

## Re: Cantor For Dummies ...

---

*Source:* <http://sci.tech-archive.net/Archive/sci.math/2007-03/msg05803.html>

---

- *From:* riderofgiraffes <[mathforum.org\\_am@xxxxxxxxxxxxxxxx](mailto:mathforum.org_am@xxxxxxxxxxxxxxxx)>
  - *Date:* Thu, 29 Mar 2007 09:41:57 EDT
- 

No, you haven't defined precisely what your list contains.

I don't have a list. But the collection of all algorithms that output a committee can be listed.

Yes. The collection of all algorithms that output a committee can be listed.

[My algorithm] does generate a committee. I've said so repeatedly.

If you have a real algorithm and it outputs a committee, then it's part of the collection.

Agreed.

However, it takes input to do so. You seem either not to have noticed that, or to be ignoring it.

That's not relevant.

I think it is. More about this below.

Since it's an important point I've been trying to make that clear.

I realize that. You don't realize that the

Re: Cantor For Dummies ...

collection of all algorithms that output a committee is general enough to include them all.

I do realise that. I also realise that the specific committee output by my algorithm depends on its input. More about that below.

I claim that my process, when given a list of committees, produces a committee. Do you agree?

If it is an actual algorithm and uses valid input.

Good. Agreed.

OK. So you want all algorithms whose output is a committee, whether they require input or not.

Sounds good to me.

Good. Agreed.

... you want to talk about the output of all these algorithms. Some of them, mine included, don't produce output without input. If you don't tell me what input to use, we can't talk about the output.

I have no input to any algorithms. I'm considering only their outputs.

Given that some of the algorithms, mine included, require input before they produce output, how can you talk about their output without talking about their input? That sounds like nonsense to me, but I'm sure it will become clearer below.

You can assume you have any valid input. Use any valid input you desire. Note that your algorithm's output is not valid to use as its input.

Re: Cantor For Dummies ...

So I'll take a single, fixed list of committees,  
and I'll use that as the input for every algorithm  
that requires a list of committees as input.

For some of them you can't list their output  
without saying what their input is. So you  
are wrong. We can't list their output.

YOU are wrong. I never said they would operate  
without input.

No. You repeatedly ignored the question of what  
input to give them. However, you've now said that  
I can use any input I like, so I've chosen the input  
I want. It's a single, fixed list of committees.

Assume whatever input you need. I have none. I  
never claimed to have any input for your algorithm.  
I only claim that if your algorithm uses ANY valid  
input and generates a committee, then it's in the  
list of algorithms that generate committees.

OK.

I suppose if we listed the outputs of all those  
algorithms, ...

So on the one hand you say that you don't have  
any input for the algorithms, and then on the  
other hand you want to use their output. That  
is inconsistent. However, we're moving on,  
because you have let me use any input I like.

... we would have a list of committees.  
But that list wouldn't be valid input for your  
algorithm since part of that list would be your  
algorithm's output.

That's fine, because my algorithm never uses that  
output list. I never said my algorithm was supposed  
to use the list of committees that the list of

algorithms produces. I never talked at all about using the output as input.

Suppose you take your list of algorithms, some of which require a list of committees as their input, others produce a committee without requiring input.

Now take any list of committees and feed it as input to all those committees requiring it – mine included.

Sorry, that's not possible. We can't use the output as input. So we can only use any list other than lists made from the output.

I never said that the list being fed in was the output list. Never once did I say that an output list was being used as input. I'd be interested to see where you claim I said that an output list was being used as input.

I suspect you'll ignore that challenge.

Consider the list of committees that you get as output.

There is a committee on that list that wasn't on the original input list, namely, the committee my algorithm produced.

I suppose that might be. Assuming valid input.

Good.

OK. So you have agreed that whenever my algorithm is fed with a list of committees, my algorithm will produce a committee not on the input list.

I've said nothing about feeding output in as input, I've said nothing about where that input list comes from, I've just said that when my algorithm is fed with a list of committees, my algorithm produces a committee that's not on that input list.

And you've agreed.

That's all I wanted to show. So I don't understand why you're claiming something different.

I'm not claiming anything different.

Hmm.

Therefore no such algorithm exists.

You haven't shown that.

I'm claiming no algorithm can take its output as its input.

And why was that relevant? I certainly never claimed that my algorithm was working on its own output.

.