Basic Structures: Sets, Functions, Sequences, Sums, and Matrices

Chapter 2

With Question/Answer Animations

Chapter Summary

- Sets
  - The Language of Sets
  - Set Operations
  - Set Identities
- Functions
  - Types of Functions
  - Operations on Functions
  - Computability
- Sequences and Summations
  - Types of Sequences
  - Summation Formulae
- Set Cardinality
  - Countable Sets
- ~~Matrices~~
  - ~~Matrix Arithmetic~~

Cardinality of Sets

Section 2.5

Section Summary

- Cardinality
- Countable Sets
- Computability

Cardinality relationships

If ∃ 1-1 function (injection) from A to B, the cardinality of A is less than or the same as the cardinality of B

$$|A| \leq |B|$$

If ∃ 1-1 correspondence (bijection) from A to B, *t*he cardinality of a set A is equal that of B

$$|A| = |\mathrm{B}|$$

If $|A| \leq |B|$ and A & B have different cardinalities, the cardinality of A is less than the cardinality of B

$$|A| < |B|$$

Countable

A set that is either finite or has the same cardinality as $\mathbf{Z}^+$ is *countable*.

A set that is not countable is *uncountable*.

- The set of real numbers **R** is an uncountable set.

When a set is *countably infinite* its cardinality is $\aleph_0$.

$\aleph$ is aleph (the 1st letter of the Hebrew alphabet).

The zero subscript is pronounced “naught” or “null”.

Showing that a Set is Countable

A 1-1 correspondence f from Z^+ to a set S can be expressed in terms of a sequence $a_1, a_2, \ldots, a_n, \ldots$ where

$$a_1 = f(1),\ a_2 = f(2), \ldots, a_n = f(n), \ldots$$

So can list the elements of S in a sequence indexed by Z^+.

Example: Z is countable:

$$0, 1, -1, 2, -2, \ldots$$

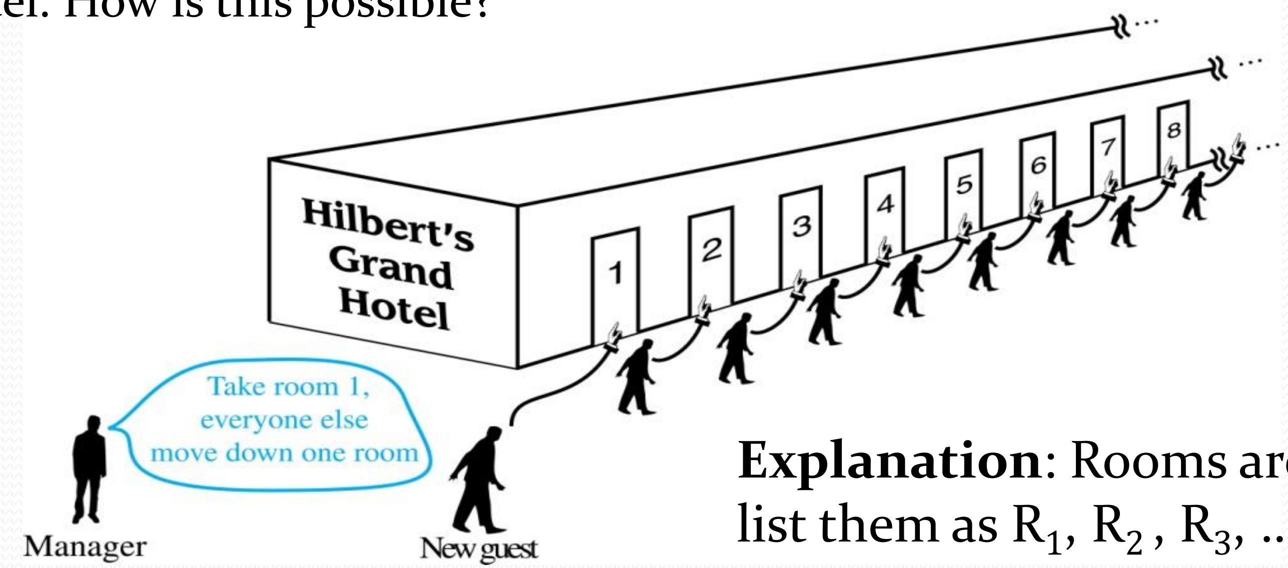
It is a good exercise but not necessary to create an explicit formula for the sequence:

$$a_n = (-1)^n \lfloor n/2 \rfloor$$

Hilbert's Grand Hotel

David Hilbert

The Grand Hotel (example due to David Hilbert) has countably infinite number of rooms, each occupied by a guest. We can always accommodate a new guest at this hotel. How is this possible?



Explanation: Rooms are countable so we list them as R_1, R_2 , R_3, … . When a new guest arrives, we move guest in R_1 to R_2, guest in R_2 to R_3 and in general guest in R_n to R_{n+1} , $\forall n \in Z^+$. This frees up R_1, which we assign to the new guest, and all the current guests still have rooms.

The hotel can also accommodate a countable number of new guests, all the guests on a countable number of buses where each bus contains a countable number of guests (see exercises).

Countable example: even>0

Show that the set of **positive** even integers E^+ is a countable set.

Solution: Let $a_n = 2n$. Or using function notation, $f(n) = 2n$

1	**2**	**3**	**4**	**5**	**6**	**.....**
↕	↕	↕	↕	↕	↕	
2	**4**	**6**	**8**	**10**	**12**	**......**

Then f is a bijection from **N** to E since f is both 1-1 and onto.

To show 1-1, if $f(n) = f(m)$, then $2n = 2m$, and so $n = m$.

To show onto, suppose that t is an even positive integer.

Then $t = 2k$ for some positive integer k and $f(k) = t$.

Countable example: even Z

Show that the set of even integers E is a countable set.

Solution: Can list in a sequence:

$$0, 2, -2, 4, -4, \ldots\ldots\ldots\ldots$$

Or can define a bijection from $\mathbf{Z}^+$ to even $\mathbf{Z}$:

$$a_n = (-1)^n \, 2\lfloor n/2 \rfloor$$

Q^+ is Countable

Recall: A *rational number* can be expressed as the ratio of two integers p and q such that $q \neq 0$.

- ¾ is a rational number
- √2 is not a rational number.

Example: Show Q^+ is Countable.

Solution: Q^+ can be arranged in a sequence:

$$r_1, r_2, r_3, \ldots$$

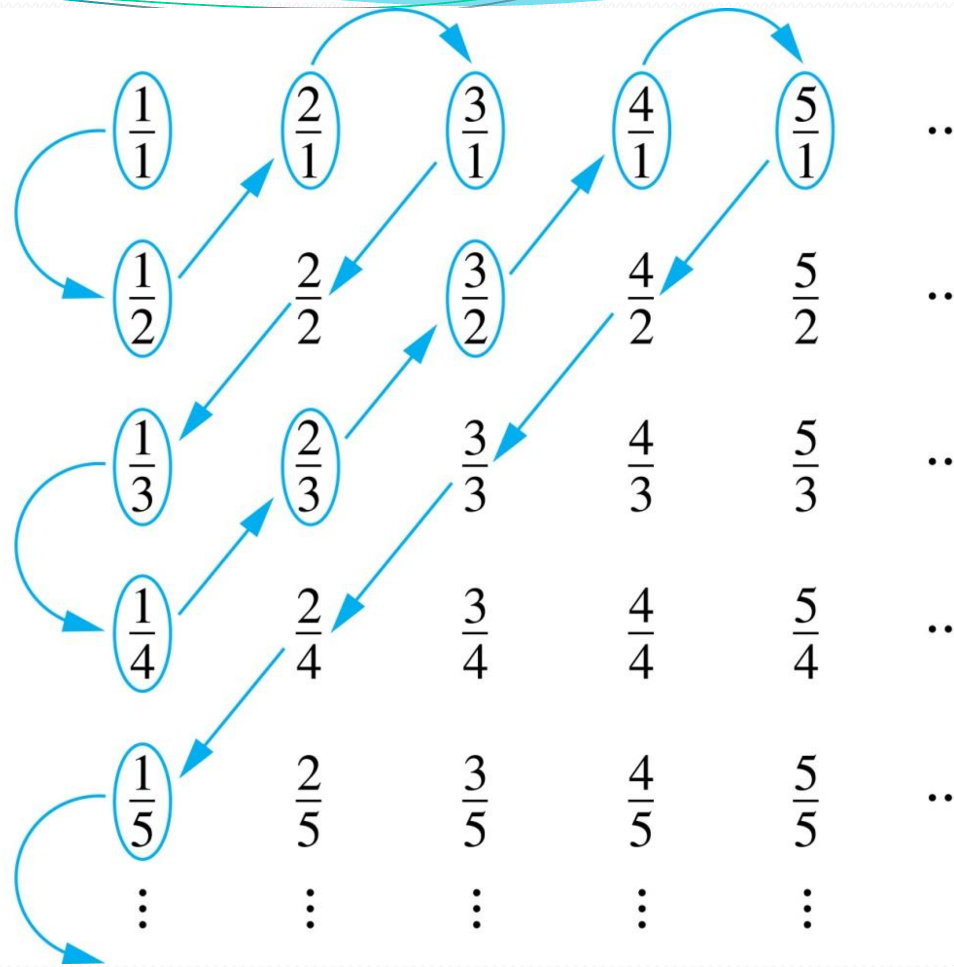
The next slide shows how this is done. →

Q^+ is Countable

Study table carefully.

- What is common in each row?
- What is common in each column?

Terms not circled are not listed because they repeat previously listed terms



Constructing List

First visit p/q with $p + q = 2$.

Next visit p/q with $p + q = 3$.

And so on.

1, ½, 2, 3, 1/3, ¼ , 2/3, 3/2, 4, 5, 1/5, 1/6, 2/5, ¾, 4/3, 5/2, 6, ….

Coming up with an explicit formula would not be advised.

Instead write a computer program!

Strings

Given finite alphabet *A*, show S = {finite strings} is countably infinite.

Solution: Assume alphabetical ordering of symbols in A and list strings in a sequence:

1. string of length 0 (empty string λ);
2. strings of length 1 in lexicographic (dictionary) order;
3. strings of length 2 in lexicographic order;
4. and so on.

This $\Rightarrow$ $\exists$bijection from $\mathbf{Z}^{+}$ to *S* & hence S is countably ∞.

◀

{all possible Java programs} is countably ∞.

Example: Show that the set of all Java programs is countable.

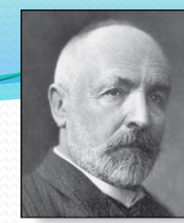
Solution: Let S = {strings} constructed from the characters which can appear in a Java program. Use ordering from previous example.

- Take each string in turn and feed into a compiler.
- Compiler will determine if input program is syntactically correct.
  - If compiler says YES, we add program to list.
- We move on to the next string.

In this way we construct an implied bijection from **N** to the set of Java programs. Hence, the set of Java programs is countable. ◀

R is Uncountable!

Georg Cantor
(1845-1918)



Proof method is *Cantor diagonalization* and is by contradiction.
We focus on subset [0, 1] & assume that its elements are listable:

$$r_1, r_2, r_3, \ldots .$$

Let decimal representation of this listing be

$$\begin{aligned} r_1 &= 0.d_{11}d_{12}d_{13}d_{14}d_{15}d_{16}\ldots \\ r_2 &= 0.d_{21}d_{22}d_{23}d_{24}d_{25}d_{26}\ldots \\ r_3 &= 0.d_{31}d_{32}d_{33}d_{34}d_{35}d_{36}\ldots \\ &\vdots \end{aligned}$$

Form new real # $r = .r_1r_2r_3r_4\ldots$ where

$$r_i = 3 \text{ if } d_{ii} \neq 3 \quad \text{and} \quad r_i = 4 \text{ if } d_{ii} = 3$$

r is not equal to any of the $r_1, r_2, r_3, \ldots$ because it differs from r_i in its i^{th} position after the decimal point.

$\therefore\ \exists\ r \in [0, 1]$ that is not on list (every real # has ! decimal expansion).
Hence, [0, 1] cannot be listed, & [0, 1] is uncountable.
Since a set with an uncountable subset is uncountable (exercise):

R is uncountable! ◀

Computability

We say that a function is *computable* if ∃ a computer program in some programming language that finds the values of this function.

- If a function is not computable we say it is *uncomputable*.
- ∃uncomputable functions.
  - We have shown that the set of Java programs is countable.
  - Exercise 38 in text shows that ∃uncountably many different functions from a particular countably infinite set to itself.
  - ∴ by Exercise 39, ∃ uncomputable functions!