the first performer will give four of them to the second "performer", one at a time, and the second performer will announce what the fifth card is. Each person should practice each "performer" part at least three times.

2. Discuss (in your groups) various methods for improving the magic trick. For example, if the first card determines the suit, after seeing the trick repeated several times, the audience might find it easier to guess how it is done. Define an algorithm for a modification of the trick which makes this harder to see.

3. Discuss how to do a magic trick, where one performer is thinking of a number between 1 and 24, tells the audience the number, gets some

cards from the audience, and passes some cards to the second performer, who announces the number.

4. Discuss how to extend this to larger numbers than 24. When might it be better to only use the color on the card (whether it is red or black, but also consider using the four different suits), rather than some permutation (ordering) of the cards?

5. Find a “bad” pair of integers for the greatest common divisor algorithm, where a pair is bad if the algorithms must perform a lot of steps relative to how large the numbers are. (One expects more steps for larger numbers.)

6. Do problem 3 in the Chapter Review Problems for Chapter 1 (page 73 or 91).

7. Design a circuit, using only AND, OR and NOT gates which takes three bits as input and outputs a 1 if the input has at least two zeros, and a 0 otherwise.

8. Design and draw a circuit containing only AND and XOR gates (with at most two inputs) which takes three bits as input and outputs a 1 if the input has at least two ones, and a 0 otherwise.

   (In the student resources for the course textbook, under the Activities for Chapter 1, there is a simulator for logic circuits which you could use to check your circuit. It is time consuming to use, though.) As an extra challenge, try to do it so that there is only one AND gate, though more XOR gates. (Minimizing the number of AND gates can be useful in some cryptographic applications.)

9. Discuss questions 2 and 4 on pages 34–35 of the textbook.

Laboratory: September 6, 14:15-16, IMADA’s terminal room

Meet in IMADA’s terminal room with your login information. Work in groups of size 2 (maybe one of size 3). This lab is about LaTeX. Look at the notes written by Torben Nielsen and Arun Vadiveal on LaTeX on the homepage for the course: http://imada.sdu.dk/∼joan/intro/latexbook.ps There are also two other useful links about LaTeX available on the course's homepage.

Assignment due 12:15, September 8

Late assignments will not be accepted. Working together is not allowed. (You may write this either in English or Danish, but write clearly if you do it by hand.) Submit a PDF file through the Blackboard system. If you have written it by hand, you will need to scan it. (Ask for help early if you need it for submitting.)

1. Write down the algorithms for the "magic trick" described in class, where one performer passes four of five playing cards to the other, so that the other can tell what the fifth card is. (If you were not in class or at the discussion section where this was presented, ask me or your instruktor to show you the trick.) There should be one algorithm for each of the two performers. Write these as algorithms, so that all steps are clearly specified, without ambiguity.

2. Either do the first two problems below or the third one. The third one is somewhat more challenging. As with all problems in this course, explain your results.

   (a) Design and draw a circuit containing only AND, OR and NOT gates (each gate having at most two inputs) which takes three bits as input and outputs a 1 if the input is 100, 011 or 110, and a 0 otherwise. (In the student resources for the course textbook, under the Activities for Chapter 1, there is a simulator for logic circuits which you could use to check your circuit. It is time consuming to use, though.)

   (b) Design and draw a circuit containing only AND, OR and NOT gates (each gate having at most two inputs) which takes four bits as input and outputs a 1 if the input is 1000, 0111 or 1110, and a 0 otherwise.

   (c) Design and draw a circuit containing only AND and XOR gates (each gate having two inputs) which takes six bits as input and outputs a 1 if the input has at least four ones, and a 0 otherwise.

Use only four AND gates. How many XOR gates do you need? Hint: Look at (and use) the problem from the discussion section where you were asked to minimize the number of AND gates. Then consider adding two numbers, each of which has two bits.