CPSC 421/501: Outline, Starting September 6, 2017

Joel Friedman

University of British Columbia
www.math.ubc.ca/∼jf

UBC

September 2017

Course Overview

- Course summary:
  - ☺ We understand pretty well: there are "uncomputatable" problems, including the halting problem.
  - ☹ We don't understand: P versus NP.
- Course material:
  - Our course mainly follows Sipser's textbook.
  - We start with two articles: (1) prerequisites and walk counts, (2) self-referencing.
  - Chapter 1 is a good warm-up for the main material.
  - The main material is about Turing machines and computability (Chapters 3–5, 7–9).

Executive Summary of First Two Weeks

- Directed Graphs and Asymptotic Tests: Prerequisites (or stuff you must quickly learn).
- Directed Graphs and Asymptotic Tests: New material: Counting walks in directed graphs (e.g., Fibonacci numbers), asymptotic tests for walk counts.
- Uncomputability and Self-Referencing (and Paradoxes): Self-referencing is a powerful tool for proving theorems.

Paradoxes and Self-Referencing (2nd Article)

- Self-referencing is a powerful tool for proving theorems and discovering serious fundamental issues with what you are doing.
  1. "This statement is a lie."
  2. "Let S be the sets of sets that don't contain themselves."
  3. "Let n be the smallest integer not described by an English sentence with fewer than one hundred words."
  4. Etc.
- Example of self-referencing (diagonalization) result:
  Theorem: The set of "computer programs" over an alphabet is (infinite but) *less than* the number of "languages" or "decision problems."
  Corollary: There exists a "language" (or a "decision problem") for which there is no corresponding "computer program" (or "algorithm").

Here's What We Don't Want

- Chapter 0: If $|A| = 3$ and $|B| = 4$, what is the largest possible value of $|A \cup B|$?
- Three weeks later: Let M be a Turing machine that invokes a universal Turing machine [specifically a multi-tape machine that can simulate k steps of an arbitrary machine M' in time $O(f(M')k \log k)$ where $f(M') = O(\mathrm{poly}(\langle M' \rangle))$] that preprocesses its input I by computing an injection $\Sigma^* \to \Sigma^*$ whose meaning is to negate... Etc.

This course:

- Review prerequisites, test our knowledge by studying "walks in directed graphs."
- Give some idea of diagonalization and self-referencing.
- Cover Chapter 1 is reasonable detail and sophistication.

Directed Graphs and Asymptotic Tests: Prerequisites

- Prerequisites (or stuff you must quickly learn, Chapter 0)
  - Conventions regarding $\mathbb{N}$, $\mathbb{R}$, $\mathbb{R}^+$, $\mathbb{Z}$, $\lim_{n\to\infty} f(n)$.
  - Big-O, little-o, $\lim_{n\to\infty}$, Θ, $\sim$ (reviewed in Section 7.1 of [Sip]).
  - Basic ideas in set theory, alphabets, words/strings, languages.
  - Recurrence equations (mainly from algorithms).
  - Proofs (by induction, by contradiction, etc.).
- Counting $f(n) \stackrel{\text{def}}{=}$ the number walks of length n there are from a given vertex to another in a fixed directed graph.
- What functions $f(n)$ can never arise as such. (By direct, simple, *asymptotic tests*.)

Discussion Ideas from “Directed Graphs ...”

- $\mathbb{N}, \mathbb{R}, \mathbb{R}^+, \mathbb{Z}$. E.g., $n \mapsto n \log_2 n$ is $\mathbb{N} \to \mathbb{R}$, problematic for [Sip] who uses $\mathbb{N} \to \mathbb{R}^+$ for big-O, little-o.
- big-O, little-o review:
  - Big-O, little-o basic examples. $n^2 + 3n + 20 = n^2 + O(n) = O(n^2)$.
  - $3n + 20 = o(n^2)$ (take limit); so $n^2 + 3n + 20 = n^2 + o(n)$; $n \log_2 n + 20n + 3 = n \log_2 n + o(n \log_2 n)$.
  - Facts: for any $a, \epsilon > 0$: $n^a = o(n^{a+\epsilon})$, $\log n = o(n^\epsilon)$; $a^n = o((a+\epsilon)^n)$; $\epsilon > 0$ connotes a “small” number.
  - Meta-facts: $f(n) = o(g(n)) \Rightarrow f(n) = O(g(n))$. $f_1(n) = O(g_1(n))$ and $f_2(n) = O(g_2(n))$ implies $f_1 f_2 = O(g_1 g_2)$, etc.
  - $n^2 + 3n + 20 = \Theta(n^2)$, since $n^2 \le n^2 + 3n + 20 \le 24n^2$ for all $n \ge 1$.
- Maybe new: Θ and $\sim$.
  - $f(n) = \Theta(g(n))$ means $C_1 g \le f \le C_2 g$ for large n; $f(n) \sim g(n)$ means $\lim_{n\to\infty} f(n)/g(n) = 1$.
  - “Order $f(n)$” can mean $\Theta(f(n))$ or $O(f(n))$: “Linear time algorithm” means “runs in $\Theta(n)$ time”; quadratic $\Theta(n^2)$; cubic $\Theta(n^3)$.
  - Simplify: asymptotic relations are determined by largest term: $5n^2 - 7n + 5$ can be replaced with $5n^2$ for the sake of $O, o, \Theta, \sim$.
  - Stirling’s approximation: $n! \sim (n/e)^n \sqrt{2\pi n}$ or $\sqrt{2\pi n}(n/e)^n \le n! \le \sqrt{2\pi n}(n/e)^n e^{1/(12\,n)}$
- $f(n) = OO(g(n))$
- Proofs by induction, contradition, etc.
- Set notation: $A \cap B$, $|A|$, $A \times B$, etc.
- Alphabet: finite set; string (word) over an alphabet; substring; concatenation; language.
- Graphs, digraphs (directed graphs), vertices, edges, etc.

Discussion Ideas from “Directed Graphs ...”

Use walk counts on directed graphs to review everything and give some new ideas.

- Definition of digraph. Fibonacci graph. Walks.
- Examples.

Many Topics Starting Friday, Sept 8, 2017

- Note: [Sip] uses $\mathcal{R}^+$ to denote $\mathbb{R}_{\geq 0}$, i.e., the non-negative reals.
- Instructions for `gradescope.com` to come.
- Today: $O, o, \Theta, \sim$ and *asymptotic ratio*.
- "Quadratic time," $3n^2 + 3n \sim 3n^2$, $n! \sim \sqrt{2\pi n}(n/e)^n$, $\log(n!) \sim n \log n$.
- Example: $\sin(1)/2 + \sin(2)/4 + \sin(3)/8 + \cdots + \sin(n)/2^n + \cdots$ converges.
- Exercise A.4(1d) $n^2 + 3n + 1 = O(n(n-1))$? Ex. A.5(3) $f \sim g$ iff $\log(f) - \log(g) = o(1)$?
- Walks in directed graphs, walk counting functions.
- Walk counts in a graph where the first vertex has five self-loops, the second has three, and there is one edge from the first to the second (and that's it).