Theory of Computation

Last updated 1/12/20

What is this course about?

- This course is about the fundamental capabilities and limitations of computers/computation
- This course covers 3 areas, which make up the theory of computation:
  - Automata and Languages
  - Computability Theory
  - Complexity Theory

Automata and Languages

- Introduces models of computation
  - We will study finite automata and context free grammars
  - Each model determines what can be expressed, as we will see in Part I of this course
  - Will allow us to become familiar with simple models before we move on to more complex models like a Turing machine
  - Given a model, we can examine computability and complexity

Computability Theory

- A major mathematical discovery in 1930s
  - Certain problems cannot be solved by computers
    - That is, they have no algorithmic solution
- We can ask what a model can and can't do
  - As it turns out, a simple model of a computer, A Turing machine, can do everything that a computer can do
  - So we can use a Turing machine to determine what a computer can and can't do (i.e., compute)

Complexity Theory

- How hard is a problem?
- You already should know a lot about this
  - You should know how to determine the time complexity of most simple algorithms
    - You should know the Big O notation. We can say that a problem is $O(n^2)$ and that is harder than an O(n) problem
- We take one step forward and study NP-completeness
  - A first course in theory of computation generally stops there and hence we will not cover Chapters 8, 9, or 10 of the text

About this Course

- Theory of Computation traditionally considered challenging
  - I expect (and hope) that you will find this to be true!
- A very different kind of course
  - In many ways, a pure theory course
    - But very grounded (the models of computation are not abstract at all)
  - Proofs are an integral part of the course, although I and the text both rely on informal proofs
    - But the reasoning must still be clear
- The only way to learn this material is by doing problems
  - You should expect to spend several hours per week on homework
  - You should expect to read parts of the text 2-4 times
  - You should not give up after 5 minutes if you are stumped by a problem
  - The best way to prepare for the exams is to put significant effort into the homework

Mathematical Preliminaries
Chapter 0

A review of discrete math with some new material

Mathematical Preliminaries

- We will now very quickly review discrete math
  - You should know most of this from 1100/1400
  - Reading Chapter 0 of the text should help you review
    - Ask me for help if you have trouble with some of this
  - Mathematical Notation
    - Sets
    - Sequences and Tuples
    - Functions and Relations
    - Graphs
    - Strings and Languages (not covered previously)
    - Boolean Logic
  - Proofs and Types of Proofs

Sets

- A set is a group of objects, order doesn't matter
  - The objects are called elements or members
  - Examples:
    - $\{1, 3, 5\}$, $\{1, 3, 5, \ldots\}$, or $\{x | x \in Z \text{ and } x \bmod 2 \neq 0\}$
  - You should know these operators/concepts
    - Subset ($A \subset B$ or $A \subseteq B$)
    - Cardinality: Number elements in set ($|A|$ or $n(A)$)
    - Intersection ($\cap$) and Union ($\cup$), Complement
      - What do we need to know to determine complement of set A?
    - Venn Diagrams: can be used to visualize sets

Sets II

- Power Set: All possible subsets of a set
  - If A = {0, 1} then what is P(A)?
  - In general, what is the cardinality of P(B)?

Sequences and Tuples

- A sequence is a list of objects, order matters
  - Example: (1, 3, 5) or (5, 3, 1)
- In this course we will use term tuple instead
  - (1, 3, 5) is a 3-tuple and a *k*-tuple has *k* elements

Sequences and Tuples II

- Cartesian product (X) is an operation on sets but yields a set of tuples
  - Example: if A = {1, 2} and B = {x, y, z}
    - A x B = {(1,x), (1,y), (1,z), (2,x), (2,y), (2,z)}
  - If we have k sets A_1, A_2, …, A_k, we can take the Cartesian product A_1 x A_2 … x A_k which is the set of all k-tuples (a_1, a_2, …, a_k) where $a_i \in A_i$
  - We can take Cartesian product of a set with itself
    - A^k represents A x A x A … x A where there are k A's.
  - The set Z^2 represents Z x Z all pairs of integers, which can be written as $\{(a,b) \mid a \in Z \text{ and } b \in Z\}$

Functions and Relations

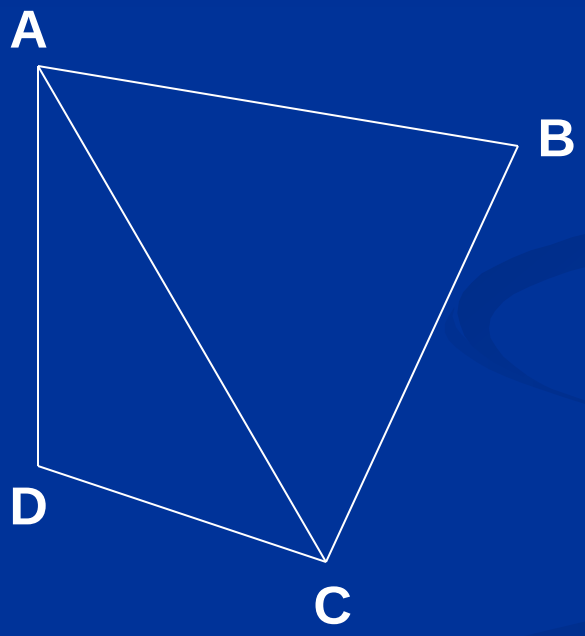
- A function maps an input to a (single) output
  - f(a) = b, f maps a to b
- The set of possible inputs is the domain and the set of possible outputs is the range
  - f: D $\rightarrow$ R
  - Example 1: for the *abs* function, if D = Z, what is R?
  - Example 2: *sum* function
    - Can say Z x Z $\rightarrow$ Z
- Functions can be described using tables
  - Example: Describe f(x) = 2x for D={1,2,3,4}

Relations

- A predicate is a function with range {True, False}
  - Example: even(4) = True
- A (k-ary) relation is a predicate whose domain is a set of k-tuples A x A x A … x A
  - If k =2 then binary relation (e.g., =, <, ...)
  - Can just list what is true (even(4))
  - The text represents the *beats* relation in Example 0.10 (page 9) using a table and a set
- Relations have 3 key properties:
  - reflexive, symmetric, transitive
  - A binary relation is an equivalence relation if it has all 3
  - Try =, <, friend

Graphs

- A graph is a set of vertices V and edges E
  - G= (V,E) and can use this to describe a graph



V = {A, B, C, D}
E = {(A,B), (A,C), (C,D), (A,D), (B,C)}

Graphs II

- Definitions:
  - The degree of a vertex is the number of edges touching it
  - A path is a sequence of nodes connected by edges
  - A simple path does not repeat nodes
  - A path is a cycle if it starts and ends at same node
  - A simple cycle repeats only first and last node
  - A graph is a tree if it is connected and has no simple cycles

Strings and Languages

- This is heavily used in this course
- An alphabet is any non-empty finite set
  - Members of the alphabet are alphabet symbols
  - $\Sigma_1 = \{0,1\}$
  - $\Sigma_2 = \{a,b,c,d,e,f,g,h,i,j,k,l,m,n,o,p,q,r,s,t,u,v,w,x,y,z\}$
  - $\Sigma_3 = \{0,1,a,b,c\}$
- A string over an alphabet is a finite sequence of symbols from the alphabet
  - 0100 is a string from Σ_1 and cat is a string from Σ_2

Strings and Languages II

- The length of a string w, |w| is its number of symbols
- The empty string, ε, has length 0
- If w has length n then it can be written as $w_1 w_2 \ldots w_n$, where $w_i \in \Sigma$
- Strings can be concatenated
  - ab is string a concatenated with string b
  - a string x can be concatenated with itself k times
    - This is written as x^k
- A language is a set of strings

Boolean Logic

- Boolean logic is a mathematical system built around True and False or 0 and 1
- Below are the boolean operators, which can be defined by a truth table
  - $\wedge$ (and/conjunction) $1 \wedge 1 = 1$; else 0
  - $\vee$ (or/disjunctions) $0 \vee 0 = 1$; else 1
  - $\neg$ (not) $\neg 1 = 0$ and $\neg 0 = 1$
  - $\rightarrow$ (implication) $1 \rightarrow 0 = 0$; else 1
  - $\leftrightarrow$ (equality) $1 \leftrightarrow 1 = 1$; $0 \leftrightarrow 0 = 1$
- Can prove equality using truth tables
  - DeMorgan's law and Distributive law

Proofs

- Proofs are a big part of this class
  - A proof is a convincing logical argument
    - Proofs in this class need to be clear, but not very formal
      - The books proofs are often informal, using English, so it isn't just that we are being lazy
  - Types of Proofs
    - A $\Leftrightarrow$ B means A if and only if B
      - Prove A $\Rightarrow$ B and prove B $\Rightarrow$ A
    - Proof by counterexample (prove false via an example)
    - Proof by construction (main proof technique we will use)
    - Proof by contradiction
    - Proof by induction

Proofs: Example 1

- Prove for every graph G sum of degrees of all nodes is even
  - Take a minute to prove it or at least convince yourself it is true
  - The book does not really say it, but this is a proof by induction
    - Their informal reasoning: every edge you add touches two vertices and increases the degree of both of these by 1 (i.e., you keep adding 2)
    - See Example 0.19 p18 and Theorem 0.21 p20
  - A proof by induction means showing 1) it is true for some base case and then 2) if true for any *n* then it is true for *n*+1
    - So spend a minute formulating the proof by induction
    - Base case: 0 edges in G means sum-degrees=0, is even
    - Induction step: if sum-degrees even with *n* edges then show even with n+1 edges
      - When you add an edge, it is by definition between two vertices (but can be the same). Each vertex then has its degree increase by 1, or 2 overall
      - even number + 2 = even (we will accept that for now)

Proofs: Example 2

- For any two sets A and B, $(A \cup B)' = A' \cap B'$ (Theorem 0.20, p 20)
  - Where X' means the complement of X
  - We prove sets are equal by showing that they have the same elements
  - What proof technique to use? Any ideas?
  - Prove in each direction:
    - First prove forward direction then backward directions
      - Show if element x is in one of the sets then it is in the other
    - We will do in words, but not as informal as it sounds since we are really using formal definitions of each operator

Proof: Example 2

- $(A \cup B)' = A' \cap B'$
- Forward direction (LHS → RHS):
  - Assume $x \in (A \cup B)'$
  - Then x is not in $(A \cup B)$ [defn. of complement]
  - Then x is not in A and x is not in B [defn. of union]
  - So x is in A' and x is in B' and hence is in RHS
- Backward direction (RHS → LHS)
  - Assume $x \in A' \cap B'$
  - So $x \in A'$ and $x \in B'$ [defn. of intersection]
  - So $x \notin A$ and $x \notin B$ [defn. of complement]
  - So x not in union $(A \cup B)$ [application of union]
  - So x must be its complement [defn. of complement]
- So we are done!

Proofs: Example 3

- For every even number n > 2, there is a 3-regular graph with n nodes (Theorem 0.22, p 21)
  - A graph is k-regular if every node has degree k
- We will use a proof by construction
  - Many theorems say that a specific type of object exists. One way to prove it exists is by constructing it.
  - May sound weird, but this is by far the most common proof technique we will use in this course
    - We may be asked to show that some property is true. We may need to construct a model which makes it clear that this property is true

Proof: Example 3 continued

- Can you construct such a graph for n=4, 6, 8?
  - Try now.
  - If you see a pattern, then generalize it and that is the proof.
  - Hint: place the nodes into a circle
- Solution:
  - Place the nodes in a circle and then connect each node to the ones next to it, which gives us a 2-regular graph.
  - Then connect each node to the one opposite it and you are done. This is guaranteed to work because if the number of nodes is even, the opposite node will always get hit exactly once.
    - The text describes it more formally.
    - Note that if it was odd, this would not work.

Proof: Example 4

- Jack sees Jill, who has come in from outside. Since Jill is not wet he concludes it is not raining (Ex 0.23, p 22)
  - This is a proof by contradiction.
    - To prove a theorem true by contradiction, assume it is false and show that leads to a contradiction
    - In this case, that translates to assume it is raining and look for contradiction
  - If we know that if it were raining then Jill would be wet, we have a contradiction because Jill is not wet.
  - That is the process, although not a very good example (what if she left the umbrella at the door!)
  - This case is perhaps a bit confusing. Lets go to a more mathematical example …

Prove Square Root of 2 Irrational

- Proof by contradiction, assume it is rational
  - Rational numbers can be written as m/n for integer m, n
  - Assume with no loss of generality we reduce the fraction
    - This means that m and n cannot both be even
      - If so, 2 goes into both so reduce it
  - Then do some math
    - $\sqrt{2} = \frac{m}{n}$
    - $n\sqrt{2} = m$
    - $2n^2 = m^2$
    - This means that m^2 is even and thus m must be even
      - Since odd x odd is odd

Prove Square Root of 2 Irrational

- So $2n^2 = m^2$ and m is even
- Any even number can be written as 2k for some integer k, so:
  - $2n^2 = (2k)^2 = 4k^2$ Then divide both sides by 2
  - $n^2 = 2k^2$
  - But now we can say that n^2 is even and hence n must be even
- We just showed that m and n must both be even, but since we started with a reduced fraction, that is a contradiction.
  - Thus it cannot be true that $\sqrt{2}$ is rational

More on Proof by Induction

- Worth going over one more example since some of you may not have used this technique before
- Please ignore Theorem 0.25, p 24 in text which is how the book explains proof by induction
  - complicated induction proof which is not very illustrative
- You have a proof by induction for HW1, so the next example should help

Another Proof by Induction Example

- Prove that $n^2 \geq 2n$ for all n 2, 3, …
- Base case (n=2): $2^2 \geq 2x2$? Yes.
- Assume true for n=m and then show it must also be true for n=m+1
  - So we start with $m^2 \geq 2m$ and assume it is true
  - we must show that this requires $(m+1)^2 \geq 2(m+1)$
    - Rewriting we get: $m^2+2m+1 \geq 2m+2$
    - Simplifying a bit we get: $m^2 \geq 1$.
    - So, we need to show that $m^2 \geq 1$ given that $m^2 \geq 2m$
      - If $2m \geq 1$, then we are done. Is it?
      - Yes, since m itself ≥ 2