Section 1.7
Sequences, Summations Cardinality of Infinite Sets

Definition: A *sequence* is a function from a subset of the natural numbers (usually of the form {0, 1, 2, . . . } to a set S.

Note: the sets

$$\{0, 1, 2, 3, . . . , k\}$$

and

$$\{1, 2, 3, 4, . . . , k\}$$

are called *initial segments* of N.

Notation: if f is a function from {0, 1, 2, . . .} to S we usually denote f(i) by a_i and we write

$$\{a_0, a_1, a_2, a_3, . . . \} = \{a_i\}_{i=0}^{k} = \{a_i\}_0^k$$

where k is the upper limit (usually ∞).

Examples:

Using zero-origin indexing, if f(i) = 1/(i + 1). then the sequence

$$f = \{1, 1/'2, 1/3, 1/4, . . . \} = \{a_0, a_1, a_2, a_3, . . \}$$

Using one-origin indexing the sequence f becomes

$$\{1/2, 1/3, \ldots\} = \{a_1, a_2, a_3, \ldots\}$$

Summation Notation

Given a sequence $\{a_i\}_0^k$ we can add together a subset of the sequence by using the summation and function notation

$$a_{g(m)} + a_{g(m+1)} + \ldots + a_{g(n)} = \sum_{j=m}^{n} a_{g(j)}$$

or more generally

$$\sum_{j \in S} a_j$$

Examples:

$$r^0 + r^1 + r^2 + r^3 + \ldots + r^n = \sum_0^n r^j$$

$$1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \ldots = \sum_1^\infty \frac{1}{i}$$

$$a_{2m} + a_{2(m+1)} + \ldots + a_{2(n)} = \sum_{j=m}^{n} a_{2j}$$

If S = {2, 5, 7, 10} then $\sum_{j \in S} a_j = a_2 + a_5 + a_7 + a_{10}$

Similarly for the *product* notation:

$$\prod_{j=m}^{n} a_j = a_m a_{m+1} \ldots a_n$$

Definition: A *geometric progression* is a sequence of the form

$$a, ar, ar^2, ar^3, ar^4, \ldots .$$

Your book has a proof that

$$\sum_{i=0}^{n} r^i = \frac{r^{n+1} - 1}{r - 1} \text{ if } r \neq 1$$

(you can figure out what it is if $r = 1$).

You should also be able to determine the sum

- if the index starts at k vs. 0
- if the index ends at something other than n (*e.g.*, n-1, n+1, etc.).

Cardinality

Definition: The cardinality of a set A is equal to the cardinality of a set B, denoted | A | = | B |, if there exists a bijection from A to B.

Definition: If a set has the same cardinality as a subset of the natural numbers N, then the set is called *countable.*

If |A| = |N|, the set A is *countably infinite*.

The (transfinite) cardinal number of the set N is

$$\textit{aleph null} = \aleph_0.$$

If a set is not countable we say it is *uncountable.*

Examples:

The following sets are uncountable (we show later)

- The real numbers in [0, 1]
- P(N), the power set of N

Note: With infinite sets proper subsets can have the same cardinality. This cannot happen with finite sets.

Countability carries with it the implication that there is a *listing* of the elements of the set.

Definition: | A | ≤ | B | if there is an injection from A to B.

Note: as you would hope,

Theorem: If | A | ≤ | B | and | B | ≤ | A | then | A | = | B |.

This implies

- if there is an injection from A to B
- if there is an injection from B to A

then

- there must be a bijection from A to B

This is <u>difficult</u> to prove but is an example of demonstrating existence without construction.

It is often easier to build the injections and then conclude the bijection exists.

Example:

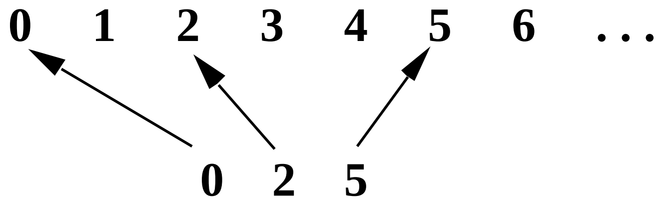
Theorem: If A is a subset of B then | A | ≤ | B |.

Proof: the function f(x) = x is an injection from A to B.

Example:

| {0, 2, 5} | ≤ À$_0$

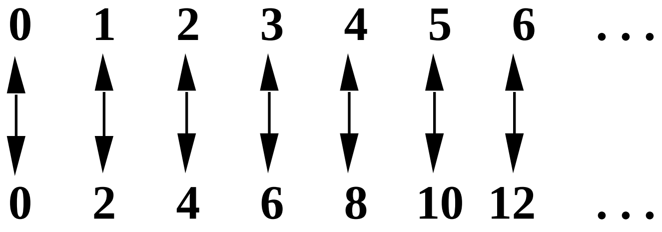
The injection f: {0, 2, 4} → N defined by f(x) = x is shown below:



Some Countably Infinite Sets

- The set of even integers E (0 is considered even) is countably infinite. Note that E is a proper subset of N!

Proof: Let f(x) = 2x. Then f is a bijection from N to E



- Z^+, the set of positive integers is countably infinite.

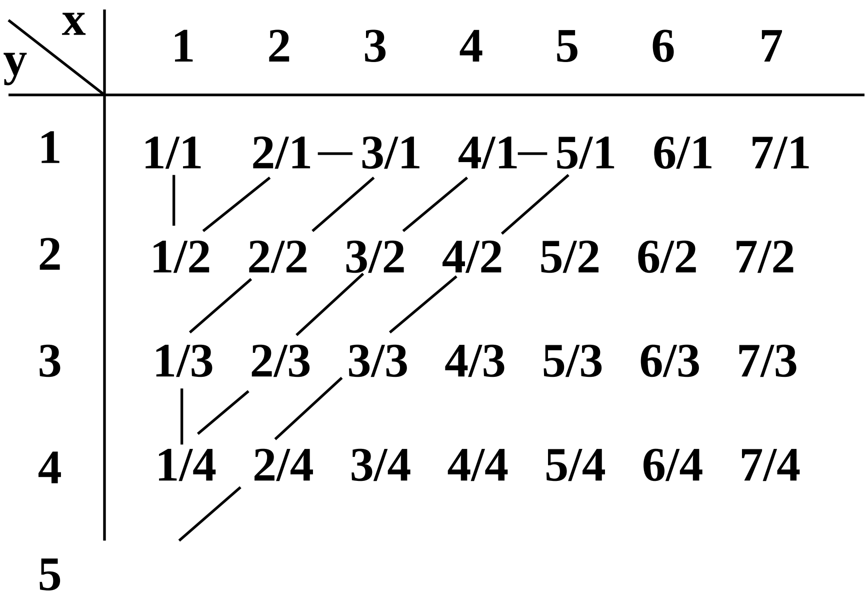
- The set of positive rational numbers Q^+ is countably infinite.

Proof: Z+ is a subset of Q^+ so $| Z^+ | = \aleph_0 \leq | Q^+ |$.

Now we have to show that $| Q^+ | \leq \aleph_0$.

To do this we show that the positive rational numbers with repetitions, Q_R, is countably infinite.

Then, since Q^+ is a subset of Q_R, it follows that $| Q^+ | \leq \aleph_0$ and hence $| Q^+ | = \aleph_0$.



y \ x	1	2	3	4	5	6	7
1	**1/1**	**2/1**	**3/1**	**4/1**	**5/1**	**6/1**	**7/1**
2	**1/2**	**2/2**	**3/2**	**4/2**	**5/2**	**6/2**	**7/2**
3	**1/3**	**2/3**	**3/3**	**4/3**	**5/3**	**6/3**	**7/3**
4	**1/4**	**2/4**	**3/4**	**4/4**	**5/4**	**6/4**	**7/4**
5							

The position on the path (listing) indicates the image of the bijective function f from N to Q_R:

f(0) = 1/1, f(1) = 1/2, f(2) = 2/1, f(3) = 3/1, and so forth.

Every rational number appears on the list at least once, some many times (repetitions).

Hence, $| N | = | Q_R | = \aleph_0$.

Q. E. D.

- The set of all rational numbers Q, positive and negative, is countably infinite.

- The set of (finite length) strings S over a finite alphabet A is countably infinite.

To show this we assume that

- A is nonvoid
- There is an “alphabetical” ordering of the symbols in A

Proof: List the strings in lexicographic order:

- all the strings of zero length,
- then all the strings of length 1 in alphabetical order,
- then all the strings of length 2 in alphabetical order,

etc.

This implies a bijection from N to the list of strings and hence it is a countably infinite set.

For example: Let A = {a, b, c}.

Then the lexicographic ordering of A is

{λ , a, b, c, aa, ab, ac, ba, bb, bc, ca, cb, cc, aaa, aab, aac, aba, ...} = {f(0), f(1), f(2), f(3), f(4), ...}

- The set of all C programs is <u>countable</u>.

Proof: Let S be the set of legitimate characters which can appear in a C program.

- A C compiler will determine if an input program is a syntactically correct C program (the program doesn't have to do anything useful).

- Use the lexicographic ordering of S and feed the strings into the compiler.

  - If the compiler says YES, this is a syntactically correct C program, we add the program to the list.

  - Else we move on to the next string.

In this way we construct a list or an implied bijection from N to the set of C programs.

Hence, the set of C programs is countable.

Q. E. D.

Cantor Diagonalization

- An important technique used to construct an object which is not a member of a countable set of objects with (possibly) infinite descriptions

Theorem: The set of real numbers between 0 and 1 is uncountable.

Proof: We assume that it is countable and derive a contradiction.

If it is countable we can list them (*i.e.*, there is a bijection from a subset of N to the set).

We show that no matter what list you produce we can construct a real number between 0 and 1 which is not in the list.

Hence, there cannot exist a list and therefore the set is not countable

It's actually much bigger than countable. It is said to have the *cardinality of the continuum,* **c**.

Represent each real number in the list using *its decimal expansion.*

e.g., 1/3 = .3333333........
1/2 = .5000000........
= .4999999........

If there is more than one expansion for a number, it doesn't matter as long as our construction takes this into account.

THE LIST....

$$r_1 = .d_{11}d_{12}d_{13}d_{14}d_{15}d_{16}. \; . \; . \; . \; .$$
$$r_2 = .d_{21}d_{22}d_{23}d_{24}d_{25}d_{26} \; . \; . \; . \; .$$
$$r_3 = .d_{31}d_{32}d_{33}d_{34}d_{35}d_{36} \; . \; . \; . \; .$$

.
.
.

Now construct the number $x = .x_1x_2x_3x_4x_5x_6x_7. \; . \; . \; .$

$$x_i = 3 \text{ if } d_{ii} \neq 3$$
$$x_i = 4 \text{ if } d_{ii} = 3$$

(Note: choosing 0 and 9 is not a good idea because of the non uniqueness of decimal expansions.)

Then x is not equal to any number in the list.

Hence, no such list can exist and hence the interval (0,1) is uncountable.

Q. E. D.

An extra goody:

Definition: a number x between 0 and 1 is *computable* if there is a C program which when given the input i, will produce the ith digit in the decimal expansion of x.

Example:

The number 1/3 is computable.

The C program which always outputs the digit 3, regardless if the input, computes the number.

Theorem: There is exists a number x between 0 and 1 which is ***not computable***.

There *does not exist* a C program (or a program in any other language) which will compute it!

Why? Because there are more numbers between 0 and 1 than there are C programs to compute them.

(in fact there are ***c*** such numbers!)

Our second example of the <u>nonexistence</u> of programs to compute things!
