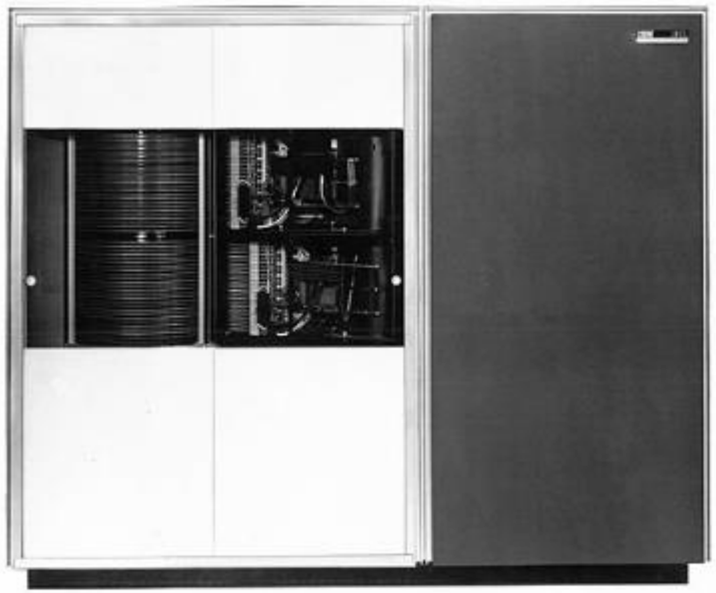
CS3102 Theory of Computation

www.cs.virginia.edu/~njb2b/cstheory/s2020

Warm up:

1. Why does math consider infinity?
2. Do infinite things exist

O, Ω, Θ



IBM 1301 disk storage unit (1961)
28MB capacity
$2,100 per month ($18,000 today)

Logistics

- Course registration survey was due Thursday
  - Didn't complete it? No problem! Just do it soon
- Exercise 0_2 due Tuesday
  - Didn't complete it? No problem (this time)! Just request an extension.
- Exercise 0_3 due Today
  - Pick one of python/java
  - Didn't complete it? No problem(this time)! Just request an extension.
- Exercise due/release dates will be more "batched" going forward, staggered this time to test out the submission system
- First Quiz
  - Released Tomorrow, due Tuesday
- Exercises 1_x released tomorrow, see assignments page for deadlines

Last Class

- What does it mean to "represent" things with strings? $f_n : \Sigma^n \to \mathbb{N}$
- How can we represent natural numbers with binary strings? $f_4(1101) = 1 \cdot 2^0 + 0 \cdot 2^1 + 1 \cdot 2^2 + 1 \cdot 2^3$
- How can we count things? bijections
- What does it mean for a set to be infinite? its card is not a natural #

Differences Between "real" computing and Math

- $(-80538738812075974)^3 + 80435758145817515^3 + 12602123297335631^3$
  - Answer?
- Nearly everything in math in **infinite**
- Everything in computing is finite
- If the numbers are large enough, computers will always start to do the math wrong

Why bother with infinity?

- Even though computers have finite resources, most of the time they are plentiful

How many binary strings of any length?

- If we don’t limit the length, how many strings are there?
  - ∞?
- What naturals can/can’t we represent
- What does it mean to “represent”?
  - A surjective mapping from a set of strings onto a set
  - Ideally a bijection, but not necessary
- Are there things we can’t represent?

Representing $\mathbb{N}$ with binary strings

$f_n(b) = x$

17,024

- Let $f_*(b) = f_n(b)$ when $|b| = n$
- For a binary string b, if $f_*(b) = x$, then $f_*(0b) = x$
  - Leading zeros don't change the value
  - Our procedure above gives an onto mapping from binary strings to the natural numbers
  - We can represent all natural numbers with binary strings

Countablility and Uncountability

- A set S is countable if $|S| \leq |\mathbb{N}|$
  - If $|S| = |\mathbb{N}|$, then S is "countably infinite"
- A set S is countable if there is an onto (surjective) function from $\mathbb{N}$ to S

$\{0,1\}^*$ is countable

$|\{0,1\}^*| \geq |\mathbb{N}|$
$\leq$

- We showed $|\{0,1\}^*| \geq |\mathbb{N}|$
- Countable if $|\{0,1\}^*| \leq |\mathbb{N}|$
- Need to "represent" strings with naturals
- Idea: build a "list" of all strings, represent each string by its index in that list

0, 1, 2, 3, ...

Listing all strings

- $\{0,1\}^0 = \{\underset{0}{""}\}$
- $\{0,1\}^1 = \{\underset{1}{0},\underset{2}{1}\}$
- $\{0,1\}^2 = \{\underset{3}{00},\underset{4}{01},\underset{5}{10},\underset{6}{11}\}$
- $\{0,1\}^3 = \{\underset{7}{000},\underset{8}{001},\underset{9}{010},\underset{10}{011},\underset{11}{100},\underset{12}{101},\underset{13}{110},\underset{14}{111}\}$

How Many Python/Java programs?

$|\Sigma| = 2^{64}$

- How do we represent Java/Python programs?
- How many things can we represent using that method?

How many functions $\Sigma^* \to \Sigma^*$?

- Short answer: Too many!
  - Uncountable
  - $|\{f \mid f : \Sigma^* \to \Sigma^*\}| > |\mathbb{N}|$ $= |\{0,1\}^*|$
- Conclusion: Some functions cannot be computed by any java/python program
- How to prove this? proof by contradiction
  diagonalization

Uncountably many functions

- If we show a subset of $\{f \mid f: \Sigma^* \rightarrow \Sigma^*\}$ is uncountable, then $\{f \mid f: \Sigma^* \rightarrow \Sigma^*\}$ is uncountable too
- Consider just the "yes/no" functions (decision problems): $\{f \mid f: \{0,1\}^* \rightarrow \{0,1\}\}$
- The right-hand column is an infinite binary string that represents that function

b	$f(b)$
""	1
0	0
1	0
00	1
01	1
10	1
11	1
000	0
001	0

⋮

$|\{0,1\}^\infty| > |\mathbb{N}|$

- Idea:
  - show there is no way to “list” all ~~finite~~ infinite binary strings
  - Any list of binary strings we could ever try will be leaving out elements of $\{0,1\}^\infty$

$|\{0,1\}^\infty| > |\mathbb{N}|$

Cantor

Attempt at mapping $\mathbb{N}$ to $\{0,1\}^\infty$

	b_0	b_1	b_2	b_3	b_4	b_5	b_6
0	1	1	1	1	1	1	1
1	0	0	0	0	0	0	0
2	1	0	1	0	1	0	1
3	1	1	0	1	1	0	1
4	1	0	1	1	0	1	0
5	1	0	0	1	1	1	0
6	0	0	0	1	1	1	1
…							
	0	1	0	0	1	0	0

A string that our attempt missed

Derive by selecting each b_i as the opposite of the b_i from row i

$|\{0,1\}^\infty| > |\mathbb{N}|$ proof summary

- Assume towards reaching a contradiction that $\{0,1\}^\infty$ is countable
- This means we can find a bijection $f: \mathbb{N} \to \{0,1\}^\infty$
- Using f, we can find $s \in \{0,1\}^\infty$ which is not in the range of f:
  - let bit i of s be the opposite of bit i of $f(i)$
  - This is missing from the range because it must be different from every output (at the position indexed by the input)

Other countable/uncountable sets

- Countable sets:
  - Integers
  - Rational numbers
  - Any finite set

$1/3 \quad \frac{2}{7} \quad \frac{1}{2}$

- Uncountable Sets:
  - Real numbers
  - The power set of any infinite set

Cantor's thm

$|S| < |2^S|$

Cantor's Theorem

- For any set S, $|S| < |2^S|$
- Even if S is infinite!
- Idea:
  - $|S| \leq |2^S|$ (why?)
  - There cannot be a bijection between S and 2^S

$|S| \neq |2^S|$

- Consider, towards reaching a contradiction, that there is a bijection $f: S \to 2^S$
- Consider the set $P = \{x : x \in S \wedge x \notin f(x)\}$
  – What are the “types” of:
    - S
    - 2^S
    - x
    - $f(x)$
    - P
- Let $f(p) = P$, is $p \in P$?

Conclusion

- There are countably many strings
  - And therefore binary strings, programs, etc.
- We can’t write down (or compute) all things from an uncountable set
- There are uncountably many functions
- Some functions can’t be implemented