Re: Another stab at Cantor

Source: http://sci.tech−archive.net/Archive/sci.math/2006−09/msg03905.html

- *From*: magidin@xxxxxxxxxxxxxxxxx (Arturo Magidin)
- *Date*: Tue, 19 Sep 2006 20:05:42 +0000 (UTC)

In article <1158694291.815365.301710@xxxxxxxxxxxxxxxxxxxxxxxxxxxx>,
georgie <geo_cant@xxxxxxxxx> wrote:

[...]

> > Namely, the set that contains exactly all strings described in (i) (the original list), plus all strings described in (ii) (the constructed strings Dk for integer k). And nothing else.

> Since the limiting case is not a list, (ii) must not apply so you didn't explain why the limiting case can't include all the reals.

Do you know what a red herring is?

A prime example is your continued invokation of the description in (ii) as if it is meant to apply, somehow, to "the limiting case." As I've gone over several times now: the points (i) and (ii) were DESCRIPTIONS OF THE ITEMS IN THE LIMITING SET. Those items naturally fall into two classes, which were described in those two points. The reason why you insist on somehow implying that (ii) is meant to be a description that applies to "the limiting case" (as opposed to what it explicitly is: a description of how a certain class of elements in the final set are constructed) is beyond me.

The procedure presented consisted of an a priori list, and a recursive description of how to obtain further strings. The procedure then said something along the lines of 'apply the procedure as long as you can', and the "limiting case" will consist of every element that was in either the a priori list (those elements I described under (i)), plus the strings obtained by applying the procedure (those elements I described under (ii), which is ONLY A DESCRIPTION OF HOW TO OBTAIN EACH OF THOSE ELEMENTS); and nothing else.

Stop your red herrings.

Now: the limiting case was not presented as a list. However, the limiting case is a set which, as it happens, CAN BE LISTED. Not in the way presented originally (which produces an integer−indexed set, not a list). But rather in an alternate way which includes each and every element of that set.

Again: the "limiting case" set contains two classes of elements. The first class consists exactly of those strings which were on the original list. The second class consists exactly of the strings Dk constructed from that original list, recursively, via the process described in (ii). Nothing else. Because the construction of the Dk is recursive, and because the construction given only makes sense at finite step, the only elements in the second class are the strings D1, D2, D3, ..., Dk, ...; i.e., Dn for each positive integer n, and nothing else.

The list described in (ii) is an AUXILIARY list, used in order to construct the corresponding string Dk.

The original poster constructed "partial lists" via a process of pre−appending. The most natural way of thinking about this is to start with a list that begins with item number 1. Pre−append an object, indexed with 0. Pre−append an object, by indexing it with −1. Pre−append an object, by indexing with −2. Etc. Etc. Etc.

He then asserted that after doing this an infinitely countable number of times, he would obtain a list. −>That<− assertion is false. What he "ends up" with in the limiting case is a "pseudo−list" (I'm making up the term right now, only for the purposes of this paragraph) which is not a list because it has no first element; it contains the original list as the items indexed by 1, 2, 3, 4, ..., and contains the added strings as the items indexed by 0, −1, −2, −3, −4, ... It can be represented by:

.
.
.
−3 String D4
−2 String D3
−1 String D2
0 String D1
1 First item of original list L1(1)
2 Second item of original list L1(2)
3 Third item of original list L1(3).
.
.
.

The "list L2" of the original poster is the list you obtain if you chop off everything above the 0 in the representation above, keep the rest, and reindex so it starts with 1 instead of 0. The "list L3" of the original poster is the list you obtain if you chop off everything above the −1 in the representation above, keep the rest, and reindex everything so it starts with 1 instead of −1. Etc. Etc. Etc.

That set, −>AS GIVEN<− is not a list. This does not mean that set that consists of all those items "cannot be listed". Just that it is not given as a list. Just like a bunch of cards thrown around the room is not a deck, but that does not mean they cannot be assembled into a deck. They can.

So this set can be reordered into a list. That's what was done.

> I'm not saying it does. I'm pointing out nobody has proven it doesn't.

No. You are simply repeating that "nobody has proven it doesn't", despite the fact that it −>has<− been proven it doesn't. You just haven't understood what is going on.

> You can't seem to accept that and have resorted to personal attacks.

Oh, I can accept perfectly well that you have failed to understand the rebuttals. I have chosen to express my contempt for you in the form of personal attacks. This has nothing to do with denial, but merely with contempt.

> > > L_Omega, as described, is a set indexed by the integers (not a list). To positive integers correspond the elements of the original list L1. To nonpositive integer indices correspond the strings Dk. What are the strings Dk? They are the strings constructed in the manner decribed in (ii). Point (ii) describes strings that are in the final set; it is not, except in your febrile imagination, an attempt at describing the full final set.

> > I think you are in denial by calling something that isn't a list a

> > > "set indexed by the integers".
>
> > And I −>know<− you don't know what you are talking about.
>
> You've yet to impress me with any of your knowledge.

You've yet to demonstrate the ability to learn anything, so that's hardly a surprise.

> > Tell me, son. What is it you think a "set indexed by the integers" means?
> >
> > If you don't know, then ask. That way you won't make a fool of yourself.
>
> You brought that terminalogy up.

So, if you didn't know, THEN ASK.

Duh.

> You said the limiting case,
> L_Omega, is not a list and then you went on to call it a set
> that was indexed by the integers.

Which is not the same thing as a list.

> > A set X is said to be "indexed by the set Y" if there is bijective set−theoretic map f:Y−>X. So saying that the set is "indexed by the integers" means there is a bijection from the set of all integers (positive, negative, and zero), to the set in question.
> >
> > In this case, our set contains exactly the elements L1(k), and the elements Dn, with k and n positive integers, and nothing else. Call this set F (for "final").
>
> "our set" is L_Omega? The one you said wasn't a list?

The one that was not given as a list. It was given as an integer−indexed set.

Do keep up, Georgie boy.

>> We define the map f: Z −−> F by:
>>
>> if k is positive, f(k) = L1(k).
>>
>> f(0) = D1
>>
>> if k is negative, f(k) = D(−k+1).
>>
>> So: the "index 0" element is D1. The "index −1" element is D(−(−1)+1)=D2. The "index −2" is D3. The index 100 is L1(100). The index −85917 is D(85918).
>>
>> So the set is indexed by the integers.
>>
>> No, the set as given is not a list, because this set, −>ordered this way<− (which is the way it is naturally ordered given the procedure given) is not a list.
>>
>> However, there is an easy way to make a list which has each and every element of this set on the list exactly once; which is by listing them in the same way we can list the integers: 0, 1, −1, 2, −2, 3, −3, etc.
>>
>> A −>list<− is a set X together with a bijection from either an initial segment of the POSITIVE INTEGERS to X (or the naturals if you like your lists to begin with 0), or from the entire set of positive integers to X.
>>
>> Every list corresponds to an indexed set, indexed by either N or an initial segment of N. Not every indexed set is a list.

> Form the OP:
>
> "Let L_Omega be the list defined by the totality of all possible steps of this procedure."

And it was pointed out that after applying "all possible steps of this process", the result is NOT a list. The result is a sequence that goes off infinitely in both directions. And it was pointed out that this was merely a minor technical problem to his argument, easily fixed by providing a DIFFERENT order to the collection of all resulting elements.

What is it that you keep having so much trouble understanding, Georgie?

The original poster describes a process whereby, if you have a list, you produce a new string, pre−append it, and obtain a new list. Lather, Rinse, Repeat.

The original poster then talks about continuing this process via the "totality of all possible steps of this procedure." The procedure can only be applied so long as you have a list. This can certainly be done at each natural−numbered step. If you take the union of these steps, what you end up with is an ORDERED set which, AS GIVEN, is not a list; among other reasons because each term has an infinite number of predecessors. At this stage, the "procedure" cannot be applied, so the process STOPS.

So we are done with −>that<− process.

At this stage, it was asserted the resulting (ordered) set contains all possible strings. That is false. While the set, ordered as given, is not a list, it can nonetheless be REORDERED so that the result −>IS<− a list, and we can easily produce strings not on THAT list, and therefore not in the original (ordered) set.

I think that what is sticking in your craw is that you somehow decided that the original poster meant to CONTINUE applying the process anytime he could take the collection of everything he had produced thus far and somehow turn it into a list.

But that is NOT what he said. What he gave was a specific deterministic algorithm that does not allow you to reorder the sets you obtain. You can only take a list, produce the diagonal string, and preappend the diagonal string to obtain a new list. You don't get to reorder what you get. That process has a limiting case, which what I described, and that limiting case is not a list if ordered as the procedure dictates it MUST be ordered.

> "If Arturo Magidin's step is a step then it is one of all possible steps." is true.

No, it is false. You are putting far more into the procedure than the original poster provided. And then you are claiming that the proof presented is insufficient because it fails to address something which was not put into the original procedure. Well, DUH.

> Just ask Virgil.

If Virgil misinterpreted what the other poster wrote, that's Virgil's problem, not mine.

What I described is not a "possible step" in the original poster's algorithm; it takes place AFTER the original poster is done with his algorithm, which can only be applied omega times. What I described "takes place" at step "omega + 1", which is incompatible with the original poster's description.

So by definition, there is no

> diagonalization steps remaining after they've all been performed.
> There
> can be no diagonalizing outside the set of all diagonalizations.

L_Omega is NOT the set obtained by applying "all possible diagonalizations", unless you change the procedure described.

> L_Omega can't be a list. It can't be diagonalized. It can't be achieved by performing all the diagonalizations either, but that doesn't mean it doesn't exist.

Who said it doesn't exist? Still savaging strawmen, while beating your wife?

−−
==
"It's not denial. I'm just very selective about
what I accept as reality."
−−− Calvin ("Calvin and Hobbes" by Bill Watterson)
==

Arturo Magidin
magidin−at−member−ams−org

.