Beyond Countability

$A \leq \mathcal{P}A$

There is always an *one-to-one* map from A to its *power set* $\mathcal{P}A$.

- $A = \emptyset$: It is *vacuously* true as A does not have any element and $\mathcal{P}\emptyset = \{\emptyset\}$.
- $A \neq \emptyset$: $f : A \longrightarrow \mathcal{P}A$, so that $f(a) = \{a\}$ for all $a \in A$, is *one-to-one*.

Cantor's Theorem

There cannot be any *onto map* (*surjection*) from any set A to its *power set* $\mathcal{P}A$.

Naturally, there cannot be any *bijection*. No set A is *equinumerous* to its *power set* $\mathcal{P}A$. Power set is always 'larger'.

Proof of Cantor's Theorem

The proof is by *reductio ad absurdum* using *diagonalization*.

Let $f : A \longrightarrow \mathcal{P}A$ be an *onto map* i.e. for each $B \in \mathcal{P}A$ ($B \subseteq A$), there is an $a \in A$, so that $f(a) = B$.

Consider the following subset of A,

$$D = \{a \in A : a \notin f(a) \subseteq A\}.$$

Nothing Fishy!

$$D = \{a \in A : a \notin f(a) \subseteq A\}.$$

This set is perfectly normal. It is not even empty if f is *onto*. Consider $\emptyset \in \mathcal{P}A$, there must be some $a_0 \in A$ so that $f(a_0) = \emptyset$. But then $a_0 \notin f(a_0) = \emptyset$ and therefore $a_0 \in D$.

Proof (*cont.*)

As f is *onto* there is some $a_1 \in A$ such that $f(a_1) = D$ because

$$D = \{a \in A : a \notin f(a) \subseteq A\} \in \mathcal{P}A.$$

By the law of *excluded middle* either $a_1 \in D$ or $a_1 \notin D$.

- $a_1 \in D = f(a_1)$ implies that $a_1 \notin D$.
- $a_1 \notin D = f(a_1)$ implies that $a_1 \in D$.
- *Contradiction* : $a_1 \in D$ *iff* $a \notin D$.

Hence f cannot be an *onto map*.

What and Where is Diagonalization?

The *diagonalization* can be understood in a better way if we take $A = \mathbb{N}$, the set of natural numbers.

- There is an *one-to-one* map f from $\mathbb{N}$ to $\mathcal{P}\mathbb{N}$ so that $f(n) = \{n\}$.
- We assume that there is also an *one-to-one* map g from $\mathcal{P}\mathbb{N}$ to $\mathbb{N}$. [The *left inverse* of g is a map from $\mathbb{N}$ *onto* $\mathcal{P}\mathbb{N}$. Therefore it is equivalent to assume the existence of an *onto map*.]

Diagonalization (*cont.*)

- By the *Schörder-Bernstein* theorem, there is a *bijection* h from $\mathbb{N}$ to $\mathcal{P}\mathbb{N}$.
- We can index the elements of $\mathcal{P}\mathbb{N}$ (subsets of $\mathbb{N}$) by the elements of $\mathbb{N}$ i.e.

$$\mathcal{P}\mathbb{N} = \{A_0, A_1, A_2, \cdots, A_{1000000}, \cdots\}$$

$\mathcal{P}\mathbb{N}$ as a Table

We may view $\mathcal{P}\mathbb{N}$ as an infinite table. For each subset of $\mathbb{N}$ there is a *row* and for each element of $\mathbb{N}$ there is a *column*. We put a '$*$' in the *ith row* and *jth column* if $j \in A_i$.

In our example the $0th$ *row* corresponds to the *null set*.

$\mathcal{P}\mathbb{N}$ as a Table

S/E	0	1	2	3	4	5	6	7	8	9	$\cdots$
A_0											$\cdots$
A_1	*	*		*					*		$\cdots$
A_2		*			*			*		*	$\cdots$
A_3			*		*		*			*	$\cdots$
A_4	*		*					*		*	$\cdots$
A_5		*				*	*				$\cdots$
A_6			*		*	*					$\cdots$
$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$

Diagonalization (*cont.*)

We construct $D \subseteq \mathbb{N}$ from the table in the following way.

$$i \in D \ \textit{ iff } \ i \notin A_i.$$

The set $D = \{0, 2, 3, 4, 6, \cdots\}$.

- The set D is constructed from the *diagonal* of the table.
- D cannot be same as any A_i, $i \in \mathbb{N}$.
- $D \in \mathcal{P}\mathbb{N}$ which does not have any *coimage* in $\mathbb{N}$ under the bijection h - a contradiction.

Conclusion

- $\mathbb{N} \leq \mathcal{P}\mathbb{N}$ but $\mathcal{P}\mathbb{N} \leq \mathbb{N}$ i.e. $\mathcal{P}\mathbb{N}$ is an infinite set but is not denumerable.
- $\mathcal{P}\mathbb{N}$ is called an *uncountable set*.
- $\mathbb{N} < \mathcal{P}\mathbb{N} < \mathcal{P}\mathcal{P}\mathbb{N} < \cdots < \mathcal{P}^n\mathbb{N} < \mathcal{P}^{n+1}\mathbb{N} < \cdots$.

$\mathbb{N}$ and $\mathbb{N}^{\mathbb{N}}$

The collection of all functions from the set of natural numbers ($\mathbb{N}$) to itself ($\mathbb{N}^{\mathbb{N}}$) cannot be *equinumerous* to $\mathbb{N}$.

Proof

The proof is again by *reductio ad absurdum* using *diagonalization*.

Let $F : \mathbb{N} \longrightarrow \mathbb{N}^{\mathbb{N}}$ be be a *bijection* so that $F(n) = f_n : \mathbb{N} \longrightarrow \mathbb{N}$. We define a new function $f : \mathbb{N} \longrightarrow \mathbb{N}$ in the following way.

$$f(n) = \begin{cases} 5 & \text{if } f_n(n) \neq 5, \\ 6 & \text{if } f_n(n) = 5, \end{cases} \quad \text{for all } n \in \mathbb{N}.$$

But then $f \neq F(n) = f_n$, for any $n \in \mathbb{N}$, because $f(n) \neq f_n(n)$ for each $n \in \mathbb{N}$. Hence F cannot be a *bijection* and $\mathbb{N}^{\mathbb{N}}$ is an *uncountabe* set.

Finally!

- The collection of all C* programs is a denumerable set $(\mathcal{F}_{C^*})$.
- The collection of all functions from the set of natural numbers to itself $(\mathbb{N}^{\mathbb{N}})$ is an uncountable set.
- There cannot be a bijection from $\mathcal{F}_{C^*}$ to $nat^{\mathbb{N}}$.
- Each function from $\mathbb{N}$ to itself cannot be computed by a C* program.

Uncountable *versus* **Denumerable**

Description or Specification

- The graph of a function $f : \mathbb{N} \longrightarrow \mathbb{N}$ or the language $L \subseteq \Sigma^*$, are objects of infinite size.
- These objects cannot be stored in a 'finite' (but potentially infinite) computer.
- It is necessary to find a finite description or specification to use them.

Description or Specification (*con.*)

- Any *description* or *specification* will use:
  - a finite[a] set of *meta-alphabet*, Γ[b], and
  - a *specification* will be a string of finite length over Γ.
- Both $\mathbb{N}^{\mathbb{N}}$ and $\mathbf{2}^{\Sigma^*}$ are *uncountable sets*.

[a] This set may be denumerable but finitely specifiable.

[b] The alphabet of the specification or the *meta language*. It is not Σ, the alphabet of the *object language*, L.

Description or Specification (*cont.*)

- There can be at most *denumerably* many *descriptions* or *specification.*
- Therefore every function or every language cannot be specified or described. We may call them *transcendental* function or language!
- There will always be languages and functions which cannot be *decsribed* and this is irrespective of the exact method of description.