CPSC 421/501 Sept 6, 2023

"In mathematics you don't understand things. You just get used to them."

— John von Neumann

My alternate form:

"In mathematics it takes time for ideas and examples to sink in."

Course website:

https://www.cs.ubc.ca/~jf/courses/421.F2023/index.html

Material: Standard 1-term into to CS Theory. The main point is that certain "problems":

(1) are provably "unsolvable"

e.g. the halting problem is undecidable

(2) may seem approachable, but no one knows how to solve them in polynomial time; if you could do so, then

(1) $P = NP$ (2) You win 10^6 USD

(minus applicable taxes)

e.g: SAT, 3SAT, 3COLOUR, PARTITION, ...

We follow textbook [Sip] + handouts

Discussion: please post to piazza page.

If this fails, please email

To: jf@cs.ubc.ca

Subject: CPSC 421

OR

Subject: CPSC 501

Grading:

421: $(10\%)\max(h,m,f) + (35\%)\max(m,f) + (55\%) f$,

h = homework, m = midterm, f = final

501: $(80\%)\left(\begin{matrix}421\\ \text{grade}\end{matrix}\right) + (20\%)\left(\begin{matrix}\text{Presentation}\\ +\text{ Report}\end{matrix}\right)$

Individual Homework: You must write up your own solution

Group Homework: You can make a group submission, groups ≤ 4 people

---

You will pass (50%) the course if on the final exam you can

1. Write a 100% correct DFA algorithm.
2. " " " " Turing machine " .

100% correct ⇒ you also explain how these algorithms work.

1st 2 weeks:

$\boxed{\text{Uncomputability in CPSC 421/501}}$

- Alphabets, strings, power set, languages, ...

- Cantor's Theorem constructs a "problem" for which there is no "algorithm"

- Proof of Cantor's theorem related to many "paradoxes"

- Halting problem is undecidable

- Use Python algorithms for now, later we use Turing machines

Terminology & Notation: $\mathbb{R}, \mathbb{Z}, \mathbb{N}$

$$\left\{ \begin{array}{l} \text{Alphabet, string, language, power set} \\ \Sigma, \Sigma^k, \Sigma^*, \text{Power}(\Sigma^*), \Sigma_{\text{ASCII}}, \Sigma_{\text{digits}} \\ \text{PRIMES, PALINDROME, DIV-BY-3, } \ldots \end{array} \right\}$$

"Problem" = "Decision Problem" = "Language"

"Algorithms" expressed in Python

(or in C, in Javascript, in APL, as Turing machines, etc.)

[Sip]

$\mathbb{R}$ = real numbers $\mathcal{R}$

$\mathbb{N}$ = natural numbers n

$= \{1, 2, 3, \ldots\}$

$\mathbb{Z}_{\geq 0} = \{0, 1, 2, \ldots\}$

$\mathbb{Z} = \{\ldots, -2, -1, 0, 1, 2, \ldots\}$

(Zahlen)

Alphabet: finite, non-empty set, often use Σ

Σ_{ASCII} = 256 ASCII characters

Often

$$\Sigma = \{a\}$$

or $\{a, b\}$

or $\Sigma_{Digits} = \{0, 1, \dots, 9\}$

Elements of an alphabet are symbols (letters)

$$\Sigma^k = \underbrace{\Sigma \times \dots \times \Sigma}_{k \text{ copies}}$$

$\Sigma = \{a, b\}$, Σ^2

$$\Sigma^2 = \{ (a,a), (a,b), (b,a), (b,b) \}$$

also: $\{ aa, ab, ba, bb \}$

aa is really $(a,a) \in \Sigma^2$

Σ^k $\left\{\begin{matrix}\text{strings}\\ \text{words}\end{matrix}\right\}$ of length k

$$\Sigma^3 = \{ (a,a,\cancel{b}a), (a,a,b), --, (b,b,b) \}$$
$$= \{ aaa, aab, --, bbb \}$$

$$\Sigma^* = \bigcup_{k=0,1,2,\ldots} \Sigma^k$$

E.g. $\Sigma = \{a\}$

$$\Sigma^* = \{a\}^*$$

$$= \{\varepsilon, a, aa, aaa, \ldots\}$$

$$|\Sigma^k| = |\Sigma|^k$$

size

so $|\Sigma^0| = |\Sigma|^0 = 1$

$$\Sigma^2 \longleftrightarrow \{\text{maps } \{1,2\} \to \Sigma\}$$

$$\Sigma^0 \longleftrightarrow \{\text{maps } \emptyset \to \Sigma\}$$

$$\Sigma^0 = \{\varepsilon\} \text{ where}$$

ε is the "empty string"

$$\{a,b\}^*$$

$$= \{\varepsilon, a, b, aa, ab, ba, bb,$$
$$aaa, aab, \ldots, bbb,$$
$$aaaa, \ldots\}$$

$$\Sigma^5 \times \Sigma^7 \cong \Sigma^{12}$$

concatenation

$$\Sigma^3 \times \Sigma^2 = \Sigma^5$$

say $\Sigma = \{a, b\}$

$abc \qquad bb \qquad ababb$

$$\Sigma^0 \times \Sigma^2 = \Sigma^2$$

$\varepsilon \qquad aa \qquad \varepsilon aa$

$= aa$

Σ^* = set of strings over Σ
(words)

=

If S is a set, we use

$Power(S)$ to denote the

set of all subsets of S

$S = \{1, 2\}$

$Power(\{1, 2\})$

$= \{ \emptyset, \{1\}, \{2\}, \{12\} \}$

$$\text{Power}(\{1,2,3\})$$

$$= \{ \emptyset, \{1\}, \{2\}, \{3\}, \{1,2\}, \{1,3\}, \{2,3\}, \{1,2,3\} \}$$

$$|\text{Power}(S)| = 2^{|S|}$$

if S is finite

Cantor's Theorem: If S is a set, and function $f: S \longrightarrow \text{Power}(S)$, then the image of f is not all of $\text{Power}(S)$, specifically

$$T = \{ s \mid s \notin f(s) \}$$

is not in the image of f, i.e. there is no $t \in S$ s.t. $f(t) = T$.

e.g.

$$f$$

$$1 \longmapsto \{1,2\}$$

$$2 \longmapsto \{1,2,3\}$$

$$3 \longmapsto \{1,2\}$$

$$1 \in f(1) = \{1,2\}$$

$$2 \in f(2) = \{1,2,3\}$$

$$3 \notin f(3) = \{1,2\}$$

$$T = \{s \mid s \notin f(s)\} = \{3\}$$