

## Re: An instance of Russell's paradox?

---

*Source:* <http://sci.tech-archive.net/Archive/sci.logic/2005-08/msg00565.html>

---

- *From:* "A.T." <[andrzej-tomaszewski@xxxxx](mailto:andrzej-tomaszewski@xxxxx)>
  - *Date:* 31 Aug 2005 06:32:40 -0700
- 

Ms Knox wrote:

[snip]

>>Ms Knox, may I also ask you to kindly make some remarks regarding the  
>>other two questions I posted, namely, the question suggesting the  
>>infinite generalization of the arity of every predicate (I got this  
>>idea from Quine, but I lost trace of the actual source),  
>  
> Sorry, that's something I'm neither knowledgeable about nor interested  
> in. Do you recall what benefit(s) there might be in making this  
> extension?

I expected that would be uniformity. But, well, I am not a professional, hence my confused queries.

> ISTM that giving up finite expressions would be giving up a  
> lot.

Your argument truly does sound most rational. But the uniformity case seems rational too. I just wish I recalled the passage from Quine where he makes a mention of it. As far as I remember, he put quite a lot of stress on this.

>>as well as the  
>>question pertaining to the actual difference between an atom and a  
>>predicate in philosopher(socrates) transformed into [philosopher,  
>>socrates], where the predicate is(?) really the first atom of the  
>>analogous list form of a proposition (the problem is, I really see no  
>>difference whatsoever).  
>  
> Firstly, there is a difference. In Prolog, the [a b c] representation  
> of lists is actually syntactic sugar for the binary (and nullary) term  
> ".":  
> .(a. .(b. .(c. .())))  
>  
> Similarly, in Lisp the (a b c) representation of lists is sugar for  
> (a . (b . (c . NIL)))

Re: An instance of Russell's paradox?

Yes.

- > Secondly, in Prolog the =.. predicate converts between an arbitrary term
- > such as philosopher(socrates) and the prefix list representation of it.

Yes.

- > These are *\*not\** interchangeable.

No, they are not.

- > Internally to a Prolog implementation
- > terms may indeed be represented as some sort of list, but that is wholly
- > different from the *\*external\** representation of Prolog "[...]" lists
- > using '.' terms.

I am aware of these facts. Thank you very much, Ms Knox.

Please, kindly let me rephrase my question. Listed below are the three alternative Prolog notations for the same term: the dot (internal) notation, the list notation, and the operator notation. In the context of the operator notation we would call "a" – a predicate, and in the other two contexts – an atom (or, better, the first atom on the list). My question is, since these are merely different ways of representing the same algebraic structure, are predicates really atoms (I mean, is it nothing but a terminological convention to call "a" [an atom] in the context of the list and dot notation, whereas [a predicate] in the context of the operator notation)?

```
..(a, .(b, .(c, .())))
[a b c]
a(b, c)
```

Thank you very much for your time (and I am sorry if I have confused something again).

Tom

```
> --
> -----
> | BBB b \ Barbara at LivingHistory stop co stop uk
> | B B aa rrr b |
> | BBB a a r bbb | Quidquid latine dictum sit,
> | B B a a r b b | altum videtur.
> | BBB aa a r bbb |
> -----
```

.

---

• *References:*

Re: An instance of Russell's paradox?

- ◆ *An instance of Russell's paradox?*
  - ◇ *From: A.T.*
- ◆ *Re: An instance of Russell's paradox?*
  - ◇ *From: Jim Spriggs*
- ◆ *Re: An instance of Russell's paradox?*
  - ◇ *From: Barb Knox*
- ◆ *Re: An instance of Russell's paradox?*
  - ◇ *From: A.T.*
- ◆ *Re: An instance of Russell's paradox?*
  - ◇ *From: Barb Knox*

- Prev by Date: *Re: Constructive Math query.*
- Next by Date: *Re: .999... = 1*
- Previous by thread: *Re: An instance of Russell's paradox?*
- Next by thread: *Re: An instance of Russell's paradox?*
- Index(es):
  - ◆ *Date*
  - ◆ *Thread*