Lecture 1

Administrivia

A Section:	Lecture	Review Session
Beginning – Mar 16:	Patrick	Chandra
Mar 18 – End:	Chandra	Patrick

Fri 9:00 – 10:30a

Chat Moderator: Vasilis

A vs B: different lecturers
Same Lab/HW/Exams
groups across sections ok!

Labs: Start <u>today</u> / tomorrow.

HW 0 / Quiz 0: available ~~soon~~ now!

Quiz: due Mon 10a
HW: due Wed 10a
Exams: synchronous, proctored via Zoom
MT1: Mar 1 6:30p
MT2: Apr 12 6:30p

conflicts next day early morn

DRES: Send LOA to Patrick & Chandra well before Midterm

Webpage:
- Lecture Schedule / Readings / Recordings
- Labs, HW
- Office Hours Schedule / Procedures
- (Too Many) policies

Other Sites:
- Piazza: Q&A + Announcements
- Discord: OH + Socialization
- Gradescope: HW / Exam grading
- PrairieLearn: Quizzes / Additional Practice

Course Structure / Philosophy

Guidance (lecture) → Guided Practice (labs) → Practice (quiz/hw) → Feedback (grading hw/exams) → Guidance (lecture)

A lot of material very quickly

refresh prereqs CS 173/225
→ HW0 / Quiz0

What are we talking about?

- Computation, formally (automata, TMs)
- How to compute "efficiently" (algorithms design)
- Limits of computation (undecidability & NP-Hardness)

Key Idea: Reduction $A \le B$

Informally: problem A, convert it to problem B
apply known solution for B to solve A.

Why?

- Computers existed for millenia
  many physical computers share "same" limitations
  capture this into abstract "idealized" mathematical machine

  physical world $\xrightarrow{\text{inspire}}$ model of computation
  model of computation $\xrightarrow{\text{implement}}$ physical world

- efficiency is important & difficult
  e.g. multiplying 2 n-digit #s.

$$\begin{array}{r} 3141 \\ \times\ 2718 \\ \hline 25128 \\ 3141 \end{array}$$

"alternatives"
e.g. lattice multiplication

25 128
3141
21987
6282

8537238

$O(n^2)$

↳ equivalent either asymptotically or algorithmic

Kolmogorov conjectured $\Omega(n^2)$

Karatsuba 1960 $O(n^{1.58})$

⋮

Schönhage-Strassen 1971 $O(n \log n \log\log n)$

Furer 2008 $O(n \log n \, 2^{O(\log^* n)})$

Harvey-van der Hoeven 2019 $O(n \log n)$

- Limits of computation also useful, & surprising.

b	ba	abb	abb	a
bbb	bbb	a	baa	ab

Post correspondence Problem

Can I arrange copies of these tiles so the top matches the bottom?

abb	ba	abb	a	abb	b
a	bbb	a	ab	baa	bbb

Cannot be solved w/ a computer???

How?

common to many "computers" throughout history to store information as "strings"

Formally: alphabet Σ finite set

Formally : alphabet Σ finite set

e.g. $\Sigma = \{0, 1\}$

$\Sigma = \{a, \dots, z\}$

$\Sigma = \{\langle A440\rangle, \langle C261.63\rangle, \dots\}$

string is a finite sequence of symbols from Σ

01001111 'string'

'$\langle C261.63\rangle\langle F\#369.99\rangle\langle G392.00\rangle$'

Simplistically talk about "Decision" problem
answer is Y/N.

- Post Correspondence Problem
- Given a list, is it sorted?

Language : a set of strings not necessarily finite.

$$\mathcal{L} = \{ x \mid \text{the answer in problem } P \text{ for } x \text{ is "Yes"} \}$$

$$\updownarrow$$

P : Given x, is $x \in \mathcal{L}$?

deciding Y/N problems $\equiv$ solving "language membership"

Defining strings recursively

a string is either

ε	(the empty string) ""
$a \cdot x$	$a \in \Sigma$, x string

string = s · tring = s · (t · ring) = ⋯ = $s \cdot (t \cdot (r \cdot (i \cdot (n \cdot (g \cdot \varepsilon)))))$

operations on strings

e.g. length |string| = 6.

operations on strings

e.g. length $|\text{string}| = 6$.

$|w| = \#$ symbols in w

$$|w| = \begin{cases} 0 & \text{if } w = \varepsilon \\ 1 + |x| & \text{if } w = ax \end{cases}$$

e.g. concatenation

$w \bullet z$

$$\underbrace{\overset{a}{\text{H}}\overbrace{\text{EAD}}^{x}}_{w} \bullet \underbrace{\text{ACHE}}_{z} = \underbrace{\overset{a}{\text{H}}\overbrace{\text{EADACHE}}^{x \bullet z}}_{w \bullet z}$$

$$w \bullet z = \begin{cases} z & w = \varepsilon \\ a(x \bullet z) & w = ax \end{cases}$$

Thm. $|w \bullet z| = |w| + |z|$

Pf. Let w, z be arbitrary strings

IH: Assume $|x \bullet z| = |x| + |z|$ for all x where $|x| < |w|$

There are two cases

- $w = \varepsilon$ (base case) $\rightarrow$

$$\begin{aligned} |w \bullet z| &= |\varepsilon \bullet z| && w = \varepsilon \\ &= |z| && \text{def of } \bullet \\ &= 0 + |z| && \text{add } 0 \\ &= |\varepsilon| + |z| && \text{def of } |\,| \\ &= |w| + |z| && w = \varepsilon \end{aligned}$$

- $w = ax$ (inductive case) $\rightarrow$

$$\begin{aligned} |w \bullet z| &= |ax \bullet z| && w = ax \\ &= |a(x \bullet z)| && \text{def of } \bullet \\ &= 1 + |x \bullet z| && \text{def of } |\,| \\ &= 1 + |x| + |z| && \text{IH} \\ &= |ax| + |z| && \text{def of } |\,| \\ &= |w| + |z| && w = ax \end{aligned}$$

$\square$

Ops on Languages

given langs L, M

$$L \cdot M = \{ x \cdot y \mid x \in L, y \in M \}.$$

$L \times W = \{ (x,y) \mid x \in L, y \in M \}$ is not a language

$\Sigma^0 = \{ \varepsilon \}$ $\quad \varepsilon \neq \{\varepsilon\} \neq \{\emptyset\} \neq \emptyset$

$\Sigma^1 = \{ a \mid a \in \Sigma \}$

for $n > 1$ $\Sigma^n = \Sigma^1 \Sigma^{n-1}$ is the set of length n strings

$\Sigma^* = \bigcup_{n \geq 0} \Sigma^n$ is the set of all possible strings

$= \bigcup_{n=0}^{\infty} \Sigma^n = \Sigma^0 \cup \Sigma^1 \cup \dots \cup \Sigma^\infty$ countably infinite set

$|\Sigma^*| = |\mathbb{N}|$

$L \subseteq \Sigma^*$ for all langs L.

every program is a string

$\#$ programs $= |\Sigma^*|$

$\#$ decision problems

$= \#$ languages $= |P(\Sigma^*)| \neq |\Sigma^*|$

exist (uncountably many) problems

that cannot be solved via a computer.