# Re: math : the problem with points.

***Source:*** http://sci.tech−archive.net/Archive/sci.math/2008−04/msg00928.html

---



- *From*: Julio Di Egidio <julio@xxxxxxxxxxxxx>
- *Date*: Sun, 06 Apr 2008 10:50:44 EDT



---

Tim Little wrote:

> On 2008−04−05, Julio Di Egidio <julio@xxxxxxxxxxxxx> wrote:
>
>> If we get into the domain of computable numbers (actual numbers), and take closed interval arithmetic as reference, that lower limit (actually) exists and it is the floating−point limit to underflow, intrinsic to any specific implementation. It is, in fact, under all respects, a rational number.
>
> No, there are plenty of computation systems that have no such element and do not have underflow.

Well, you quote me out of context, and that might be the guise of the problem. I have shown that such a lower limit _exists_ under specific _circumstances_, that is, it exists in the strongest sense, and, that I am not the only one who has encountered it, you confirm by mentioning floating−point implementations in the real world. Aren't they indeed a pretty widespread family? This given, we also have a trivial counter−proof to the general argument "such an entity does not exist", which happened to be the major argument around this topic.

In any case, I must tell you I am very interested in, and slowly slowly trying to digest, what is coming out from these discussions, namely on the status of the so called "points". In the meantime, to fill an apparent gap, please let me share some further view at closed systems, as they are inductive, yet counter−intuitive:

(1) "A closed number system is and only is a closed logical system."
(2) "Overall, a closed system exhausts the domain of tractability."

Assertion (1), about the equivalence between logical and number systems, is easily, though maybe "unsharply", proved by the existence of this very argument: I am (tentatively) using "interval−logic" right here, to set up an argument. This means, in analogy with our previous discussion on the existence of such and such points, that there is also embedded a counter−proof to the general non−existence argument, though here it is the significant one, itself a revenge of inductive logic: interval−logic equivalent inductive−mathematics, and it exists.

By trivial interval−logic, then follows assertion (2), about closed systems exhausting the whole domain of tractability: a result, however unsharp, satisfying containment, so correctness. The point of containment is capital to understand how closed systems stay always correct, while "sharpness" becomes a matter of "quality of the implementation" (and, in a sense, a function of "time"): though it might not be that useful to always get an Empty, correctness is always preserved. The sharper result here (remember: Empty is prevalent to Entire), is in recognizing that "tractability exists", again by trivial inductive argument: induction is not deduction, I guess.

A sharp connection to non−trivial logic (and to non−arithmetical mathematics, I suppose), might be postulated in a sharpening of our concept of "computing systems". An equivalence between Turing machines and Church's lambdas has long been proved, and I'd rather ask if computation systems exist that even escape the reach of such models I come to mention... Yep, maybe I have already got it: "intelligent systems". Up to now, these are simply an "open question", i.e. Empty.

Few basic questions, if you won't mind:

> Exact arbitrary−precision rational
> arithmetic, for example.

I wander how can that be different, in principle, from closed intervals? Moreover, do they have something in analogy to the containment concept? That is, how do they "behave" with respect to correctness? Overall, what is the improvement with respect to closed systems, if any.

> There are even systems that can handle some exact irrationals and even
> transcendentals.

I wander how this can imply systems beyond the Turing−Church's concept: _that_ would be a massive surprise to me, given the potential implications.

> Fixed−precision floating point numbers are just one system that has
> fast hardware support in practice, and certainly doesn't define all
> computational possibilities.

Yet I am using it right now, to implement a simple closed system, namely playing with visualizing basic transformations, and I must tell you it seems already enough, if not ideal, to get the most basic properties of such closed systems, indeed a great support to my own learning on the matter.

Thank you very much for the feedback.

Julio
.