Re: Restrictable Primitives?

Source: http://sci.tech−archive.net/Archive/sci.logic/2008−03/msg00663.html

- *From*: G. Frege <nomail@invalid>

- *Date*: Wed, 12 Mar 2008 01:41:34 +0100

On Tue, 11 Mar 2008 11:50:45 −0700 (PDT), MoeBlee <jazzmobe@xxxxxxxxxxx>
wrote:

Just changed your notation and formatting slightly (I hope you don't
mind):

> For our first order languages, the definition of /term/ is recursive.
> For a language with [...] first order function symbol[s] 'f' ['g', 'h',
> ...], one of the clauses in the definition of /term/ would be:
>
> If t is a term [and f is a function symbol], then f(t) is a term.
>
> Now that is part of a RECURSIVE definition, so everything going ON
> UPWARDS to build more and more terms depends on that clause.
>
> Then for our first order languages, the definition of /satisfies/ is
> recursive.
>
> First, though is the recursive definition of what a term denotes:
>
> One of the clauses would be:
>
> If t is a term [and f is a function symbol], the f(t) denotes the
> result of the function denoted by f applied to the object denoted
> by t. So the function symbol f must map to a function that is total
> on the universe of the model, so that we won't be "jammed" when
> [we] seek the denotation of f(t) for any term t.
>
> And that is part of a recursive definition, so everything going ON
> UPWARDS to determine the denotation of more and more terms depends on
> that clause, and then the recursive definition of /satisfies/ depends
> on the whole recursive definition of the denotation of a term.
>
> For any fuller explanation than this, since I don't have time to just
> pour out an entire chapter of a book in a few posts, you really need
> to read a good book on mathematical logic, where all of this is
> carried out both formally and with an author's explanation.

Well done.

F.

−−

E−mail: info<at>simple−line<dot>de
.