Quick Review:

Cardinality: a generalization of "size".

- Allows us to reason about / compare infinite sets
- Use injections / surjections instead of # of elements.

Strategies to show $|A| = |B|$:

- Construct a bijection
- Construct two injections
- Construct two surjections

S is <u>countable</u> if it is either finite or $|S| = |\mathbb{N}|$.

- Enough to present an injection $S \to \mathbb{N}$

S is <u>uncountable</u> if it is not countable.

Can be shown using <u>diagonalization</u>:

- Prove it using contradiction
  (1) You assume its enumerable
  (2) Construct something that you didn't count.

Some Sets Whose Cardinalities You Should Know:

- Countable: ϕ, $\mathbb{N}$, $\mathbb{Z}$, $\mathbb{Q}$
- Uncountable: $\mathbb{R}$, $\mathbb{C}$
- Power sets
  - If S is finite, then $\mathcal{P}(S)$ is finite
  - If S is infinite + countable, then $\mathcal{P}(S)$ is uncountable.
- Bit String
  - The set ~~fe~~ of finite strings from a countable alphabet is <u>countable</u>.
  - Replace 'finite' with 'infinite' or 'countable alphabet' with 'uncountable alphabet', then it is <u>uncountable</u>.

Computability - Can you write a program that executes any given fn?

- No! # of functions is uncountable, while # of programs is countable
- Ex: TestHalt(P, x) is uncomputable.
  - ↳ Common strat to show that P is uncomputable is to use P to solve TestHalt.

Idea: Try to build a function that induces a contradiction.

Turing(P): oppusite of what P(P) does.

```
Turing(P):
  if TestHalt(P, P):
    run forever
  else:
    halt.
```

Turing(Turing) induces a contradiction

$f: \mathbb{N} \to \mathbb{N}$, $F = \{ f \mid f: \mathbb{N} \to \mathbb{N} \}$

Assume that this is countable.

$$
\begin{array}{c|cccc}
 & 0 & 1 & 2 & 3 \; \cdots \\
f_1 & f_1(0) & f_1(1) & f_1(2) & \cdots \\
f_2 & f_2(0) & f_2(1) & f_2(2) & \cdots \\
f_3 & f_3(0) & f_3(1) & f_3(2) & \cdots \\
\vdots & & & &
\end{array}
$$

$g(0) = f_1(0) + 1$ 0

$g(1) = f_2(1) + 1$ 6

$g(2) = f_3(2) + 1$ 0

$g(3) = f_4(3) + 1$

$\vdots$

g is not in the enumeration, so F is not countable.

Unions and Intersections

2A #1

Decide if the following expressions are either "Always Countable," "Sometimes countable," "Always Uncountable," or "Sometimes Uncountable." Provide proof / examples.

(a) $A \cap B$, where A is countable and B is uncountable.

Always Countable, as

$$A \cap B \subseteq A,$$

and A is countable.

(b) $A \cup B$, where A is countable and B is uncountable.

Always Uncountable, as

$$B \subseteq A \cup B$$

and B is uncountable.

(c) $\bigcap_{i \in A} S_i$, where A is a countable set of indices and S_i is uncountable for all i.

Sometimes. If $S_i = \mathbb{R}$ $\forall i$, then it is uncountable. If $S_i = [i, i+1]$ $\forall i$, then it is empty and therefore countable.

Counting Cartesian Products

2A # 2

(a) The Cartesian product of two sets A, B is

$$A \times B := \{(a,b) \mid a \in A, b \in B\}.$$

Prove that if A and B are countable, then $A \times B$ is countable.

$$\begin{array}{c|cccc} & & B & & \\ \hline A & (a_0, b_0) & (a_0, b_1) & (a_0, b_2) & \cdots \\ & (a_1, b_0) & (a_1, b_1) & (a_1, b_2) & \\ & (a_2, b_0) & (a_2, b_1) & (a_2, b_2) & \\ & & \vdots & & \ddots \end{array}$$

(b) For all positive integers $n \geq 2$, prove that the set

$$A_1 \times A_2 \times \cdots \times A_n$$

is countable when A_i is countable for all $1 \leq i \leq n$.

$$A_1 \times A_2 \times \cdots \times A_n = \{(a_1, a_2, \ldots, a_n) \mid a_i \in A_i \ \forall i\}$$

Idea: Induction on n.

<u>Base:</u> $A_1 \times A_2$ is countable by part (a)

<u>IH:</u> Assume $A_1 \times \cdots \times A_K$ is countable

<u>IS:</u>

$$\underbrace{[A_1 \times A_2 \times \cdots \times A_K]}_{= B} \times A_{K+1}$$

$$\underbrace{B \times A_{K+1}}_{\text{countable by part (a)}}$$

(c) Consider a countable collection of countable sets $B_1, B_2, \ldots$. Under what conditions is $B_1 \times B_2 \times \cdots$ a countable set? Provide proof.

Condi At most finitely many B_i ~~can~~ can have more than one element.

$A \times \{1\}$. $A \to \{(a, 1) \mid a \in A\}$

$B_1 = \{1, 2\}$

$B_2 : \{N, Z\}$

$B_3 : \{-1, 1\}$

Scanned with CamScanner

$$\begin{array}{lllll} & B_1 & B_2 & B_3 & \\ 1. & (b_1^1, & b_2^1, & b_3^1, & \dots \\ 2. & (b_1^2, & b_2^2, & b_3^2, & \dots \\ 3. & (b_1^3, & b_2^3, & b_3^3, & \dots \\ 4. & (b_1^4, & b_2^4, & b_3^4, & \dots \\ \vdots & & & & \end{array}$$

Assumption: B_i has more than one element $\forall i$.

$$\begin{array}{cccc} b_1^1 & b_2^2 & b_3^3 & \\ \neq & \neq & \neq & \\ \end{array}$$

$$C = (\underline{c_1}, \underline{c_2}, \underline{c_3}, \dots)$$

$$B_1 \times B_2 \times B_3 \times \cdots$$

Scanned with CamScanner

Hello World

2A # 3

Determine if the following tasks are computable. Provide either a program or a proof of uncomputability.

(a) A program that takes in a program P and an input x, and determines whether P(x) prints "Hello World".

Uncomputable. Consider the following program

```
def Q(P):
    run P while suppressing print
    statements
    print "Hello World".
```

We can implement TestHalt as below:

```
def TestHalt(P, x)
  if HelloWorld(Q,(P, x)).
     return True
  return False.
```

(b) A program that takes in a program P and an integer K and determines whether P prints "Hello World" before the Kth line of P is executed.

Uncomputable. We can implement part (a) using this.

```
def HelloWorld (P, x)
    for k ∈ P:
      if HelloWorldK (P, x, k):
          return True.
    return False.
```

(c) A program that takes in a program P and an integer K, and determines whether P prints "Hello World" when the first K lines are executed.

Computable. Just execute the first K lines.