Re: Provability

Source: http://sci.tech−archive.net/Archive/sci.math/2007−10/msg04530.html

- *From*: galathaea <galathaea@xxxxxxxxx>
- *Date*: Wed, 24 Oct 2007 23:19:03 −0700

On Oct 24, 10:09 pm, Robert Israel
<isr...@xxxxxxxxxxxxxxxxxxxxxxxxxxxxx> wrote:

> Michael Press <rub...@xxxxxxxxxxx> writes:
>
> > In article
> > <1193251799.743613.21...@xxxxxxxxxxxxxxxxxxxxxxxxxxxx>,
> > Marshall <marshall.spi...@xxxxxxxxx> wrote:
> >
> > > On Oct 24, 10:57 am, Michael Press <rub...@xxxxxxxxxxx>
> > > wrote:
> > >
> > > > An algorithm is a procedure that takes input
> > > > and
> > > > terminates with a well−defined and asserted
> > > > result.
> > > > That it does terminate with the asserted
> > > > result must be
> > > > proven. Until we accept the proof, it is not
> > > > an
> > > > algorithm. The notion of right answer is not
> > > > part of
> > > > the definition of algorithm.
> > >
> > > That definition is extremely narrow, and does not correspond
> > > to any usage of the term that I can recall in many years
> > > of being a programmer.
> > >
> > > I'm not even sure I'd buy in to the "well−defined" part.
> > > Having a "probabilistic algorithm" doesn't sound like
> > > a contradiction in terms. In fact, Googling it just

> > > now it gets a lot of hits. Neither does "proven correct,
> > > proven terminating algorithm" sound redundant.

> > I fail to see the utility in defining an algorithm
> > to be no more than a partial recursive function.
> > A theorem is not a theorem until it is proven.
> > What is your standard for implementing a method
> > into production code?

> Whether you see the utility or not is beside the point. The
> fact is that standard terminology in mathematics and theoretical
> computer science does consider a partial recursive function is to be "an
> algorithm", and there is a difference between "algorithm A solves problem P"
> and "we have a proof that algorithm A solves problem P". Similar
> situations exist is most of classical mathematics: objects may have
> certain properties even if there is no proof that they have those
> properties.

although true in many other parts of math
that is not true in rigorous computer science
where properties are strongly operational
due to the inherent relation between syntax and semantics
(galois adjunction)

proof theory is also formulated
very often in this operational form
and generally
this is common when properties evolve in time

some theories of definition
have this built in

−=−
galathaea: prankster, fablist, magician, liar

.