Infinities in Mathematics and Computation

This lecture answers the following questions

- Are there different “infinities”?
- How does the number of mathematical functions compare with the number of computer programs (both are infinite)?
- Can we precisely specify a function that cannot be written as computer program (there are more functions than programs) ?

Proof by Contradiction

- Assume a statement is TRUE.
- By mathematical logic, deduce the consequences of such a statement.
- If a statement known to be FALSE (a contradiction) is deduced, the original statement must be FALSE.

So, to prove S is TRUE, assume S is FALSE and show that such an assumption leads to a contradiction: then, S is proved TRUE.

$\sqrt{2}$ is Irrational: a Proof by Contradiction

To prove $\sqrt{2}$ is irrational, assume the opposite: that it is rational and can therefore be written as p/q, where p and q are two integers that have NO common factors (this is important).

- $\sqrt{2} = p/q$ Assumed above
- $2 = p^2/q^2$ Square both sides
- $2q^2 = p^2$ Multiply by q^2
- p^2 is even It has a factor of 2
- p is even If p odd -> p^2 odd
- write $p = 2m$ p is even
- $2q^2 = (2m)^2$ Substitute 2m for p
- $2q^2 = 4m^2$ Expand $(2m)^2$
- $q^2 = 2m^2$ Divide by 2
- q^2 is even It has a factor of 2
- q is even If q odd -> q^2 odd

Contradiction: p and q are both even, so they have a common factor, 2.

Since a contradiction was reached, then the original assumption must be FALSE; therefore $\sqrt{2}$ cannot be written as p/q, so it is irrational.

Comparing Sizes of Finite Sets (let |X| denote the size of set X)

1) Count the elements

A = {a,b,c}

X = {x,y,z}

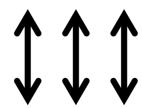
|A| = 3

|X| = 3

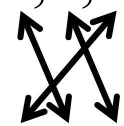
Therefore, |A| = |X|

2) Pair the elements

A = {a,b,c} {a,b,c}



or



X = {x,y,z} {x,y,z}

In a 1-1 mapping, every element in a set appears at the end of exactly 1 arrow. Therefore,

|A| = |X|

We do not need to know the actual size of either set to know they are the same size.

Comparing Sizes of Infinite Sets

Sets of Positive & Whole numbers have the same size:

P = {1, 2, 3, 4, 5, …}

↕ ↕ ↕ ↕ ↕

W={0, 1, 2, 3, 4, …}

P-to-W(x) = x-1

W-to-P(x) = x+1

Sets of Positive & Even numbers have the same size:

P = {1, 2, 3, 4, 5, …}

↕ ↕ ↕ ↕ ↕

E = {0, 2, 4, 6, 8, …}

P-to-E(x) = 2(x-1)

E-to-P(x) = (x+2) / 2

Comparing Sizes of Infinite Sets (continued)

Do sets of Positive numbers and Integers also have the same size?

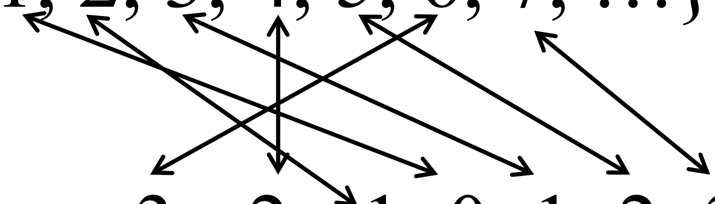
P = {1, 2, 3, 4, 5, 6, 7, …}

I = {…, -3, -2, -1, 0, 1, 2, 3, …}

Comparing Sizes of Infinite Sets (continued)

Do sets of Positive numbers and Integers also have the same size?

$P = \{1, 2, 3, 4, 5, 6, 7, \ldots\}$



$I = \{\ldots, -3, -2, -1, 0, 1, 2, 3, \ldots\}$

P-to-I(odd x) = (x-1) /2 I-to-P($_{x>=0}$ x) = 2x+1

P-to-I(even x) = (-x)/2 I-to-P($_{x<0}$ x) = -2x

The First Infinity: $\aleph_0$

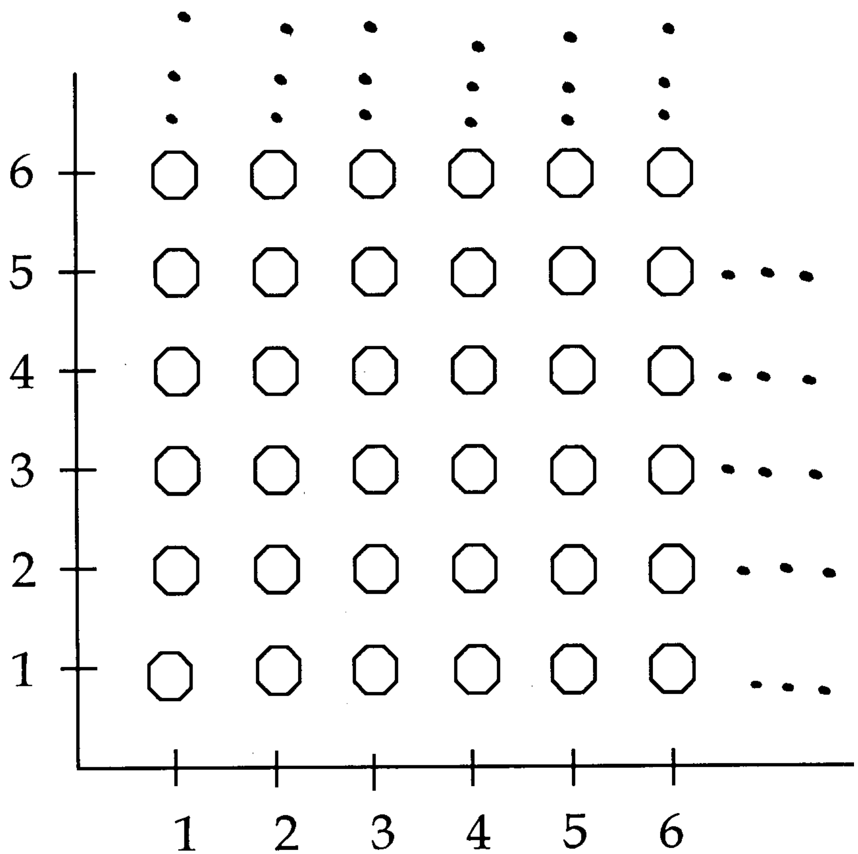
The sets of positive, whole, even, and integer numbers all have the same size

$|P| = |W| = |E| = |I| = \aleph_0$(aleph-naught)

Georg Cantor (1845-1918): "A set is infinite if its elements can be put into a 1-1 mapping with a proper subset of themselves."

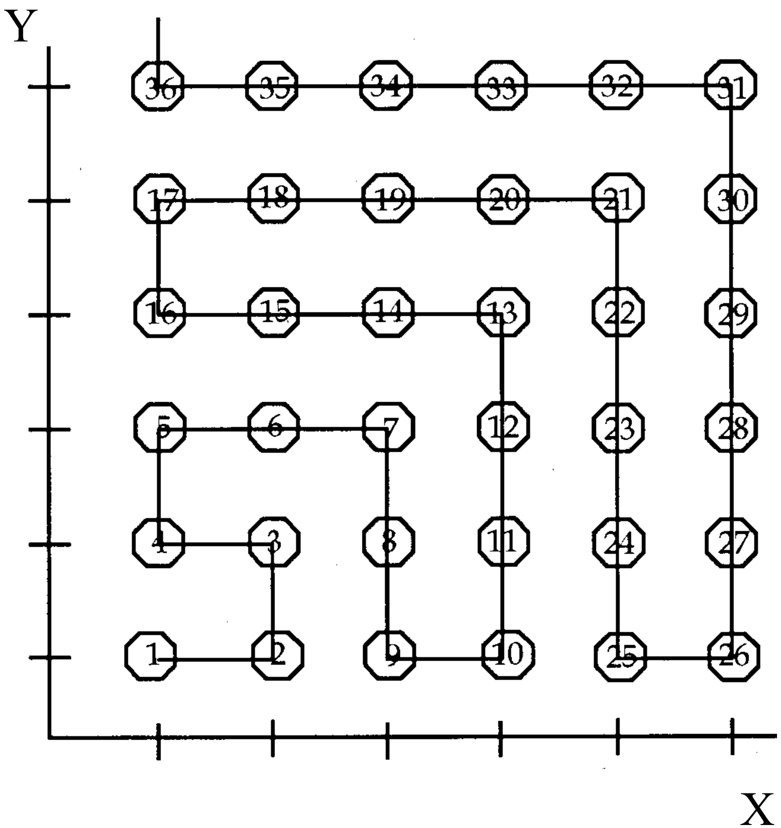
Dauben, *Georg Cantor: His Mathematics and Philosophy of the Infinite*, Princeton, 1979.

Rationals(Q): X_0 or Bigger?



Let Y / X represent the rational number at coordinate (X, Y). To show that |Q| = X_0, produce a "path" that systematically walks through every (X, Y) coordinate in this lattice: visit a 1st lattice point, a 2nd lattice point, a 3rd lattice point, …

Rationals(Q): X_0 or Bigger?



Let Y / X represent the rational number at coordinate (X, Y). Then the mapping is

{1, 2, 3, 4, 5, 6, 7, 8, …}

{1/1, 1/2, 2/2, 2/1, 3/1, 3/2, 3/3, 2/3, …}

Therefore, $|Q| = X_0$

Real (R): X_0 or Bigger

$|R| > X_0$: Proof by Contradiction (Diagonalization)
Assume there is a 1-1 Mapping from P to R[0,1]

1 ↔ .0 0 0 0 0 0 …
2 ↔ .5 0 0 0 0 0 …
3 ↔ .3 3 3 3 3 3 …
4 ↔ .6 9 3 1 4 7 …
5 ↔ .3 1 8 3 0 9 …
6 ↔ .1 0 1 0 0 1 …

Real (R): X_0 or Bigger

$|R| > X_0$: Proof by Contradiction (Diagonalization)

Assume there is a 1-1 Mapping from P to R[0,1]

	1	2	3	4	5	6	…
1 ↔	.0	0	0	0	0	0	…
2 ↔	.5	0	0	0	0	0	…
3 ↔	.3	3	3	3	3	3	…
4 ↔	.6	9	3	1	4	7	…
5 ↔	.3	1	8	3	0	9	…
6 ↔	.1	0	1	0	0	1	…
⋮							

We can construct a value V that differs from every value in this list. Make the i^{th} digit of V be 1+ (the i^{th} digit of the i^{th} number_, or 0 if the i^{th} digit is 9. For this mapping:

V = .114212…

So V is not on the list, leading to a contradiction, so there is no possible mapping.

We say $|R| = X_1$

The Continuum Hypothesis

In summary, $X_0 = |P| < |R| = X_1$

The Continuum Hypothesis (unproved):

> "There exists no set S such that
> $X_0 < |S| < X_1$"

Although the Continuum Hypothesis (CH) remains unproved, it has been proven that most of mathematics remains the same regardless of whether the CH is TRUE or FALSE.

R[0,1] x R[0,1]: $=X_1$ or Bigger?

R[0,1] x R[0,1]: = {(x,y) | x in [0,1] and y in [0,1]}

This set describes all points in a unit square.

Proof that |R[0,1] x R[0,1]| = X_1

Let (x,y) be written (.$x_1 x_2 x_3 x_4 x_5$ ..., .$y_1 y_2 y_3 y_4 y_5$ ...

Map (x,y) ↔ .$x_1\, y_1 x_2 y_2 x_3 y_3 x_4 y_4 x_5 y_5$

So |R[0,1] x R[0,1]| = |R| = X_1

English Statements(E): $\aleph_0$ or Bigger

Assume an alphabet with 26 letters, a space (written ~), and a period (written .); e.g., SEE~DICK~RUN.

1	A
2	B
…	
26	Z
27	~
28	.
29	AA
30	AB
…	
54	AZ
55	A~
56	A.
57	BA
…	
784	..
…	
6.5×10^{18}	SEE~DICK~RUN.

Thus, we can list all possible *statements* in the following order: first all one-letter *statements* in dictionary order then all two-letter *statements* in dictionary order, etc. mapping each positive number to a *statement*.

Therefore $|E| = \aleph_0$

Computer Programs (C): X_0 or Bigger?

Computer programs are written in a special alphabet that, like English, includes letters and punctuation. They can be considered *statements* written over this enlarged alphabet.

Therefore by the same reasoning process $|C| = X_0$

Mathematical Functions (M): X_0 or Bigger?

$|M| > X_0$: Look at functions mapping P to T/F

Assume there is a 1-1 Mapping from P to M

	(1)	(2)	(3)	(4)	(5)	(6)	...
f_1 ↔	F	F	F	F	F	F	...
f_2 ↔	T	T	T	T	T	T	...
f_3 ↔	T	F	T	F	T	F	...
f_4 ↔	F	F	F	T	T	T	...
f_5 ↔	F	F	F	T	F	F	...
f_6 ↔	F	T	T	F	T	F	...

We can construct a function f that differs from every f_i on this list. Make the ith value of f be the opposite of $f_i(i)$: e.g.

$$f(1) = T, f(2) = F, f(3) = F, \ldots$$

So $f(i)$ differs from every $f(i)$ and therefore is not on the list, leading to a contradiction, so there is no possible mapping

$$|M| > X_0$$

Mathematical Functions and Programs

$|C| < |M|$ so there are more mathematical functions than computer programs.

Therefore, some mathematical functions cannot be programmed on a computer.

Are there any "interesting" mathematical functions that cannot be programmed?

The Halting Problem

Does there exist a program H, which given any program P and data D determines whether or not P halts when run on D?

Let P(D) denote running program P on data D.

So H(P,D) is either T or F, depending on whether or not P(D) halts.

H itself must always halt and produce an answer telling whether P(D) halts.

Half Solving the Halting Problem

We can *almost* compute H by running program P on data D and returning T whenever P(D) halts; but such a function would never return a value if P(D) never halted. At some point an actual H would have to return F – when it *knew* that P(D) would never halt – if it could somehow know.

Proving the Halting Problem is Unsolvable

Assume H(P,D) exists as described; define G(x) = if H(x,x) then *loop forever* else *halt*;

Does G(G) halt?

If we assume it halts, we can prove it runs forever; if we assume it runs forever, we can prove it halts. Therefore, we have constructed a function G that cannot exist; therefore H cannot exist, because if H existed, we could easily construct G as described above.

H is a Powerful Theorem Prover

If H existed, we could use it as a powerful theorem prover in mathematics.

Fermat's Conjecture:

> "There are no integral solutions to the equation: $a^n + b^n = c^n$ (with n > 2)"

Write a program that generates every possible integral value for (a,b,c,n similar to generating rationals) and halts when $a^n + b^n = c^n$ and n>2 .

The program halts iff the conjecture is FALSE.

Computability References

- Davis, *Computability and Unsolvability*, Dover, 1973.
- Hopcroft & Ullman, *Introduction to Automata Theory, Languages, and Computability*, Addison Wesley, 1979.
- Minsky, *Computation: Finite and Infinite Machines*, Prentice hall, 1968.
- Rayward-Smith, *A First Course in Computability,* Blackwell, 1986.
- Walker, *The Limits of Computing,* Jones and Bartlett, 1994.