

Re: Different size infinities?

Source: <http://sci.tech-archive.net/Archive/sci.math/2004-10/8541.html>

From: HERC777 (herc777_at_hotmail.com)

Date: 10/29/04

Date: 29 Oct 2004 14:52:03 -0700

David C. Ullrich <ullrich@math.okstate.edu> wrote in message

> *On Fri, 29 Oct 2004 10:28:27 +0200, Han de Bruijn*

> *<Han.deBruijn@DTO.TUDELFT.NL> wrote:*

>

> *>Michael Barr wrote:*

> *>> Dave Seaman <dseaman@no.such.host> wrote in message news:<clpnj1*

> *>>>*

> *>>>>On 28 Oct 2004 02:07:05 GMT, R3769 wrote:*

> *>>>>*

> *>>>>>I was exposed to the idea in class today (in a cursory manner), that*

> *>>>>>there are smaller and larger infinities.*

> *>>>>>*

> *>>>>>The explanation had to do with the infinity of "space" (or partitions)*

> *>>>>>of the interval [0,1] as compared with infinity of integers 1, 2,*

> *>>>>>3....*

> *>>>>>*

> *>>>>>There is a proof of an infinity being larger than another type of*

> *>>>>>infinity?*

> *>>>>>*

> *>>>>>*

> *>>>>>Yes. In fact it's not hard to see there are an infinite number of different*

> *>>>>>types of infinities.*

> *>>>>*

> *>>>>Which infinite number would that be? :-)*

> *>>>*

> *>>>*

> *>>> And the answer is: None. More precisely, the class of infinite*

> *>>> cardinal numbers is not a set. There are more of them than can be*

> *>>> counted with any infinite number! And the variety of large infinite*

> *>>> cardinals is just astonishing.*

> *>>*

> *>>GAUSS: "I must protest most vehemently against the use of*

> *>>the infinite as something consummated, as this is never*

> *>>permitted in mathematics";*

> *>>*

> *>>KRONECKER: "I don't know what predominates in Cantor's*

> *>>theory – philosophy or theology, but I am sure that there*

> *>>is no mathematics there";*

sci.math: Re: Different size infinities?

> >
 > >*POINCARÉ: "There is no actual infinity; Cantorians forgot
 > >that and fell into contradictions. Later generations will
 > >regard Mengenlehre as a disease from which one has to be
 > >recovered";*
 > >
 > >*BROUWER: Cantor's theory as a whole is "a pathological
 > >incident in history of mathematics from which future
 > >generations will be horrified";*
 >
 > *This is very funny. It's like this was sci.chemistry
 > and I quoted four famous scientists from a century
 > or so ago doubting the existence of atoms. (Don't
 > know whether you've noticed, but those predictions
 > about how future generations would recoil in shock
 > and horror turned out to be simply wrong. What would
 > your point be in quoting people from a century ago
 > who turned out to be simply wrong about something?)*
 >
 > *As quoted from:*
 > >
 > > <http://www.mlahanas.de/Greeks/Infinite.htm>
 > >
 > >*Han de Bruijn*
 >
 >
 > *****
 >
 > *David C. Ullrich*

Atoms proved to be useful.

It is good to see some truth on the matter here. What is wrong with this proof?

From: HERC777

Subject: The Ten Turing Machines Real Number Counting Proof

Not everyone follows what I mean when I use

UTM(n, 0) neN

as the countable real list. UTM is any one of a special set of Turing Machines that can emulate *any* other Turing Machine.

A Turing Machine is essentially a computer program in the format of a binary finite state machine augmented with an infinite memory tape.

The smallest UTM is a large program several thousand states in size, it is very clumsy and slow. It has to read 2 numbers off

Re: Different size infinities?

the tape so it has to decode a ternary input constructed out of binary pairs of digits and also emulate a table of numbered states with its primitive binary operations. The input, working memory and the data structure of states (the program it's emulating) all have to be adjacent on the single infinite tape. Its not a practical device, no RAM.

But there are small UTMs, using different machines to TMs yet they are equivalent in the algorithms they can emulate.

The EVAL function in LISP (a functional language) is essentially a UTM. $\text{EVAL}(\text{"print"} + \text{" 10"}) \Rightarrow 10$. Functional languages were only theoretical until a student of John McCarthy suggested handcoding the eval function and the rest of LISP would run. Miranda uses a smaller subset of lambda calculus again, a set of about 10 'supercombinator' functions that can emulate any other function.

$S a b c = a c (b c)$

$K x y = x$

These 2 functions are equal in power to a Turing Machine.

$S K K x$

$= K x (K x)$

$= x$

I (the identity function) is also used but is not necessary since $SKK = I$ here.

There are other models of computation, the high order function $\text{TWICE}()$ that applies its function–argument to itself is thought of as the number 1. $\text{TWICE}(\text{TWICE}())$ is the number 2. Constructions of this one function yield addition and multiplication functions, in theory some function of the form $\text{TWICE}(\text{TWICE}((\text{TWICE})\text{TWICE}(\text{TWICE})))$ is a UTM. Neural nets are Turing equivalent, recursive functions, machine code, RISC reduced instruction set code, microcode, firmware, relational languages, reverse polish notation, Babbages original general function machine that used cogs, there are several classes and types of computation, numerous computer languages, often completely different in how they operate, and numerous ways to program a UTM on each, yet they are all equivalent in WHAT THEY COMPUTE.

Given this, it should be easy to find 10 different UTMs.

Each UTM is input one natural number at a time, from 1 onwards it emulates the 1st computer program and outputs the result, emulates the 2nd computer program and outputs the result...

UTM1(1, 0) = 7
UTM1(2, 0) = 555555555555
UTM1(3, 0) = ?
UTM1(4, 0) = 1111..

Now I'll briefly address the doorman arguments.

1/ We place a "0." at the start of the number and focus the demonstration

on reals between 0 and 1.

2/ We can also use digit places instead of 0 as the program argument, so the

program terminates for each digit. A cartesian counting method like ones

use for counting rationals can be used for multitasking infinite numbers with

infinite digits.

3/ UTM1(3, 0) hasn't terminated yet, will it ever? Scheduling the processes

in a multi tasking format this is of little concern. The point is whether the

set of all emulated programs can compute every real. We're not going to

achieve a standard bijection, not-halted-yet can be considered an 11th digit

for the purposes of diagonalisation, and these digits may update to a fixed

digit after more processing. The 11th digit work around is not a weakness

of using programs to generate numbers, it is a constraint of the diagonalisation technique.

Now we run the 10 Universal Turing Machines.

UTM1
1 0.0494873739
2 0.000004343939
3 ?
4 0.9999999999
5 0.500000000000

UTM2
1 0.5000000000
2 0.0000000000
3 0.9999999999
4 0.1010101010

...

UTM10
1 0.0000000000
2 0.1111111111

3 0.1111111111
4 0.9999999999

Instead of 1 anti-diag number, we have 10.

The diagonals of each UTM :

UTM1 diagonal 0.0059
UTM2 diagonal 0.5000
UTM3 diagonal 0.3746
UTM4 diagonal 0.2234
UTM5 diagonal 0.1223
UTM6 diagonal 0.6563
UTM7 diagonal 0.3483
UTM8 diagonal 0.5403
UTM9 diagonal 0.2355
UTM10 diagonal 0.0119

Assume each UTM is independent of one another.

The 1st digit of the diagonal has several different numbers from each of the 10 UTMs.

1st digit of the diagonals {0,5,3,2,1,6,3,5,2,0}
2nd digit of the diagonals {0,0,7,2,2,5,4,4,3,1}
3rd digit of the diagonals {5,0,4,3,2,6,8,9,5,1}
4th digit of the diagonals {9,0,6,4,3,3,3,3,5,9}

Each UTM list of reals has an anti-diag number, but does this number appear on another UTM list?

For an anti-diag number to be completely unique, its 1st digit cannot be any of {0,5,3,2,1,6,3,5,2,0}, that leaves 4, 7, 8, 9.

The 2nd digit cannot be in {0,0,7,2,2,5,4,4,3,1}, that leaves 6, 8, 9.

The 3rd digit of anti-diag cannot be in {5,0,4,3,2,6,8,9,5,1}.

All digits are taken here, so there is no possible anti-diag number.

Frequently along the diagonal, all 10 digits will occur across the 10 UTM lists. Any supposed number not on the lists could occur at these positions.

So diagonalisation only finds new numbers from a single list, or from a small set of lists less than the number of digits.

There are 2 possibilities,

1 each list stands as a complete set of real numbers,

2 the 10 lists combined(#2) are 'complete'.

If the 10 lists have a union(#2) to count all numbers then we can construct a single list enumerating all the numbers.

UTM1 real1

UTM2 real1

UTM3 real1
..
UTM10 real1
UTM1 real2
UTM2 real2
..

This list is amenable to diagonalisation and a new real could be constructed.

By the Church Turing Thesis each list is exactly the same set of reals therefore 1/ each list stands as a complete set of real numbers.

ADDENDUM

From: HERC777 (herc777@hotmail.com)
Subject: Cantor's complete list assumption is not extrapolated

I'll borrow the word permutation to mean unique sequence.

- 1 Assume a real number list
- 2 Form a new permutation different to every number on the list
- 3 The list is incomplete

- 1 Assume a COMPLETE real number list containing every permutation
- 2 Form a new permutation different to every number on the list
- 3 CONTRADICTION – the list is already complete

The assumption 1, is that every infinitely long permutation is listed. The anti–diag number is literally constructed as "form a new permutation different to every number on the list".

With a finite list anti–diag always forms a new number and is a valid technique, its easy to imagine an infinite anti–diag but is it valid? It forms a contradiction so either 1 or 2 is wrong.

Given every infinite permutation is present on the list, the *construction* of a new permutation is itself flawed.

A binary example.

Small samples of radioactive material have their particle emission rate measured. A sample frame rate is established and the output of a digitised poisson distribution is recorded, 0 for no emission, 1 for a particle emmitted – a ping on the gieger counter.

sample 1 0101010010010100101010110111010101010..
sample 2 010101011010101010101010101010101010..
sample 3 11101101011101010111010110101010101010..

sci.math: Re: Different size infinities?

Each sample runs forever, there is unlimited sampling and resources for the experiment. After about 10 samples, all possible permutations for the 1st 3 frames are recorded.

- 000
- 001
- 010
- 011
- 100
- 101
- 110
- 111

8 possibilities, with some doubling up of results.

After several million samples, the variations in the first 15 frames are all covered.

- 000000000000000101010101010...
- 0000000000000001101010101010...
- 00000000000000010101010101010...
- 00000000000000011101010101010...
- ..
- 11111111111111110101010101010...

This process is logarithmic, it takes much longer for the covered permutations to grow as the list grows. What is the behaviour as the list approaches infinity?

$$\log(\infty) = \infty$$

Do all permutations eventually occur? This is the expected physical and probabilistic result. Using infinite diagonalisation this theoretically does not happen, even with infinite resources.

There are infinite samples, all *known* permutations are covered yet anti-diag results in a new permutation??

According to accepted mathematics today, the $\log(\infty)$ length of the covered permutations reaches an end, and the tail end of the numbers are again random and unique.

This is what hyperinfinity is based on, the privileged concept all mathematicians understand beyond the man in the streets' reasoning.

ASSUME the list of permutations is complete, IT IS!
Construct the anti-diagonal, anti-diagonal is not unique after 3 digits, its not unique after 15 digits, its not unique after infinite digits, all permutations are present.

sci.math: Re: Different size infinities?

You have an INFINITE sample of radioactive readouts. Are you 100% absolutely certain you can find a new sequence of 1s and 0s?

Herc