

Re: How Lisp's Nested Notation Limits The Language's Utility

Source: <http://coding.derkeiler.com/Archive/Lisp/comp.lang.lisp/2007-05/msg00557.html>

- *From:* Kent M Pitman <pitman@xxxxxxxxxxx>
 - *Date:* 09 May 2007 11:41:16 -0400
-

[comp.lang.lisp only.
<http://www.nhplace.com/kent/PFAQ/cross-posting.html>]

David Hansen <david.hansen@xxxxxxxx> writes:

On Wed, 9 May 2007 16:24:41 +0200 Kjetil S. Matheussen wrote:

For larger mathematical expressions, I think s-expressions are much better because you can use more lines and indentations to make things clearer.

Show me one person that uses sexps to scribble large equations by hand.

There's a fallacy in your question because it presupposes that large expressions are constructed by scribbling. They often aren't. Constructing large expressions by scribbling is error-prone and not best practice.

It also supposes that the people who do this construction by hand do the full set of operations on their expressions that we do on ours. They may do some of that symbolic rearranging in their head, but they may also just deny themselves options that are not easy to administer.

In this regard, it reminds me of writers who stubbornly resisted word processors for so long, claiming they preferred fixed pieces of paper for editing. It seems to me objectively clear that the computer / word processor is a better device because of its ability to save old copies, to compare copies, to rearrange large blocks, and in every way to enable "trying alternatives" to make sure what you have is good. Those who manipulate symbolic algebra without benefit of a mechanical tool to assure the integrity and validity of the transforms they do are really just hampering themselves.

Infix notation is, objectively, a device for compact depiction of a

Re: How Lisp's Nested Notation Limits The Language's Utility

result, and for people who don't want to type a lot. The compact display has its purpose, and compact input may have its purpose, but ultimately, trying to argue that there's no purpose in having other display forms is like trying to argue that mathematical transforms have no purpose.

Or, if you prefer a more obscure analogy, infix algebra is like clinging to non-realtime TECO after WYSIWYG editors came along. [I'd use the the correct names of the tools here, but it would obscure my analogy, which is obscure already.]

Another problem problem with infix is that its irregular rules of presentation obscure structural regularities that would probably produce great insights, but that are hidden by the non-parallel nature of the notation [in the English grammar sense of parallel construction; no allusion to computational parallism here] .

Also, there are lots of people that use parens around algebraic notation because they don't understand and trust "precedence".

$((a*b)+c)/d$

is not an uncommon style for people to adopt.

And, there are people like me who don't like doing math at all. I prefer to hand it over to a computer to do it for me. Algebraic notation is fine when the task I want to farm out is just "compute this". But it's not when the task is more complicated.

And, also, even when editing things myself, I frequently find that it's hard to safely and reliably rearrange an expression to understand it. If you see an algebraic expression that goes on for pages, it's nearly unintelligible. (I deal with symbolic algebra systems all day for work and I have this problem every day. I have to stop and indent the expression Lisp-style in order to make any sense of it, and that's a complicated and tedious process.)

.