

Introduction to Course and Housekeeping Items

Websites that pertain to our course

1. http://wiki.math.yorku.ca/index.php/Math_5300
2. “Math 5300 Computation in Mathematics for Teachers” Group on Facebook
3. Your own blog website.
4. <http://www.flickr.com/> and the account you set up on it.

Mark breakdown

Fifty percent of our final grade is determined by the following percentages from work we do with Mike Zabrocki. The other fifty percent will come from work we do with Hongmei Zhu. Zabrocki’s breakdown is as follows.

<i>Percentage of 50%</i>	<i>Component and description (taken from the wiki website)</i>
25%	Class participation: This includes attendance. If you miss a class you are missing the equivalent of one full week of class time in a normal year long course. Your assignments must be on time and dedication (or lack thereof) to this class will be taken into consideration for this component.
15%	Class summary: Each of us will be asked to organize a page on the wiki which has class notes and material and a summary of what we did for one day of class. This includes supplementary material we wish to add (say, for instance, a link to a “Youtube” video or website or a fellow student’s blogs, etc.). We should be doing a good job on this piece because we are only responsible for one day.
40%	Homework assignments: Zabrocki doesn’t want any paper homework assignments so these will be posted on our blogs. He expects us to work together and help out our fellow students and read other assignments for ideas.
20%	Final project: This project is only half the course so it will only use some of the basic components of image processing that we will learn.

Some History on Computation¹

- This is a topic that could take a whole course (and then some!)
- Definition: “Computation is a general term for any type of information processing that can be represented mathematically. This includes phenomena ranging from human thinking to calculations with a more narrow meaning. Computation is a process following a well-defined model that is understood and can be expressed in an algorithm, protocol, network topology, etc.”²
- We start off with a word with which we are familiar: algorithm
 - It is an abstract concept
 - It describes a recipe/procedure for completing a task
 - The word can be used daily!
- We didn’t have computers 60-100 years ago
- In 1936, the question arose, “What do we mean by a computer?” and “What does it mean to do a computation?”
- In 1936, the beginning of WWII, we were able to define what is “true.”
- The reason for this was because we needed computers to listen in on others’ conversations during WWII.
- Because messages were able to be intercepted, we had to be able to encode *and* decode. Therefore, we needed computation to decode. This implied the need for computers.
- So, we asked, “How do we decide what is true?”
- The following people answered this question:³

Alonzo Church	Alan Turing
<ul style="list-style-type: none">• (June 14, 1903 – August 11, 1995)• An American mathematician and logician who was responsible for some of the foundations of theoretical computer science	<ul style="list-style-type: none">• (23 June 1912–7 June 1954)• An English mathematician, logician, and cryptographer.• broke codes during WWII• often considered to be the father of modern computer science

¹ An interesting website (with a lot to read on it) is <http://www.csc.liv.ac.uk/~ped/teachadmin/histsci/htmlform/lect1.html>. The author, Paul E. Dunne, has his Ph.D in Computer Science and works at the University of Liverpool (at least, he was in 2006). On this website, is posted six lectures on the history of science and the development of algorithms.

² Definition taken from <http://en.wikipedia.org/wiki/Computation>.

³ All information in the table is taken from Zabrocki’s lecture and searched through http://en.wikipedia.org/wiki/Main_Page.

<ul style="list-style-type: none">• Responsible for lambda-calculus:• In mathematical logic and computer science, lambda calculus, also λ-calculus, is a formal system designed to investigate function definition, function application and recursion.	<ul style="list-style-type: none">• Responsible for the Turing machine:• Turing provided an influential formalisation of the concept of the algorithm and computation with the Turing machine.
<p>They each were able to define the term “algorithm.” It was shown that these two perspectives on what an algorithm is were equivalent. Together, they came up with the Church-Turing Thesis.</p> <p>Informally, “the Church–Turing thesis states that if an algorithm (a procedure that terminates) exists then there is an equivalent Turing Machine or applicable λ-function for that algorithm”.</p>	

More on Turing⁴

- Turing wanted to create a machine that would include:
 - States
 - An alphabet
 - Transitions between states
 - Example: “start at a state, read an alphabet” → “go to another state, write alphabet, move left or right”
 - Special substates of all the states (i.e., start state, accept state, and reject state).
 - Example: see a picture of a Turing machine.^{5,6}
 - Example: an infinite tape
- Thus, a Turing machine can be defined as a “finite-state machine associated with an external storage or memory medium.” (Minsky (1967), p. 117)⁷
- With respect to the infinite tape, we can say:

“The concept of the **Turing machine** is based on the idea of a person executing a well-defined procedure by changing the contents of an unlimited paper tape,

⁴ Someone in class mentioned a film loosely based on Turing’s accomplishments, but not about Turing himself. I believe the film is [Enigma](http://en.wikipedia.org/wiki/Enigma_%282001_film%29) and was made in 2001. More detail on the film’s plot can be found at http://en.wikipedia.org/wiki/Enigma_%282001_film%29. A link that describes Alan Turing as UK-USA link, from 1942 and onwards is <http://www.turing.org.uk/turing/scrapbook/ukusa.html>

⁵ http://en.wikipedia.org/wiki/Turing_machine_examples

⁶ For the Turing machine drawn in class, see <http://wiki.math.yorku.ca/index.php/Class2>.

⁷ Definition gotten from http://en.wikipedia.org/wiki/Turing_machine#Formal_definition_of_single-tape_Turing_machine.

which is divided into squares, where each square contains one of a finite set of symbols. The person needs to remember one of a finite set of states and the procedure is formulated in very basic steps in the form of ‘If your state is 42 and the symbol you see is a “0” then replace this with a “1,” move one symbol to the right, and assume state 17 as your new state.’⁸

- In theory, every machine accepts/rejects and inputs.
- A computer is just changing/manipulating bits
- A computer computation is just 0’s, 1’s and $_$ ’s
- Now that we have an idea of a language and system at work, we ask, “So, how do we manipulate bits?” (where bits are 0’s and 1’s).
- bit = 1 or 0 = true or false
- 1 byte = 8 bits
- 1 kilobyte = 1 KB = 1024 bytes = 2^{10} bytes
- 1 megabyte = 1 MB = 1024 KB
- 1 gigabyte = 1 GB = 1024 MB (like a hard drive)
- 1 terabyte = 1 TB = 1024 GB (Google handles a few TB a day!)

- One character on a keyboard corresponds to one byte (= 8 bits)
- This byte is a different sequence of eight 1’s and 0’s (for reference see the ASCII table that stands for “American Standard Code for Information Interchange”. It is a table that represents all characters. Since a computer can only understand numbers, the ASCII code is the numerical representation of characters such as “a” or “@” for example.)⁹
- Example: A = 65, B = 66, ... and then you would convert these numbers into binary code
- There are 127 codes. Different languages have different tables with different codes.

Possible Outputs

0 0 0 1 → and
 0 1 1 1 → or
 1 1 0 1 → implies

Example of “implies”:

X	Y	X IFF Y	X IMPL Y	Y IMPL X
0	0	1	1	1

⁸ Quote gotten from http://en.wikipedia.org/wiki/Turing_machine#Formal_definition_of_single-tape_Turing_machine.

⁹ For more information and to see the table see <http://www.asciitable.com/>.

0	1	0	1	0
1	0	0	0	1
1	1	1	1	1

- Microchips were invented(?) by the Japanese
- They made vacuum tubes that acted like gates.
- They then realized that a microchip/resistor would do the same job.
- See examples of gates on page 5.

Examples of gates (and truth tables, De Morgan equivalents, and Venn diagrams).¹⁰

	y	
∧	0	1
x	0	0
	1	0

	y	
∨	0	1
x	0	1
	1	1

	y	
→	0	1
x	0	1
	1	0

	y	
⊕	0	1
x	0	1
	1	0

Figure 1. Truth tables

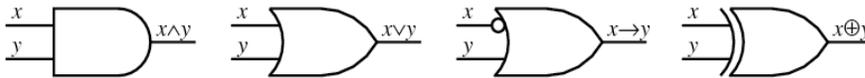


Figure 2. Logic gates

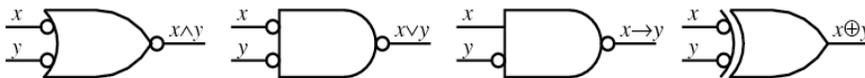


Figure 3. De Morgan equivalents



Figure 4. Venn diagrams

¹⁰ Picture gotten from http://en.wikipedia.org/wiki/Boolean_algebra_%28logic%29.