Re: Question: Given |X|>0 and |Y|>0, can X x Y be empty?

Source: http://sci.tech−archive.net/Archive/sci.logic/2007−08/msg00086.html

---

- *From*: magidin@xxxxxxxxxxxxxxxxx (Arturo Magidin)
- *Date*: Sat, 4 Aug 2007 00:19:01 +0000 (UTC)

In article <1186184637.040165.72610@xxxxxxxxxxxxxxxxxxxxxxxxxxxx>,
Scott <ToaTerra@xxxxxxxxx> wrote:

> On Aug 3, 4:00 pm, magi...@xxxxxxxxxxxxxxxxx (Arturo Magidin) wrote:
>
> > In article <1186161736.517279.72...@xxxxxxxxxxxxxxxxxxxxxxxxxxxx>,
> >
> > > Let f be a function f:S −> P(S).
>
> ...
>
> > So, you are saying in Prop 4: if p is an element of P(S), then there is a function (with domain where?) whose image includes p.
> >
> > Better:
> >
> > If p in P(S), then there exists f(p) : S−>P(S) such that p is in the range of f(p).
>
> I was trying to define the domain and codomain of any f by the single sentence. I can see that was a mistake. The better sentence should be:
>
> If p in P(S), then there exists f(s) : S −> P(S) such that p is in the range of f(s).

What does "s" refer to in your sentence?

Better is to use f(p), like I did. Or f_p. Or something that makes the connection between f and p.

> > > Proof: If p in P(S), then (As in S)((s,p) in M) [Prop 3]. Let f(s) =
> > > p <−> (As in S)((s,p) in M).

This is lousy. Are you trying to define f as "constant p"? Then DO SO. What you write here is pretty near unintelligible, close to

As a constant function − yes. I thought that is what I was defining: for any p in P(S), there exists a constant function f(s) = p. I was showing this function exists because its a subset of M.

But you did not define it correctly; that's the problem, not whether "it exists". You defined it incoherently.

nonsense. Since (s,q) in M for all s in S and all q in S(p), your condition on the right ALWAYS holds, so it does not define a function. Your "function" is in fact "assign to each element of S every element of P(S)" and that is not a function.

No, the other way around:

The other way around is what you tried to do. Alas, it was not what you −>actually<− did. What you wrote made f = M.

for each p define a function f_p that maps
every element of S to one element of P(S). To use your notation below:

(Ap)(let f_p(s) = { (s,p) in M })

Perhaps you can educate me on the correct way to say this?

I did. I wrote it in my reply.

f is functional

No, it is note. Your definition makes f = M, and that is in general not a function.

Example: Given f: {0, 1, 2} −> {cat,dog} where f = { (0,cat), (1,cat), (2,cat) } why is this not functional and total?

Sigh. Because THIS IS NOT WHAT YOU HAD DEFINED BEFORE! My comment was about what you defined before, not this; this is the CORRECT, specific instance of what you −>meant<−. It is not, however, an instance of what you WROTE. I cannot know what you meant, I can only go by what you WRITE.

See? This is −>exactly<− the problem. Now you are wondering why I am saying that what you were THINKING of is not a function. I was not. I was saying that what YOU SAID did not give a function. If what you said was not what you were thinking of, then the problem is that you are unable to say what you mean; but you insist on trying to apply comments on what you SAY to what you THOUGHT you were saying. Since they seem to seldom be the same thing, isn't it a time for you to stop that annoying practice?

What you wrote was that you would define f(x)=y if and only if for all x in the domain, (x,y) was in D x C, where D is the domain and C the codomain.

So, with your sets above, you wrote that you would define f(x)=y if and only if for all x in {1,2,3}, (x,y) is in {1,2,3} x {cat, dog}.

Under this "definition", f(1)=cat, since (1,cat), (2,cat), and (3,cat) are elements of {1,2,3} x {cat, dog}. But also, f(1)=dog, because (1,dog), (2,dog), (3,dog) are elements of {1,2,3} x {cat,dog}. In fact, f(x) = cat and f(x)=dog for x=1, for x=2, and for x=3, under your original attempt at a definition.

This is of course NOT what you write above. Because now you CHANGED what you wrote. But you insist on complaining about my comments as if they applied to this new assertion. They were made about your ORIGINAL, INCORRECT, INCOHERENT definition.

This is one thing I find so infuriating and annoying about your writings.

Let A = { x in S : x notin f(x) }.

What f? The one you define in Prop 4 for a given p, or the one we

At this point, any f.

Which is a problem, since you use "f" for two different things. Which is why what you wrote was incoherent.

It does no good for you to now change what you wrote; you still messed up then, for the reasons I explained.

Since I am trying to make this general, I am not
restricting which f.

But you also had very specific meanings for "f", which were not general. Remember? They were in Prop. 4. That is why you were so confused. You were using "f" to mean two different things.

So, let f be an arbitrary function, f:S−>P(S). It need not be equal to f(p) for any p in P(S), i.e., it may not be the conditional one we have from Proposition 4.

Correct

If only you had done so to begin with, instead of the mess you did instead.

it can be any f, not necessisarily the one from Prop 4.

Given f, we define A_f to be

A_f = { x in S : x notin f(x) }.

Okay, I can see this is better.

This is simply Cantor's Argument.

From what I read, Cantor's Argument deals with surjective

functions.

No. Cantor's Theorem shows that given any f:X−>P(X), there exists Y_f in P(X) such that (Ax) (x in X −> f(x) =/= Y_f). It is not about surjective functions.

I
am claiming there are *no* functions with A_f in the range including simple ones constructed with finite sets.

Sigh. No, you are not saying that there are no functions with A_f in the range.

What you are saying is that for EACH function f, the CORRESPONDING set A_f is not in the range of f.

You are not making a universal statement about all functions realtive to a single set. You are making a universal statement about all functions, with each statement refering to a possibly different set.

The set A_f depends on f. Change f, it may change A_f. So it is FALSE to say that you are claiming there are no functions that have A_f in the range. You are claiming ONLY that f, the function you start with, does not have A_f in the range.

A_f is within the bounds of the quantification of f. It is not a free variable. It is allowed to change based on f.

Look; the following two statements say two different things:

(Ax)(Ey) p(x,y)
(Ey)(Ax) p(x,y).

The first statement says that for every x, there exists a y (which may depend on x; that is, if you change x, the corresponding y may change) such that p(x,y) holds.

The second statement says that there is a single y such that for every x, the statement p(x,y) holds. Even if you change x, the y need not change.

For example, the statement

(Ax)(Ey) (x = y)

is a statement that is true in any theory. However, the statement

(Ey)(Ax) (x = y)

is only true in models which have only one element.

It is like the difference between "Everyone has a mother", and "There is someone who is everyone's mother."

From Prop 4 and the
contrapositive, ~(Ef)(A_f in range(f)) −> A_f notin P(S).

> > Wrong. Proposition 4 was incorrect and does not refer to the same f as proposition 5. Here is your mistake, born out of that LOUSY notation where you use f to mean many different things at the same time.

> Can you explain this a bit more.

No more than I do above.

> Prop 5 applies to any function,
> including the function in Prop 4.

Prop 5 says: given a function f:S−>P(S), we can construct a set A_f, THAT DEPENDS ON f, with the property that f(x)=/=A_f for any x in S.

Prop 4 says: if S is not empty, then given an element p in P(S), we can construct a function g(p):S−>P(S) (which I will call g(p) to emphasize the separation between Prop 4 and 5), such that p is in the range of g; that is, there exists s in S such that g(s)=q.

So, take a function f. Apply Prop. 5 to obtain the set A_f in P(S). Now apply Prop. 4 to the set A_f to obtain a NEW function (not necessarily equal to the f you started with), g(A_f):S−>P(S), such that there exists s in S for which g(A_f)(s) = A_f.

Can we now apply Prop 5 to g(A_f)? Yes, we can. But g(A_f) is not the same as f, and so there is no reason to assume or suspect that A_(g(A_f)) will be the same as A_f. In fact, it won't. Our function g(A_f) is defined in Prop 4 as g(A_f)(s) = A_f for all s in S. Therefore, the set A_g(A_f) is:

A_g(A_f) = { s in S : s notin A_g(A_f)(s) }
= { s in S : s notin A_f } by definition of A_g(A_f);
= S \ A_f.

That is, not only is the set you get from g(A_f) not the same as the one you got from f, it is the COMPLEMENT of that set.

> Given ~(Ef)(A_f in range(f)) isn't
> it true that ~(A_{f(s)} in range(f(s))?

Not the way you are interpreting it, which is as if A_{f(A_f)} were equal to A_f.

> Proposition 4 says:
>
> (Ap) (p in S(p) −>
> (Ef(p))(f(p):S−>P(S), f(p) is a function,
> and (Ex)(x in S and f(p)(x)=p)))

Err,no.

It's the meaning of what you wrote: for every p, if p is an element of S(p), then there exists a function f(p) from S to P(S) such that p is in the range of f(p). That is exactly what that says.

Now, the PROOF in fact constructs f(p) as a constant function, but that was not what Prop 4 stated.

It says:

> (Ap)(p in S(p) −>
> (Ef(s))(f(s):S −> P(S), f(s) is a function, and (Ax)(x in S and f(s)=p).

No, that is not what Prop 4 stated. That is the function that was exhibited in the proof that satisfied the statement.

> Question: Would it have been a whole lot clearer if I constructed the proofs using just the one function from Prop 4?

What would have been better is if you had kept the difference between functions you get from subsets per Prop 4 separate from subsets you get from functions per Prop 5.

Look: Prop 4 is a "machine" that, given a subset p, gives you a function g(p) with some properties.

Prop 5 is a "machine" that, given a function f:S−>P(S), gives you a set A_f with some properties.

To get Prop 5 started you need a function, and the set you get will depend on the function.

To get Prop 4 started you need a set, and the function you get will depend on the subset.

So, you have a way to go from P(S) to the set {f : f is a function from S to P(S)}.

And you have a wa to go from the set
{g : g is a function from S to P(S)} to P(S).

But you seem to be assuming that if you go from P(S) to the set of functions and then back, you will end up where you started with. You do NOT. In fact, you end up in the COMPLEMENT of what you started with. And likewise, if you start with a function, then obtain the corresponding set, and then you look for the function corresponding to that set, you will NOT obtain the function you started with.

By using a single letter to represent both the function you start with, and the function you obtain after applying Prop 4 to the set you get from Prop 5, you blur that line and create the confusion that is hobbling you.

--
==
"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

.