CS 121: Lecture 14
Uncomputability

Madhu Sudan

https://madhu.seas.Harvard.edu/courses/Fall2020

Book: https://introtcs.org

How to contact us

The whole staff (faster response): CS 121 Piazza
Only the course heads (slower): cs121.fall2020.course.heads@gmail.com
Announcements:

• Midterm 1 graded. Solutions to be posted today-ish.
• Homework 4 due in 9 days.
• Thanks for participating in Midterm Feedback Survey.
Where we are:

Part I: Circuits: 
Finite computation, quantitative study

Part II: Automata: 
Infinite restricted computation, quantitative study

Part III: Turing Machines: 
Infinite computation, qualitative study

Part IV: Efficient Computation: 
Infinite computation, quantitative study

Part V: Randomized computation: 
Extending studies to non-classical algorithms
Today:

- Finiteness and Infinities
- Cantor: \#Reals > \#Rationals (Uncountable vs. Countable sets)
- Uncomputable function by counting
- Explicit Uncomputable function: HALT
Background: Finiteness & Infinites

• Back prior to 1800s:
  • Understood finite vs. infinite
    • Set $S$ is finite if $\exists n \in \mathbb{N}$ s.t. $\exists$ 1-1 function $E: S \rightarrow [n]$
    • Infinite otherwise.
      • Example infinite sets: $\mathbb{N}, \mathbb{Z}, \mathbb{Z}^2, \mathbb{R}, \mathbb{R}^{10}$
      • Thinking then: All of same size? No point comparing?

• ... Cantor ’1800s:
  • $|S| \leq |T| \iff \exists$ 1 – 1 $E: S \rightarrow T$: Applies also to infinite sets.
  • Examples: $|\mathbb{N}| = |\mathbb{Z}| = |\mathbb{Q}| = |\mathbb{Z}^2| = |\{0,1\}^*|$
  • Thm: No 1-1 function $E: \mathbb{R} \rightarrow \mathbb{Z}$ exists. ($|\mathbb{Q}| < |\mathbb{R}|$)

Glossary of terms:
• $\mathbb{N} =$ Natural numbers
• $\mathbb{Z} =$ Integers
• $\mathbb{Q} =$ Rationals
• $\mathbb{R} =$ Reals
• $[0,1] = \{x \in \mathbb{R}|0 \leq x \leq 1\}$
• $E: A \rightarrow B$ 1-1 (aka injective): $E(a) = E(a') \Rightarrow a = a'$
• $F: B \rightarrow A$ onto (aka surjective): $\forall a \in A \exists b \in B$ s.t. $F(b) = a$
Cantor’s Proof

• Suppose \( 1 - 1 E : [0,1] \rightarrow \mathbb{N} \). Then \( \exists \) onto \( F : \mathbb{N} \rightarrow [0,1] \)
• Then ... can draw matrix with \( F(i)_j = j \)th bit in binary expansion of \( F(i) \).
• Consider \( \bar{F} = F(0)_0 F(1)_1 F(2)_2 F(3)_3 \ldots \) where does it lie? [Can’t be \( i \)th row.]
• Doesn’t! Hence \( F \) can’t exist!
Uncomputable functions by counting

• Q1: How many computable functions are there?
  • Claim: At most $|\{0,1\}^*|$
  • Why?

• Q2: How many functions $f: \{0,1\}^* \to \{0,1\}$
  • Claim: $|[0,1]|$

• Put together: $|R| < |ALL|$, where
  
  $R = \{F: \{0,1\}^* \to \{0,1\}|F$ is computable\};
  
  $ALL = \{F: \{0,1\}^* \to \{0,1\}\}$

• $\implies \exists F \in ALL \setminus R$
Exercise Break 1

Give direct proof a la Cantor that $|\{0,1\}^*| < \text{ALL} \ \overset{\text{def}}{=} \{f : \{0,1\}^* \to \{0,1\}\}$

$$
\begin{array}{ccccccc}
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \\
0 & 0 & 1 & 0 & 0 & 0 & 1 \ 0 \ 0 \\
\end{array}
$$

- Rows = computable function
- Row = description of Turing machines
\[ \begin{array}{c}
0 \quad 1 \quad 0 \quad 1 \quad 0 \quad 0 \\
\downarrow \downarrow \downarrow \downarrow \downarrow \downarrow
\end{array} \]

\[ f \rightarrow 0 \quad 1 \quad 0 \quad 1 \quad \cdots \]

→ Rows \equiv All \ Thriving \ Machines

\[ f : \{0, 1\}^* \rightarrow \{0, 1\} \]

\[ \downarrow \]

\[ f(\cdot) : f(0), f(1), f(00), f(01), \ldots \]

→ Columns = enumeration of inputs to function

→ Matrix = evaluation & row on column.
Rows of matrix are $S_0, 1, 8^* \leftarrow$

Columns of matrix are $S_0, 1, 8^*$

Entry of matrix at row $y$, column $x$:

- if $y = \text{encoding of Turing Machine } M$
  
  & then matrix at $(y, x) = M(x)$. [If $M$ halts & outputs a bit]

[In all other cases, write 0]
Explicit Uncomputable Functions?

- Motivation: Are “uncomputable” functions of interest to us?
  - Maybe they exist but can’t even be described.
  - (#describable functions = countable! By definition!)
  - If they can’t be described why would we be interested in computing them?
  - Turns out: Many natural problems uncomputable.
  - As we will see, the following (very describable!) problem is uncomputable.

  \[
  \text{HALT}(M, x) = \begin{cases} 
  1 & \text{if } M \text{ halts on input } x \\
  0 & \text{otherwise}
  \end{cases}
  \]

- Will see in next lecture: HALT is uncomputable
An Explicit Uncomputable Function

- Cantor inspired by the diagonalization proof

- Idea:
  - columns = \{0,1\}^* = inputs
  - rows = \{0,1\}^* \cong Turing machines
  - Mth row, xth column = (M, x)
  - If row not TM – fill with 0s.
  - If M does not halt on x enter 0.
  - Consider function that computes diagonal entries and flips them.

- Cantor(M) = \overline{M(M)}
Exercise Break 2

• Prove Cantor is uncomputable, where \( \text{Cantor}(M) = \overline{M(M)} \)

- say \( \exists \) Turing machine that computes Cantor
- say \( X \) is encoding of this Turing Machine

\[
\forall m \quad X(m) = \text{Cantor}(m)
\]

What is \( X(X) \)?

\[
X(X) = \text{Cantor}(X) = \overline{M(M)} \bigg|_{M=X} = \overline{X(X)}
\]
Proof:

• Assume for contradiction that Turing Machine \( A \) computes Cantor
• Then we have \( \forall M \quad A(M) = \overline{M(M)} \)
• So \( A(A) = \overline{M(M)}|_{M=A} = \overline{A(A)} \). ... Contradiction!!
Next Lecture: More Uncomputability

- Uncomputability of new problems
  - \textsc{Halt}, \textsc{Halt-On-Zero}

- Two proof techniques
  - Using presumed (non-existent) Turing Machine
  - \textsc{Reductions}!!!
    - Using a hard problem to show other problems are also hard.