Re: Implementable Set Theory and Consistency of ZFC

Source: http://sci.tech−archive.net/Archive/sci.math/2007−10/msg04365.html

- *From*: Han de Bruijn <Han.deBruijn@xxxxxxxxxxxxxx>

- *Date*: Wed, 24 Oct 2007 15:42:17 +0200

Jesse F. Hughes wrote:

> Han de Bruijn <Han.deBruijn@xxxxxxxxxxxxxx> writes:
>
> > Jesse F. Hughes wrote:
> >
> > > Han de Bruijn <Han.deBruijn@xxxxxxxxxxxxxx> writes:
> > >
> > > > Of course, as a philosopher, you have no
> > > > idea what Database Applications
> > > > are all about. And what set theory has to do
> > > > with them.
> > >
> > > Educate me. How does your toy set theory contribute to
> > > database
> > > applications?
> >
> > My "toy set theory" includes all set theory that is needed to build any
> > database applications. But it's nothing new in that respect.
>
> In what sense can someone use set theory to build a database
> application? And if it's nothing new in this respect, why did you
> repeatedly claim, "Look, my Implementable Set Theory e.g. is covering
> all Database Applications on Earth, which is .. a billion dollar
> business!"

It has a simple explanation: emphasis not on "my" but on "Implementable Set Theory", business more important than HdB (: doesn't care so much).

> Hey, like you say, I'm just a lowly philosopher, so I don't understand important things like database applications. You'll have to help me out here. Suppose I have a database consisting of employee records (name, gender, income, let's say) and I have an application that sorts the database on name and prints a payroll check for each employee. How do I build that application using your set theory? And why is building that application in set theory something I might want to do?

Posted this in response to Virgil as well:

I still remember quite vividly the birth of Relational Database Systems, with the advent of Oracle (somewhere in the seventies). They reached me a folder and in that folder it was mentioned explicitly that relational database systems were firmly supported by a solid piece of mathematics, called, guess what: Set Theory ! Sounded very impressive to me, at that time. (Meanwhile, Oracle databases have become my bread and butter.)

And let e.g. Google be your friend.

Han de Bruijn

.