# Re: Godel proved maths inconsistent not incompleteness theorem

***Source:*** http://sci.tech−archive.net/Archive/sci.logic/2008−05/msg00459.html

- *From*: Charlie−Boo <shymathguy@xxxxxxxxx>
- *Date*: Tue, 6 May 2008 22:06:52 −0700 (PDT)

On May 6, 10:29 pm, herbzet <herb...@xxxxxxxxx> wrote:

> Charlie−Boo wrote:
>
> > herbzet wrote:
> >
> > > Charlie−Boo wrote:
> > >
> > > > On May 6, 5:59 am, David C. Ullrich <dullr...@xxxxxxxxxxx> wrote:
> > > >
> > > > > On Mon, 5 May 2008 21:25:23 +0000 (UTC), Chris Menzel <cmen...@xxxxxxxxxxxxxxxxxxxx> wrote:
> > > > >
> > > > > > On Mon, 5 May 2008 12:25:46 −0700 (PDT), Charlie−Boo <shymath...@xxxxxxxxx> said:
> > > > > >
> > > > > > > On Apr 30, 12:59 pm, Chris Menzel <cmen...@xxxxxxxxxxxxxxxxxxxx> wrote:
> > > > > > >
> > > > > > > > On Wed,

30 Apr 2008 01:54:34 −0700 (PDT), Charlie−Boo

Can you code from specs that contain "e.g." and "etc."?

I'm not giving you a spec.

That's the problem. You don't even have a spec and you say it's trivial.

That's because a

decent programmer can often tell when a task is trivial.  It is no surprise, of course, that you are unable to see the triviality in question, given that you are still terribly confused anout the distinction between theorem proving and proof checking.

That of couse depends on how wffs are represented.

No kidding. Presumably, they are "represented" they way WFFs are typically "represented" in first−order languages, as strings in a certain recursively defined set. Standard stuff.

Then why do the various expositions use different syntaxes?

Uh, what? Different syntaxes still use strings of symbols. (Also, but for minor details (e.g., the presence of "0" as a primitive symbol), the different expositions use exactly the same syntax. It's the *axioms* they use that can differ.)

Zohar Manna and Richard Waldinger gave up trying to implement Inductive proofs years ago.

*Inductive* proofs? You think that is even *relevant* in a discussion of *deductive* proof checkers? Just how confused *are* you about the issues here?

ZF without

Induction??

I understood you to mean "induction" in the sense of "inductive logic", which of course is extremely difficult to implement. Not that this matters in the least to the point. Even if we are talking about induction over sets and numbers in the deductive sense, you are still confusing proof checkers with theorem proving. The difficulties of implementing proof mechanisms for induction in either sense is simply irrelevant to the issue of proof checking.

And just FYI: ZF doesn't "have" induction, at least, not as an axiom. It is derivable.

As mentioned already, I gave you a link to a fully implemented proof checker for a complete system of natural deduction, one that, in particular, implements 20 of those rules of inference that you call "the hard part". All one would have to do to convert it into a dedicated proof checker for ZF, PA or what have you would be to implement some elementary pattern matching routines for the axioms/schemas of your chosen theory.

Oh, I see. It's trivial if

you already have a program that does 90% of it.

Yes indeed, all of logical infrastructure including the rules of inference are coded. That you are trying to make an issue of the fact that the simple pattern matching routines for axiom checking have not actually been coded shows that you are either blatantly dishonest or a completely incompetent programmer.

1. For each

line in the purported proof

a. Run one step of the theorem prover
2. For each theorem generated, compare it to the theorem at that line.
3. If equal, then go to the next line (1)

a. If no more lines, the proof is valid.
4. If no more

theorems to generate, the proof is not valid.

So you think hard things are easy and easy things are hard.

There is a pretty serious problem with your algorithm, specifically the "If no more theorems to generate" condition in line 4 –– you *do* know there are infinitely many theorems of first−order logic, right?

There are a finite number of theorems at that single step.

Er, you really think that helps? Seriously? You do?

Whether it helps or not, it's simply not true. At least not necessarily − there are plenty of formal systems for which it's false.

My goodness, nobody said that formal systems must be finite. The question regards axiomatic systems, specifically designed to be of finite construction. There is a finite number of axioms and rules of inference. Otherwise it's not an axiomatic system. In fact, you are bastardizing the notion of an axiomatic system to the point that it couldn t even be programmed in general anymore, so there is no proof generator to compare to a proof checker in the first place.

From dictionary.com:

No. http://en.wikipedia.org/wiki/Finitary

No no.

Yes, obviously.

"A finitary argument is one which can be translated into a finite set of symbolic propositions starting from a finite set of axioms. In other words, it is a proof that can be written on a large enough sheet of paper (including all assumptions).

The emphasis on finitary methods has historical roots. In the early 20th century, logicians aimed to solve the problem of foundations; that is, answer the question: "What is the true base of mathematics?" The program was to be able to rewrite all mathematics starting using an entirely syntactical language without semantics.

The stress on finiteness came from the idea that human mathematical
thought is based on a finite number of principles and all the
reasonings follow essentially one rule: the modus ponens. The project
was to fix a finite number of symbols (essentially the numerals
1,2,3,... the letters of alphabet and some special symbols like "+", "−

", "(", ")", etc.), give a finite number of propositions expressed in

those symbols, which were to be taken as "foundations" (the axioms),
and some rules of inference which would model the way humans make
conclusions. From these, regardless of the semantic interpretation of
the symbols the remaining theorems should follow formally using only
the stated rules (which make mathematics look like a game with symbols
more than a science) without the need to rely on ingenuity. The hope
was to prove that from these axioms and rules all the theorems of
mathematics could be deduced.

The aim itself was proved impossible by Kurt Gödel in 1931, with his
Incompleteness Theorem, but the general mathematical trend is to use a
finitary approach, arguing that this avoids considering mathematical
objects that cannot be fully defined."

That's all very nice, but irrelevant. Your "proof checker" alogorism above does not work, and for exactly the reason that David Ullrich has stated. In particular, it will not work for proofs in PA and ZFC.

> > There are an infinite number of theorems generated in one step? How?
> > Then it is not an axiomatic system and if it were you couldn't program
> > them in general anyway as I said.

> HA HA HAAAAAAAA.
>
> You don't know that the set of theorems of PA or of ZFC are
> r.e.? That both have an infinite number of axioms? That
> both have an infinite number of terms that can be substituted
> for variables?

Obvioulу the set of theorems can be r.e. The question is whether the set of theorems in always r.e., not whether it is ever r.e. And already you are then going outside of the axiomatic method as it does not specify that the set of axioms is r.e. (explicitly) much less how that would occur. It says the set is finite, so we are all set.

It is easy to program a finite set of axioms, which is along the lines of the purpose of a finitary system. But if you want an infinite number of axioms, not only are you violating the rules of an axiomatic system, you also are not able to program it in general and you have something not defined in an axiomatic system: a scheme for an infinite set − as opposed to finite sets of axioms and rules that is specified by definition of an axiomatic system.

I have been saying this all along, but still people walk the plank blindfolded.

See http://www.reference.com/search?r=13&q=Finitary :

"In the early 20th century, logicians aimed to answer the question: "What is the true base of mathematics?" The program was to rewrite all mathematics.

The stress on finiteness came from the idea that human mathematical thought is based on a finite number of principles. The project was to give a finite number of propositions which were to be taken as "foundations" (the axioms), and some rules of inference. The hope was to prove that from these axioms and rules all the theorems of mathematics could be deduced.

The aim itself was proved impossible by Kurt Gödel, but the general mathematical trend is to use a finitary approach, arguing that this avoids considering mathematical objects that cannot be fully defined."

Note the reference to a finite set of axioms, as well as the last sentence: This avoids considering mathematical objects that cannot be fully defined. This is exactly what I am saying you are doing when you allow infinite sets of axioms − you cannot program that in

general, so there is no proof generator. That's why it is not allowed
by an axiomatic system. The idea is to be able to generate the
theorems.

C−B

> Your algorism doesn't work, not even in principle!
>
> This is rich!
>
> −−
> hz− Hide quoted text −
>
> − Show quoted text −− Hide quoted text −
>
> − Show quoted text −

.