Theory of Computation
Countability and Diagonalization

Arjun Chandrasekhar

Countability and Diagonalization

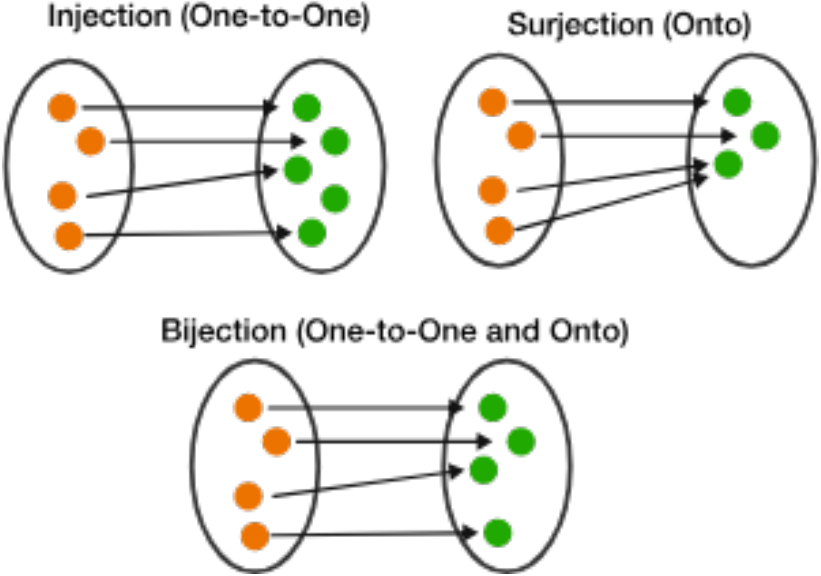
- We will show that some infinite sets are "bigger" than others
- We will show that there are strictly more languages than there are Turing machines
- This will imply that there is not a Turing machine for every language

Bijection

Let S_1 and S_2 be sets. A **bijection** between S_1 and S_2 is a one-to-one correspondence between their elements

- **Surjective:** Every element from S_2 is mapped to at least once
- **Injective:** Every element of S_1 maps to exactly one element of S_2

Bijection


Injection (One-to-One)
Surjection (Onto)
Bijection (One-to-One and Onto)

Bijection Example

- Let $\mathbb{N} = \{0, 1, 2, \dots\}$ (natural numbers)
- Let $S = \{n^2 | n \in \mathbb{N}\} = \{0, 1, 4, 9, 16, \dots\}$ (square integers)
- Note that $S \subsetneq \mathbb{N}$. And yet...
- ...there exists a bijection between $\mathbb{N}$ and S
  - $0 \mapsto 0$
  - $1 \mapsto 1$
  - $2 \mapsto 4$
  - $3 \mapsto 9$
  - $\dots$

Countable Sets

- **Axiom:** The natural numbers $\mathbb{N} = \{0, 1, 2, \dots\}$ are countable
- A set S is **countably infinite** if there exists a bijection $\mathbb{N} \mapsto S$
- Can also think of it as follows: can we write a program to print out the elements of S one by one, such that every element eventually gets printed if we let the program run long enough?

Countability of Square Numbers

- **Proposition**: The set of squares $S = \{0, 1, 4, 9, 16, \dots\}$ is countably infinite
- **Proof**: There exists a bijection $\mathbb{N} \mapsto S$
  - $n \mapsto n^2$
- **Alternate interpretation**: We can write a program that prints out n^2 for $n = 0, 1, 2, \dots$
  - Given enough time, every square number will (eventually) be printed

Countability of $\mathbb{Z}$

Let's prove that the set of integers $\mathbb{Z} = \{\dots, -3, -2, -1, 0, 1, 2, 3, \dots\}$ is countably infinite

- $0 \mapsto 0$
- $1 \mapsto 1, 2 \mapsto -1$
- $3 \mapsto 2, 4 \mapsto -2$
- $5 \mapsto 3, 6 \mapsto -3$
- $\dots$

Countability of $\mathbb{N}^2$

Let's prove that the following set is countably infinite

$$\mathbb{N}^2 = \{(x, y) | x, y \in \mathbb{N}\}$$

i.e. every combination of 2 natural numbers

- Go through all combinations that add up to 0
  - $0 \mapsto (0, 0)$
- Go through all combinations that add up to 1
  - $1 \mapsto (1, 0)$
  - $2 \mapsto (0, 1)$
- Go through all combinations that add up to 2
  - $3 \mapsto (2, 0)$
  - $4 \mapsto (1, 1)$
  - $5 \mapsto (0, 2)$

Countability of $\mathbb{N}^2$: dovetailing

New “set” of combinations that add up to a common sum total

	0	**1**	**2**	**3**	**...**
0	(0, 0)	(0, 1)	(0, 2)	(0, 3)	...
1	(1, 0)	(1, 2)	(1, 2)	(1, 3)	...
2	(2, 0)	(2, 1)	(2, 2)	(2, 3)	...
3	(3, 0)	(3, 1)	(3, 2)	(3, 3)	...
...	...	...	...	...	...

Countability of $\mathbb{Q}$

Theorem: The rational numbers are countable

$$\mathbb{Q} = \{\, {}^{a}\!/_{b} \mid a, b \in \mathbb{N}, b \neq 0\}$$

Hint: Go through all possible numbers that the numerator and denominator can add up to

Countability of $\mathbb{Q}$

Theorem: The rational numbers are countable

$$\mathbb{Q} = \{\, {}^{a}/_{b} \mid a, b \in \mathbb{N}, b \neq 0\}$$

- $0 \mapsto 0$
- numerator and denominator add up to 2
  - $1 \mapsto {}^{1}/_{1}$
- numerator and denominator add up to 3
  - $2 \mapsto {}^{2}/_{1}$
  - $3 \mapsto {}^{1}/_{2}$
- numerator and denominator add up to 4
  - $4 \mapsto {}^{3}/_{1}$
  - $5 \mapsto {}^{1}/_{3}$
- ...

Countability of $\mathbb{Q}$: dovetailing

New “set” of combinations that add up to a common sum total

	1	**2**	**3**	**4**	**...**
1	1/1	1/2	1/3	1/4	...
2	2/1	1	2/3	2/4	...
3	3/1	3/2	1	3/4	...
4	4/1	2	4/3	1	...
...	...	...	...	...	...

Skip over redundant combinations

Countability of Finite Binary Strings

Proposition: The set of all possible finite binary strings is countable

- List all possible strings of length 1
  - $0 \mapsto 0$
  - $1 \mapsto 1$
- List all possible strings of length 2
  - $3 \mapsto 00$
  - $4 \mapsto 01$
  - $5 \mapsto 10$
  - $6 \mapsto 11$
- List all possible strings of length 3
  - $\ldots$

Countability of Java Programs

Proposition: The set of all possible java programs is countable

- List all possible programs with 0 characters
- List all possible programs with 1 character
- List all possible programs with 2 characters
- ...

Countability of Turing Machines

Proposition: The set of all possible Turing machines on the alphabet $\{0, 1\}$ is countable

- List all possible TMs with 1 state
- List all possible TMs with 2 states
- List all possible TMs with 3 states
- ...

Uncountability of $\mathbb{R}$

Theorem: The real numbers $\mathbb{R}$ are uncountable

- **Proof Idea:** Assume for sake of contradiction that $\mathbb{R}$ is countable, and construct a paradoxical number r^*
- **Technique** diagonalization

Uncountability of $\mathbb{R}$

- AFSOC $\mathbb{R}$ is countable. Then there exists a bijection $0 \mapsto r_0, 1 \mapsto r_1, \ldots$
- Create a real number r^*
  - The i-th digit of r^* is different from the i-th digit of r_i (diagonalization)
  - r^* disagrees with every single r_i in the bijection

Uncountal

- AFSC … xists a bijecti
- Creat
  - T … e i-th digit o
  - r … bijection

Uncountability of $\mathbb{R}$

- AFSOC $\mathbb{R}$ is countable. Then there exists a bijection $0 \mapsto r_0, 1 \mapsto r_1, \ldots$
- Create a real number r^*
  - The i-th digit of r^* is different from the i-th digit of r_i (diagonalization)
  - r^* disagrees with every single r_i in the bijection
  - **Case 1:** If r^* was listed at index i, then it disagrees with itself at the i-th digit
  - **Case 2:** If r^* isn't part of the list, then our bijection is not valid

Diagonalization of $\mathbb{R}$

	Digit 0	Digit 1	Digit 2	...	Digit *
r_0	**0.**	1	2	...	8
r_1	1.	**6**	2	...	9
r_2	6.	4	**7**	...	3
...	...	...	...	...	...
r^*	**1.**	**7**	**8**	...	**???**

The assumption that $\mathbb{R}$ is contable allows us to list out all of the real numbers (and then construct a paradoxical number)

Construct r^* by modifying the **diagonals digits** until we reach a **contradiction**

Uncountability of infinite binary strings

Proposition: The set of infinite binary strings is uncountable

- Hint: proceed by contradiction
- construct a binary string that causes problems

Uncountability of Binary Strings

- AFSOC the inifinite binary strings are countable. Then there is a bijection with $\mathbb{N}$
- Create an infinite binary string s^* that disagrees with every string in the bijection
  - The i-th bit of s^* is different from the i-th bit of s_i (diagonalization)
- **Case 1:** If s^* was listed at index i, then it disagrees with itself at the i-th digit
- **Case 2:** If s^* isn't part of the list, then our bijection is not valid

Diagonalization of Infinite Binary Strings

	Bit 0	**Bit 1**	**Bit 2**	**...**	**Bit ***
s_0	**0**	1	0	...	0
s_1	0	**0**	0	...	0
s_2	1	1	**1**	...	1
...	...	...	...	...	...
s^*	1	1	0	...	**???**

The assumption that the infinite binary strings are countable allows us to list out all of the infinite binary strings (and then construct a paradoxical binary string)

Construct s^* by modifying the **diagonals bits** until we reach a **contradiction**

Uncountability of Formal Languages

- **Proposition:** The set of formal languages on any finite alphabet is uncountable
- **Proof:** We can draw a bijection between infinite binary strings and formal languages
- Let $\Sigma^* = \{w_1, w_2, \ldots\}$ be the set of all possible stirngs
- Represent a language $L_j \subseteq \Sigma^*$ using it's characteristic binary string S_j
  - The i-th digit of S_j s 1 if $w_i \in L_j$ and 0 otherwise
- The set of formal languages has a bijection with an uncountable set; thus it must be uncountable

Uncountability of Formal Languages

	w_0	w_1	w_2	...
L_0	1	0	0	...
L_1	0	0	1	...
L_2	1	1	1	...
...	...	...	...	...

Box (i, j):
“Is string w_i part of language L_j”?

We can represent every language using an infinite binary string

Existence of Unrecognizable Languages

Corollary: There exist unrecognizable languages

- The set of Turing machines is countable
- The set of languages is uncountable
- So there cannot possibly be a Turing machine to recognize every language