Introduction to the Theory of Computation

Jingde Cheng

Saitama University

Introduction to the Theory of Computation

Enumerable (Countable) Sets

♣ Enumerable (countable) set 可枚举(可数)集

- [D] An *enumerable (countable) set* is one whose members can be enumerated: *arranged in a single list* with a first entry, a second entry, and so on, such that *every member of the set appears sooner or later on the list as the nth entry, for some finite natural number n*.

n必然有穷.

♣ Examples

- The set of positive integers less than 10 is enumerated by the list: 1, 2, 3, …, 9
- The set *N* of natural numbers is enumerated by the list: 0, 1, 2, 3, …, *n*, …
- The set of negative integers is enumerated by the list: −1, −2, −3, …, −*n*, …

Enumerable (Countable) Sets

♣ Notes

- The entries in these lists are not numbers but numerals, or names of numbers.
- In general, in listing the members of a set you manipulate names, not the things named.
- We regard the empty set, ∅, as enumerable. 空集也是可枚举的
- An enumerable (countable) set may be finite or infinite. 可枚举集可能是有穷也可能是无穷.

Enumerably Infinite (Denumerable) Sets

♣ Enumerably infinite (denumerable) set

正整数→该集上有满射(或更严格的双射)

- [D] An infinite set that is enumerable is said to be *enumerably infinite* or *denumerable*.
- [D] An infinite set is said to be *enumerably infinite* or *denumerable* IFF there is a surjection (or more strongly bijection) from the set *P* of positive integers to that set.

♣ Examples

- The set *P* of positive integers can be enumerated/arranged in a single infinite list: 1, 2, 3, 4, 5, 6, …, *n*, …
- The following is not acceptable as a list to enumerate/arrange the set *P* of positive integers: 1, 3, 5, ..., 2, 4, 6, ... (Why?)

2的位置n不是有穷的

有无穷个奇数

Enumerating Sets by Functions

♣ Arrangement 排列

- We might define an enumeration of a set not as a listing, but as an *arrangement* in which each member of the set is associated with one of the positive integers 1, 2, 3, … .

♣ Enumerating sets by functions

- An infinite list of a set determines a function *f* that takes positive integers as arguments, say *n*, and takes members of the set as values, say *f(n)*.
- Then we may speak of sets as being enumerated by functions, as well as by lists.
- Ex.: Enumerating the odd positive integers 1, 3, 5, 7, … by the function $f(n) = 2n - 1$; enumerating the even positive integers 2, 4, 6, 8, … by the function $f(n) = 2n$.

from set: 正整数 to set: 该集合.

Enumerating Sets by Functions

- The identity function 恒等函数
  - Instead of enumerating the set *P* of all positive integers by the list 1, 2, 3, 4, …, we may enumerate *P* by the function that assigns to each positive integer *n* the value *n* itself.
  - [D] *Identity function*: *id(n)* = *n* for each positive integer *n*.
- Enumerating sets by functions
  - [D] An infinite set is said to be *enumerably infinite* or *denumerable* IFF there is a surjection (or more strongly bijection) from the set *P* of positive integers to that set.

Enumerating Sets by Multiple Functions

- Enumerating the set *P* of positive integers by another function
  - The set *P* can be enumerated by the function *g* determined by the following list: 2, 1, 4, 3, 6, 5, ...
  - This list is obtained from the list 1, 2, 3, 4, 5, 6, … by interchanging entries in pairs: 1 with 2, 3 with 4, and so on.
  - This list is a strange but perfectly acceptable enumeration of the set *P*: every positive integer shows up in it, sooner or later.
  - The corresponding function, *g*, can be defined as follows: $g(n) = n + 1$ if n is odd, $g(n) = n - 1$ if n is even.
  - The function *g* does indeed associate one and only one member of *P* with each positive integer *n*. And the function *g* so defined does indeed enumerate *P*: For each member *m* of *P* there is a positive integer *n* for which $g(n) = m$.

有限的换位也是可枚举的.

Requirements for Enumerating Sets (Bags)

- Enumerating the set (bag) allows repetitions 允许重复
  - In enumerating a set by listing its members, it is perfectly all right if a member of the set shows up more than once on the list (Therefore, the list may be not a set but a bag, malti-set).
  - The requirement is rather that each member show up at least once. It does not matter if the list is redundant: All we require is that it be complete (a redundant list can always be thinned out to get an irredundant list).
- Enumerating the set (bag) allows repetitions: Example
  - A flawless enumeration of the positive integers are given by the following repetitive list: 1, 1, 2, 2, 3, 3, 4, 4, … .

Requirements for Enumerating Sets

- Enumerating the set allows gaps 允许有空.
  - It is also perfectly all right if a list has gaps in it, since one could go through and close up the gaps.
  - The requirement is that every element of the set being enumerated be associated with SOME positive integer, not that every positive integer have an element of the set associated with it.
- Enumerating the set allows gaps: Example
  - A flawless enumeration of the positive integers are given by the following gappy list: 1, −(undefined), 2, −, 3, −, 4, −, …
  - The function corresponding to the above list (call it *h*) assigns values corresponding to the first, third, … entries, but assigns no values corresponding to the gaps (second, fourth, … entries); *h* is a partial function of positive integers.

Requirements for Enumerating Sets

- Enumerating the set allows gaps: Example
  - A flawless enumeration of the positive integers are given by the following gappy list: 1, −(undefined), 2, −, 3, −, 4, −, … .
  - The function corresponding to the above list (call it *h*) assigns values corresponding to the first, third, … entries, but assigns no values corresponding to the gaps (second, fourth, … entries); *h* is a partial function of positive integers.
  - $$h(n) = \begin{cases} (n+1)/2 & \text{if } n \text{ is odd} \\ \text{undefined} & \text{otherwise.} \end{cases}$$

Enumerating Sets by Partial Functions

- Enumerating sets of positive integers by a partial function
  - Any set *S* of positive integers is enumerated quite simply by a partial function *s*, which is defined as follows:
    $$s(n) = \begin{cases} n & \text{if } n \text{ is in the set S} \\ \text{undefined} & \text{otherwise.} \end{cases}$$
  - Fact: Every set of positive integers is enumerable.
- Enumerating sets by functions
  - To say that a set *A* (of anything) is enumerable is to say that there is a function all of whose arguments are positive integers and all of whose values are members of *A*, and that each member of *A* is a value of this function: For each member *a* of *A* there is at least one positive integer *n* to which the function assigns *a* as its value (Any function satisfying the condition is OK).

Enumerable Sets: Examples

♣ Ex1: The set of integers

- The simplest list is 0, 1, −1, 2, −2, 3, −3, …
- Then if the corresponding function is called *f*, we have *f*(1) = 0, *f*(2) = 1, *f*(3) = −1, *f*(4) = 2, *f*(5) = −2, and so on.
- *f*(*n*) = −(*n* − 1)/2 if *n* is odd, *f*(*n*) = *n*/2 if *n* is even.

正整数的可枚举时.

Enumerating Pairs of Positive Integers (Important!)

♣ Ex2: The set of ordered pairs of positive integers (the 1st way)

- We organize the set of ordered pairs of positive integers into a rectangular array, and then traverse the array in *Cantor's zig-zag manner* to get the list:
  (1,1), (1,2), (2,1), (1,3), (2,2), (3,1), (1,4), (2,3), (3,2), (4,1), …
- If we call the function involved here *G*, then we have *G*(1) = (1,1), *G*(2) = (1,2), *G*(3) = (2,1), and so on.
- Note: Enumerating pairs of positive integers is very important and useful.

Enumerating Pairs of Positive Integers (Cantor's zig-zag Manner)

♣ Cantor's zig-zag manner 康托方式

- First comes the pair the sum of whose entries is 2, then come the pairs the sum of whose entries is 3, then come the pairs the sum of whose entries is 4, and so on.
- Within each block of pairs whose entries have the same sum, pairs appear in order of increasing the first entry.
- (1,1), (1,2), (2,1), (1,3), (2,2), (3,1), (1,4), (2,3), (3,2), (4,1), …

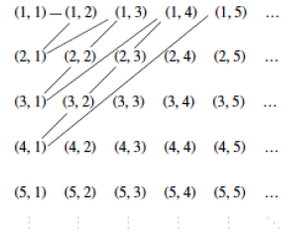

Figure 1-1. Enumerating pairs of positive integers.

Enumerable Sets: Examples

♣ Ex2: The set of ordered pairs of positive integers (the 2nd way)

- Let us use up every other place in listing the pairs (1, *n*), every other place then remaining in listing the pairs (2, *n*), every other place then remaining in listing the pairs (3, *n*), and so on.
- The result is:
  (1,1), (2,1), (1,2), (3,1), (1,3), (2,2), (1,4), (4,1), (1,5), (2,3), …
- (1,1), …, (1,2), …, (1,3), …, (1,4), …, (1,5), …, (1,6), …, (1,7),
  (2,1) (2,2) (2,3)
  (3,1) (3,2)
  (4,1)
- If we call the function involved here *g*, then *g*(1) = (1, 1), *g*(2) = (2, 1), *g*(3) = (1, 2), and so on.

Enumerating Sets by Functions: Decoding and Encoding

♣ Code number 编码数

- Given a function *f* enumerating the pairs of positive integers, such as *G* or *g* above, an *a* such that *f*(*a*) = (*m*, *n*) may be called a *code number* for the pair (*m*, *n*).

♣ Decoding and Encoding 编码、解码

- Applying the function *f* may be called *decoding*, while going the opposite way, from the pair to a code for it, may be called *encoding*.
- Formulas for the encoding functions *J* and *j* that go with the decoding functions *G* and *g* above (HW: Prove them):
  $J(m, n) = (m^2 + 2mn + n^2 - m - 3n + 2)/2$
  (Ex.: *J*(3,2) = 9, *G*(9) = (3,2))
  $j(m, n) = 2^{m-1}(2n - 1)$ (Ex.: *j*(3,2) = 12, *g*(12) = (3,2))

Enumerable Sets: Examples

♣ Ex3: The set of positive rational numbers

- A positive rational number is a number that can be expressed as a ratio of positive integers, that is, in the form *m*/*n* where *m* and *n* are positive integers.
- Therefore we can get an enumeration of all positive rational numbers by starting with our enumeration of all pairs of positive integers and replacing the pair (*m*, *n*) by the rational number m/n.
- The list: 1/1, 1/2, 2/1, 1/3, 2/2, 3/1, 1/4, 2/3, 3/2, 4/1, …

♣ Ex4: The set of rational numbers

- Combining the ideas of examples of integers and positive rational numbers.

正有理数的集合是可枚举的

有理数集是可枚举的.

Enumerable Sets: Examples

正整数的有序三元组集.

♣ Ex5: The set of ordered triples of positive integers

- Based on the list of ordered pairs of positive integers (1, 1), (1, 2), (2, 1), (1, 3), (2, 2), (3, 1), …, we go through this list, and in each pair replace the second entry or component n with the pair that appears in the nth place on this very list; in other words, replace each 1 that appears in the second place of a pair by (1, 1), each 2 by (1, 2), each 3 by (2, 1), and so on.
- This gives the list (1, (1, 1)), (1, (1, 2)), (2, (1, 1)), (1, (2, 1)), (2, (1, 2)), (3, (1, 1)), …, and that gives a list of triples (1, 1, 1), (1, 1, 2), (2, 1, 1), (1, 2, 1), (2, 1, 2), (3, 1, 1), …

Enumerable Sets: Examples

♣ Ex5: The set of ordered triples of positive integers

- In terms of functions, this enumeration may be described as follows.
- The original enumeration of pairs corresponds to a function associating to each positive integer n a pair $G(n) = (K(n), L(n))$ of positive integers.
- The enumeration of triples we have just defined corresponds to assigning to each positive integer n instead the triple $(K(n), K(L(n)), L(L(n)))$.
- We do not miss any triples (p, q, r) in this way, because there will always be an $m = J(q, r)$ such that $(K(m), L(m)) = (q, r)$, and then there will be an $n = J(p, m)$ such that $(K(n), L(n)) = (p, m)$, and the triple associated with this n will be precisely (p, q, r).

Enumerable Sets: Examples

有穷k元组

♣ Ex6: The set of ordered k-tuples of positive integers, for any fixed k

- The method by which we obtained an enumeration of triples from an enumeration of pairs can give us an enumeration of quadruples from an enumeration of triples.
- Go back to the original enumeration pairs, and replace each second entry n by the triple that appears in the nth place in the enumeration of triples, to get a quadruple.
- The first few quadruples on the list will be (1, 1, 1, 1), (1, 1, 1, 2), (2, 1, 1, 1), (1, 2, 1, 1), (2, 1, 1, 2), …
- We can go on from here to quintuples, sextuples, or k-tuples for any fixed k.

Enumerable Sets: Examples

♣ Ex7: The set of finite sequences of positive integers less than 10

- A finite sequence whose entries are all positive integers less than 10, such as (1, 2, 3), can be read as an ordinary decimal or base-10 numeral 123.
- The number this numeral denotes, one hundred twenty-three, could then be taken as a code number for the given sequence.
- Actually, for later purposes it proves convenient to modify this procedure slightly and write the sequence in reverse before reading it as a numeral. Thus (1, 2, 3) would be coded by 321, and 123 would code (3, 2, 1).
- In general, a sequence $s = (a_0, a_1, a_2, \ldots, a_k)$ would be coded by $a_0 + 10a_1 + 100a_2 + \ldots + 10^k a_k$ which is the number that the decimal numeral $a_k \ldots a_2 a_1 a_0$ represents.

进制的转换.

Enumerable Sets: Examples

♣ Ex8: The set of finite sequences of positive integers less than b, for any fixed b

- Generally, we can code a finite sequence $s = (a_0, a_1, a_2, \ldots, a_k)$ of positive integers less than b by $a_0 + ba_1 + b^2a_2 + \ldots + b^k a_k$.
- In general, working with base b, the ith entry (counting the initial one as the 0th) of the sequence coded by (b, n) (i.e., n in number system based b) will be entry(i, n) = rem(quo(n, b^i), b) where quo(x, y) and rem(x, y) are the quotient and remainder on dividing x by y.
- Example: When working with base 12, to obtain the 5th entry of the sequence coded by 123 456 789, we divide 123 456 789 by 12^5 to get the quotient 496. Now divide by 12 to get remainder 4.

Enumerable Sets: Examples

正整数的有穷序列.

♣ Ex9: The set of finite sequences of positive integers (1st way)

- Coding finite sequences is important.
- Let $G_1(n)$ be the 1-term sequence (n) (i.e., (1), (2), (3), …). Let $G_2 = G$ (i.e., $G(a) = (m, n)$), the function enumerating all 2-tuples or pairs from Ex2. Let G_3 ($G(a) = (p, q, r)$) be the function enumerating all triples as in Ex5. Let G_4, G_5, …, be the enumerations of quadruples, 5-tuple, …, from Ex6.
- We can get a coding of all finite sequences by pairs of positive integers by letting any sequence s of length k be coded by the pair (k, a) where $G_k(a) = s$.
- Since pairs of positive integers can be coded by single numbers, we indirectly get a coding of sequences of numbers.

Enumerable Sets: Examples

♣ **Ex9: The set of finite sequences of positive integers (2nd way)**

- Another way to describe what is going above is as follows.
- We go back to our original listing of pairs, (i.e., (1,1), (1,2), (2,1), (1,3), (2,2), (3,1), …), and replace the pair (*k*, *a*) by the *a*th item on the list of *k*-tuples.
- Thus (1, 1) would be replaced by the first item (1) on the list of 1-tuples (1), (2), (3), …; while (1, 2) would be replaced by the second item (2) on the same list; whereas (2, 1) would be replaced by the first item (1, 1) on the list of all 2-tuples or pairs; and so on.
- This gives us the list (1), (2), (1, 1), (3), (1, 2), (1, 1, 1), (4), (2, 1), (1, 1, 2), (1, 1, 1, 1), ...

Enumerable Sets: Examples

♣ **Ex9: The set of finite sequences of positive integers (2nd way)**

- Ex8 showed that we can code sequences of any length whose entries are less than some fixed bound, but what we now want to do is show how to code sequences of any length whose entries may be of any size.
- We take a sequence $s = (a_0, a_1, a_2, \ldots, a_k)$ to be coded by any pair (b, n) such that all a_i are less than b, and n codes s in the sense that $n = a_0 + ba_1 + b^2a_2 + \ldots + b^ka_k$.
- Since pairs of positive integers can be coded by single numbers, we indirectly get a coding of sequences of numbers.

Enumerable Sets: Examples

♣ **Ex9: The set of finite sequences of positive integers (3rd way)**

- Fundamental theorem of arithmetic (Unique prime factorization theorem): $n = p_1^{e1}p_2^{e2}\ldots p_r^{er}$, $p_1, p_2, \ldots, p_r$ are different prime numbers, $e1, e2, \ldots, er$ are natural numbers.
- Every integer greater than 1 can be written in one and only one way as a product of powers of larger and larger primes, a representation called its prime decomposition.
- We can code a sequence $s = (i, j, k, m, n, \ldots)$ by the number $2^i * 3^j * 5^k * 7^m * 11^n * \ldots$

算术基本定理.
(唯一预分解定理)

Enumerable Sets: Examples

♣ **Ex10: The set of finite sets of positive integers**

- It is easy to get an enumeration of finite sets of positive integers from an enumeration of finite sequences.
- Using the first method in Ex9, for instance, we get the following enumeration of sets of positive integers: {1}, {2}, {1, 1}, {3}, {1, 2}, {1, 1, 1}, {4}, {2, 1}, {1, 1, 2}, {1, 1, 1, 1}, ...
- The bag {1, 1} whose only elements are 1 and 1 is just the set {1} whose only element is 1, and similarly in other cases, so this list can be simplified to look like this: {1}, {2}, {1}, {3}, {1, 2}, {1}, {4}, {1, 2}, {1, 2}, {1}, {5}, ... The repetitions do not matter.

♣ **An important fact**

- The set of all sets of positive integers is not enumerable!

正整数集合的集合是不可枚举的

Enumerable Sets: Examples

♣ **Ex11: Any subset of an enumerable set**

- Given any enumerable set A and a listing of the elements of A: $a_1, a_2, a_3, \ldots$, we easily obtain a gappy listing of the elements of any subset B of A simply by erasing any entry in the list that does not belong to B, leaving a gap.

♣ **Ex12: The union of any two enumerable sets**

- Let A and B be enumerable sets, and consider listings of their elements: $a_1, a_2, a_3, \ldots, b_1, b_2, b_3, \ldots$.
- We obtain the following listing of the elements of the union A∪B: $a_1, b_1, a_2, b_2, a_3, b_3, \ldots$.
- If the intersection A∩B is not empty, then there will be redundancies on this list: If $a_m = b_n$, then that element will appear both at place 2m −1 and at place 2n, but this does not matter.

可枚举集的子集也可枚举

可枚举集的并集也可枚举

Enumerable Sets: Examples

♣ **Ex13: The set of finite strings from a finite or enumerable alphabet of symbols**

- Given an 'alphabet' of any finite number, or even an enumerable infinity, of symbols $S_1, S_2, S_3, \ldots$, we can take as a code number for any finite string $S_{a1}, S_{a2}, S_{a3}, \ldots, S_{ak}$ the code number for the finite sequence of positive integers $(a_1, a_2, a_3, \ldots, a_k)$ under any of the methods of coding considered in Ex9.
- For instance, with the ordinary alphabet of 26 symbols letters S_1 = 'A', S_2 = 'B', and so on, the string or word 'CAB' would be coded by the code for (3, 1, 2), which (on the third method of Ex9) would be $2^3 * 3 * 5^2 = 600$.

♣ **Ex14: The set of all finite subsets of an enumerable set**

- Home work: Show a proof.

有限/可枚举字符集构成的有限字符串的集合可枚举

可枚举集的有限子集的集合可数.

Equinumerosity (Equipollence, Equipotence)

♣ **Bijection** 双射

- A *bijection*, or *bijective function*, or "*one-to-one correspondence*" is a function which maps every element of the source on to every element of the target in a one-to-one relationship.
- A *bijection*, or *bijective function*, or "*one-to-one correspondence*" is an injective (one-to-one), surjective (on-to), and total function.

♣ **Equinumerous** 等势

- Two sets A and B are said to be *equinumerous (equipollent, equipotent)*, $A \sim B$, IFF there is a bijection/one-to-one correspondence between A and B.

有双射

Equinumerosity (Equipollence, Equipotence)

♣ **Properties of equinumerosity**

- Reflexivity: For any set A, $A \sim A$.
- Symmetry: If $A \sim B$, then $B \sim A$.
- Transitivity: If $A \sim B$ and $B \sim C$, then $A \sim C$.
- The equinumerous relation is an equivalence relation.

♣ **Fact**

- Any enumerable set is either finite or equinumerous with the set of all positive integers.

Diagonalization: An Example (A correspondence of N and Q)

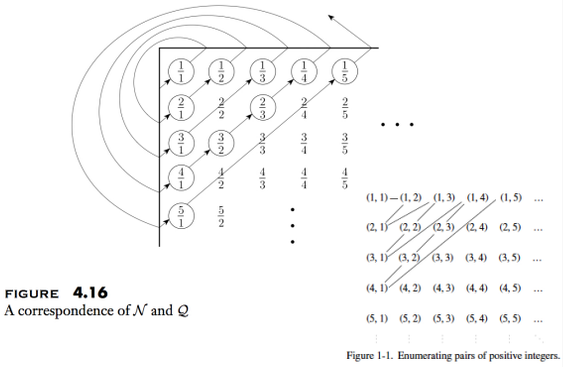

FIGURE **4.16**
A correspondence of $\mathcal{N}$ and $\mathcal{Q}$

Figure 1-1. Enumerating pairs of positive integers.

Diagonalization: An Example (A correspondence of N and Q)

♣ **The set of positive rational numbers is enumerable**

- Let $Q = \{ m/n \mid m, n \in N \}$ be the set of positive rational numbers.
- First, we make an infinite matrix containing all the positive rational numbers.
- The *ith* row contains all numbers with numerator *i* and the *j*th column has all numbers with denominator *j*.
- So the rational number *i*/*j* occurs in the *i*th row and *j*th column.

Diagonalization: An Example (A correspondence of N and Q)

♣ **The set of positive rational numbers is enumerable**

- Now we turn this matrix into a list that has a correspondence with N.
- We list the elements on the diagonals, which are superimposed on the diagram, starting from the corner.
- The first diagonal contains the single element 1/1, and the second diagonal contains the two elements 2/1 and 1/2. So the first three elements on the list are 1/1, 2/1, and 1/2.
- In the third diagonal, a complication arises. It contains 3/1, 2/2, and 1/3. If we simply added these to the list, we would repeat 1/1 = 2/2 . We avoid doing so by skipping an element when it would cause a repetition. So we add only the two new elements 3/1 and 1/3.
- Continuing in this way, we obtain a list of all the elements of Q.

Cantor's Theorem and Its Corollary

♣ **Cantor's Theorem** 康托定理

- The set of all sets of positive integers (i.e., the power set of the set of positive integers) is not enumerable.
- Let P the set of all positive integers, we denote the power set of P by P^*.
- Cantor's theorem says that P^* is not enumerable.

正整数的幂集不可枚举.

♣ **Corollary of Cantor's Theorem**

- The set of real numbers is not enumerable. 实数集不可枚举.

Proof of Cantor's Theorem and Diagonalization

♣ **Proof of Cantor's Theorem**

- Giving a method that can be applied to any list L of sets of positive integers in order to discover a set $\Delta(L)$ of positive integers which is not named in the list.
- If you then try to repair the defect by adding $\Delta(L)$ to the list as a new first member, the same method, applied to the augmented list L^* will yield a different set $\Delta(L^*)$ that is likewise not on the augmented list.
- Therefore, you can not find a list of all sets of positive integers that enumerates P^*.

♣ **Diagonalization**

- The method of the proof is called "*diagonalization method*", originally proposed by G. Cantor in 1837.

Proving Cantor's Theorem by Diagonalization

♣ **Defining $\Delta(L)$**

- Confronted with any infinite list L: $S_1, S_2, S_3, \ldots$, of sets of positive integers, we define a set $\Delta(L)$ as follows:
  (*) For each positive integer n, n is in $\Delta(L)$ IFF n is not in S_n.
- It should be clear that this genuinely defines a set $\Delta(L)$; for given any positive integer n, we can tell whether n is in $\Delta(L)$ if we can tell whether n is in the nth set in the list L.
- Thus, if S_3 happens to be the set E of even positive integers, the number 3 is not in S_3 and therefore it is in $\Delta(L)$.
- As the notation $\Delta(L)$ indicates, the composition of the set $\Delta(L)$ depends on the composition of the list L, so that different lists L may yield different sets $\Delta(L)$.

正整数n在Δ(L)里 ⇔n不在Sn里.

Proving Cantor's Theorem by Diagonalization

♣ **Showing $\Delta(L)$ is never in L by *reductio ad absurdum***

归谬法

- Suppose that $\Delta(L)$ does appear some where in list L, say as entry number m, and deduce a contradiction, thus showing that the supposition must be false.
- Supposition: For some positive integer m, $S_m = \Delta(L)$.
- [Note: Thus, if 100 is such an m, we are supposing that $\Delta(L)$ and S_{100} are the same set under different names: we are supposing that a positive integer belongs to $\Delta(L)$ IFF it belongs to the 100th set in list L.]
- To deduce a contradiction from this assumption we apply definition (*) to the particular positive integer m: with $n = m$, (*) tells us that m is in $\Delta(L)$ IFF m is not in S_m.
- Now a contradiction follows from our supposition: if S_m and $\Delta(L)$ are one and the same set we have m is in $\Delta(L)$ IFF m is in S_m. Therefore, the supposition must be false.

Proving Cantor's Theorem by Diagonalization

♣ **Showing $\Delta(L)$ is never in L by *reductio ad absurdum***

- For no positive integer m do we have $S_m = \Delta(L)$. In other words, the set $\Delta(L)$ is named nowhere in list L.
- Therefore, the method works. Applied to any list of sets of positive integers it yields a set of positive integers which was not in the list.
- Then no list enumerates all sets of positive integers: the set P^* of all such sets is not enumerable.
- This completes the proof.

Diagonalization

♣ **Looking at the proof again from a slightly different viewpoint**

- Accordingly, we think of the sets $S_1, S_2, S_3, \ldots$, as represented by functions $s_1, s_2, s_3, \ldots$ of positive integers that take the numbers 0 and 1 as values.
- The relationship between the set S_n and the corresponding function s_n is simply this: for each positive integer p we have: $s_n(p) = 1$ if p is in S_n, $s_n(p) = 0$ if p is not in S_n.
- Then the list can be visualized as an infinite rectangular array of zeros and ones, in which the nth row represents the function s_n and thus represents the set S_n.
- The nth row $s_n(1)\ s_n(2)\ s_n(3)\ \ldots$ is a sequence of zeros and ones in which the pth entry, $s_n(p)$, is 1 or 0 according as the number p is or is not in the set S_n (as shown in Figure 2-1).

Diagonalization [BBJ-ToC-07]

	1	2	3	4	
s_1	$s_1(1)$	$s_1(2)$	$s_1(3)$	$s_1(4)$	
s_2	$s_2(1)$	$s_2(2)$	$s_2(3)$	$s_2(4)$	
s_3	$s_3(1)$	$s_3(2)$	$s_3(3)$	$s_3(4)$	
s_4	$s_4(1)$	$s_4(2)$	$s_4(3)$	$s_4(4)$	
⋮	⋮	⋮	⋮	⋮	⋱

Figure 2-1. A list as a rectangular array.

Diagonalization

♣ Looking at the proof again from a slightly different viewpoint

- The entries in the diagonal of the array (upper left to lower right) form a sequence of zeros and ones: $s_1(1)\ s_2(2)\ s_3(3)\ s_n(4)$ …
- This sequence of zeros and ones (the *diagonal sequence*) determines a set of positive integers (the *diagonal set*).
- The diagonal set may well be among those listed in L, i.e., there may well be a positive integer d such that the set S_d is none other than our diagonal set.
- The sequence of zeros and ones in the dth row of Figure 2-1 would then agree with the diagonal sequence entry by entry: $s_d(1) = s_1(1), s_d(2) = s_2(2), s_d(3) = s_3(3), s_d(4) = s_4(4), \cdots$

Diagonalization

♣ Looking at the proof again from a slightly different viewpoint

- The diagonal set may or may not appear in the list L, depending on the detailed makeup of the list.
- What we want is a set we can rely upon not to appear in L, no matter how L is composed.
- Such a set lies near to hand: it is the *antidiagonal set*, which consists of the positive integers not in the diagonal set.
- The corresponding antidiagonal sequence is obtained by changing zeros to ones and ones to zeros in the diagonal sequence.

Diagonalization

♣ Looking at the proof again from a slightly different viewpoint

- We may think of this transformation as a matter of subtracting each member of the diagonal sequence from 1: we write the antidiagonal sequence as $1 - s_1(1), 1 - s_2(2), 1 - s_3(3), 1 - s_4(4)$, …
- This sequence can be relied upon not to appear as a row in Figure 2-1, for if it did appear — say, as the mth row — we should have: $s_m(1) = 1 - s_1(1), s_m(2) = 1 - s_1(2)$, …, $s_m(m) = 1 - s_m(m)$, …
- But the mth of these equations cannot hold ($s_m(m)$ must be zero or one. If zero, the mth equation says that 0 = 1. If one, the mth equation says that 1 = 0).

Diagonalization

♣ Looking at the proof again from a slightly different viewpoint

- Then the antidiagonal sequence differs from every row of our array, and so the antidiagonal set differs from every set in our list L.
- This is no news, for the antidiagonal set is simply the set $\Delta(L)$. We have merely repeated with a diagram Figure 2-1 our proof that $\Delta(L)$ appears nowhere in the list L.
- In fact, to call a set enumerable is simply to say that it is the range of some total or partial function of positive integers.
- To say that the set P^* is not enumerable is simply to deny the existence of any function of positive integers which has P^* as its range.

Another Proof of Cantor's Theorem by Diagonalization [LP-ToC-98]

Theorem 1.5.2: *The set $2^{\mathbf{N}}$ is uncountable.*

Proof: Suppose that $2^{\mathbf{N}}$ is countably infinite. That is, we assume that that there is a way of enumerating all members of $2^{\mathbf{N}}$ as

$$2^{\mathbf{N}} = \{R_0, R_1, R_2, \ldots\}$$

(notice that these are the sets R_n in the statement of the diagonalization principle, once we consider the relation $R = \{(i, j) : j \in R_i\}$). Now consider the set

$$D = \{n \in \mathbf{N} : n \notin R_n\}$$

(this is the the diagonal set). D is a set of natural numbers, and therefore it should appear somewhere in the enumeration $\{R_0, R_1, R_2, \ldots\}$ But D cannot be R_0, because it differs from it with respect to containing 0 (it does if and only if R_0 does not); and it cannot be R_1 because it differs from it with respect to 1; and so on. We must conclude that D does not appear on the enumeration at all, and this is a contradiction.

To restate the argument a little more formally, suppose that $D = R_k$ for some $k \geq 0$ (since D is a set of natural numbers, and $\{R_0, R_1, R_2, \ldots\}$ was supposed to be a complete enumeration of all such sets, such a k must exist). We obtain a contradition by asking whether $k \in R_k$:

(a) Suppose the answer is yes, $k \in R_k$. Since $D = \{n \in \mathbf{N} : n \notin R_n\}$, it follows that $k \notin D$; but $D = R_k$, a contradiction.
(b) Suppose the answer is no, $k \notin R_k$; then $k \in D$. But D is R_k, so $k \in R_k$, another contradiction.

We arrived at this contradiction starting from the assumption that $2^{\mathbf{N}}$ is countably infinite, and continuing by otherwise impeccably rigorous mathematical reasoning; we must therefore conclude that this asumption was in error. Hence $2^{\mathbf{N}}$ is uncountable. ■

Proof of Corollary of Cantor's Theorem

♣ Corollary of Cantor's Theorem

- The set of real numbers is not enumerable.
- We show that an enumeration of the set K of all real numbers ξ with $0 < \xi < 1$ would immediately give rise to an enumeration of the set P^* of all sets of positive integers.
- If ξ is a real number and $0 < \xi < 1$, then ξ has a decimal expansion $0.x_1x_2x_3\ldots$ where each x_i is one of the cyphers 0 - 9.
- Then associate to all ξ the set of all sets of positive integers n such that n is in a set of positive integers IFF a 1 appears in the nth place in the expansion of ξ.
- Thus, every set of positive integers is associated to some real number ξ (the sum of 10^{-n} for all n in the set).
- Therefore, an enumeration of the set K would immediately give rise to an enumeration of the set P^*, which cannot exist, by the preceding Cantor's theorem.

Another Proof of Corollary of Cantor's Theorem [S-ToC-13]

THEOREM 4.17

$\mathcal{R}$ is uncountable.

PROOF In order to show that $\mathcal{R}$ is uncountable, we show that no correspondence exists between $\mathcal{N}$ and $\mathcal{R}$. The proof is by contradiction. Suppose that a correspondence f existed between $\mathcal{N}$ and $\mathcal{R}$. Our job is to show that f fails to work as it should. For it to be a correspondence, f must pair all the members of $\mathcal{N}$ with all the members of $\mathcal{R}$. But we will find an x in $\mathcal{R}$ that is not paired with anything in $\mathcal{N}$, which will be our contradiction.

The way we find this x is by actually constructing it. We choose each digit of x to make x different from one of the real numbers that is paired with an element of $\mathcal{N}$. In the end, we are sure that x is different from any real number that is paired.

We can illustrate this idea by giving an example. Suppose that the correspondence f exists. Let $f(1) = 3.14159\ldots$, $f(2) = 55.55555\ldots$, $f(3) = \ldots$, and so on, just to make up some values for f. Then f pairs the number 1 with $3.14159\ldots$, the number 2 with $55.55555\ldots$, and so on. The following table shows a few values of a hypothetical correspondence f between $\mathcal{N}$ and $\mathcal{R}$.

n	$f(n)$
1	3.14159...
2	55.55555...
3	0.12345...
4	0.50000...
⋮	⋮

Another Proof of Corollary of Cantor's Theorem [S-ToC-13]

We construct the desired x by giving its decimal representation. It is a number between 0 and 1, so all its significant digits are fractional digits following the decimal point. Our objective is to ensure that $x \neq f(n)$ for any n. To ensure that $x \neq f(1)$, we let the first digit of x be anything different from the first fractional digit 1 of $f(1) = 3.14159\ldots$. Arbitrarily, we let it be 4. To ensure that $x \neq f(2)$, we let the second digit of x be anything different from the second fractional digit 5 of $f(2) = 55.55555\ldots$. Arbitrarily, we let it be 6. The third fractional digit of $f(3) = 0.12345\ldots$ is 3, so we let x be anything different—say, 4. Continuing in this way down the diagonal of the table for f, we obtain all the digits of x, as shown in the following table. We know that x is not $f(n)$ for any n because it differs from $f(n)$ in the nth fractional digit. (A slight problem arises because certain numbers, such as $0.1999\ldots$ and $0.2000\ldots$, are equal even though their decimal representations are different. We avoid this problem by never selecting the digits 0 or 9 when we construct x.)

n	$f(n)$
1	3.14159...
2	55.55555...
3	0.12345...
4	0.50000...
⋮	⋮

$x = 0.4641\ldots$

Non-enumerable Sets: Examples

- **The set of all subsets of an infinite enumerable set is non-enumerable.**
- **The set of all real numbers is equinumerous with the set of all real numbers ξ with $0 < \xi < 1$, and therefore, it is non-enumerable.**
- **The set of points on a plane is equinumerous with the set of points on a line (i.e., the set of all real numbers), and therefore, it is non-enumerable.**
- **The set of points in space is equinumerous with the set of points on a line (i.e., the set of all real numbers), and therefore, it is non-enumerable.**

Diagonalization: Another Example [L-ToC-17]

THEOREM 11.1

Let S be an infinite countable set. Then its powerset 2^S is not countable.

Proof: Let $S = \{s_1, s_2, s_3, ...\}$. Then any element t of 2^S can be represented by a sequence of 0's and 1's, with a 1 in position i if and only if s_i is in t. For example, the set $\{s_2, s_3, s_6\}$ is represented by 01100100..., while $\{s_1, s_3, s_5, ...\}$ is represented by 10101.... Clearly, any element of 2^S can be represented by such a sequence, and any such sequence represents a unique element of 2^S. Suppose that 2^S were countable; then its elements could be written in some order, say $t_1, t_2, ...$, and we could enter these into a table, as shown in Figure 11.2. In this table, take the elements in the main diagonal, and complement each entry, that is, replace 0 with 1, and vice versa. In the example in Figure 11.2, the elements are 1100.... so we get 0011... as the result. The new sequence along the diagonal represents some element of 2^S, say t_i for some i. But it cannot be t_1 because it differs from t_1 through s_1. For the same reason it cannot be t_2, t_3, or any other entry in the enumeration. This contradiction creates a logical impasse that can be removed only by throwing out the assumption that 2^S is countable. ■

t_1	(1)	0	0	0	0	...
t_2	1	(1)	0	0	0	...
t_3	1	1	(0)	1	0	...
t_4	1	1	0	(0)	1	...
⋮					⋱	

FIGURE 11.2

Diagonalization: Another Example [HS-ToC-11]

Theorem 1.4. *The set of all functions from N to N is not countable.*

Proof. Let $A = \{f \mid f : N \to N\}$. Suppose A is countable. Then there is an enumeration $f_0, f_1, \ldots$ of A. (Think of all the values of each f_i laid out on an infinite matrix: The idea is to define a function that cannot be on this matrix because it differs from all of the values on the diagonal. This is illustrated in Fig. 1.3.) Define a function g by $g(x) = f_x(x) + 1$, for all $x \in N$. Then, g is a function on N, but observe that g cannot be in the enumeration of all functions. That is, if $g \in A$, then for some natural number k, $g = f_k$. But g cannot equal f_k because $g(k) \neq f_k(k)$. Thus, we have contradicted the assumption that the set of all functions can be enumerated. Thus, A is not countable. □

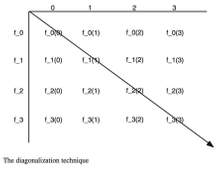

Fig. 1.3 The diagonalization technique

The Diagonalization Principle [LP-ToC-98]

♣ **The diagonalization principle**

The Diagonalization Principle: *Let R be a binary relation on a set A, and let D, the diagonal set for R, be $\{a : a \in A \text{ and } (a, a) \notin R\}$. For each $a \in A$, let $R_a = \{b : b \in A \text{ and } (a, b) \in R\}$. Then D is distinct from each R_a.*

If A is a finite set, then R can be pictured as a square array; the rows and columns are labeled with the elements of A and there is a cross in the box with row labeled a and column labeled b just in case $(a, b) \in R$. The diagonal set D corresponds to the complement of the sequence of boxes along the main diagonal, boxes with crosses being replaced by boxes without crosses, and vice versa. The sets R_a correspond to the rows of the array. The diagonalization principle can then be rephrased: the complement of the diagonal is different from each row.

- $D_{A,R} =_{df} \{ a \mid a \in A \wedge (a, a) \notin R \}$
- $R_a =_{df} \{ b \mid b \in A \wedge (a, b) \in R \}$

The Diagonalization Principle: An Example [LP-ToC-98]

Example 1.5.3: Let us consider the relation $R = \{(a,b),(a,d),(b,b),(b,c),(c,c),(d,b),(d,c),(d,e),(d,f),(e,e),(e,f),(f,a),(f,c),(f,d),(f,e)\}$; notice that $R_a = \{b,d\}$, $R_b = \{b,c\}$, $R_c = \{c\}$, $R_d = \{b,c,e,f\}$, $R_e = \{e,f\}$ and $R_f = \{a,c,d,e\}$. All in all, R may be pictured like this:

	a	b	c	d	e	f
a		×		×		
b		×	×			
c			×			
d		×	×		×	×
e					×	×
f	×		×	×	×	

The Diagonalization Principle: An Example [LP-ToC-98]

The sequence of boxes along the diagonal is

	×	×		×	

Its complement is

×			×		×

which corresponds to the diagonal set $D = \{a,d,f\}$. Indeed, D is different from each row of the array; for D, because of the way it is constructed, differs from the first row in the first position, from the second row in the second position, and so on.$\Diamond$

Introduction to the Theory of Computation

- Enumerability and Diagonalization
- Finite Automata and Regular Languages
- Context-Free Languages
- Computation: Turing Machines
- Computation: Turing-Computability (Turing-Decidability)
- Computation: Reducibility (Turing-Reducibility)
- Computation: Recursive Functions
- Computation: Recursive Sets and Relations
- Equivalent Definitions of Computability
- Advanced Topics in Computability Theory
- Computational Complexity
- Time Complexity
- Space Complexity
- Intractability
- Advanced Topics in Complexity Theory