# Re: Reason for operator precedence

***Source:*** http://sci.tech−archive.net/Archive/sci.math/2006−03/msg02474.html

---

- *From*: matt271829−news@xxxxxxxxxxx
- *Date*: 14 Mar 2006 12:48:10 −0800

---

briggs@xxxxxxxxxxxxxxxxx wrote:

> In article <1142360037.330590.76450@xxxxxxxxxxxxxxxxxxxxxxxxxxxx>,
> matt271829−news@xxxxxxxxxxx writes:
>
>> briggs@xxxxxxxxxxxxxxxxx wrote:
>>
>>> In article
>>> <1142344511.262841.322440@xxxxxxxxxxxxxxxxxxxxxxxxxxxx>,
>>> matt271829−news@xxxxxxxxxxx writes:
>>>
>>>> bri...@xxxxxxxxxxxxxxxxx wrote:
>>>>
>>>>> In article
>>>>> <1142342196.542632.294210@xxxxxxxxxxxxxxxxxxxxxxxxxxxx>,
>>>>> matt271829−news@xxxxxxxxxxx
>>>>> writes:
>>>>>
>>>>>> Tony wrote:
>>>>>>
>>>>>>> Hi
>>>>>>> all.
>>>>>>>
>>>>>>> Hope
>>>>>>> this
>>>>>>> isn't
>>>>>>> a
>>>>>>> silly
>>>>>>> question.
>>>>>>>
>>>>>>> I
>>>>>>> was
>>>>>>> wondering
>>>>>>> what
>>>>>>> the
>>>>>>> reason

is for having multiple levels of operator precedence?

Phrased another way, why is it that we don't just evaluate everything from left to right?

Having multiple levels of precedence obviously adds complexity, so I assume there must be some payback. However, I don't see what it is.

As far as addition/subtraction vs multiplication/division is concerned, one reason is to ensure that the distributive property of multiplication works sensibly. For example, we want 3*(4 + 6) = 3*4 + 3*6 = 3*(6 + 4) = 3*6 + 3*4.

Remember that what we're talking about here is merely a notational convention. It has nothing whatsoever to do with the distributive property of multiplication over addition.

You can express the distributive law for multiplication over division using parentheses:

a*(b+c) = (a*b) + (b*c)

Obviously you can. I meant to make it work without needing parentheses,
but it seems that wasn't clear.

Ok. Try doing it using infix notation and the operator precedence
convention of your choice. Remember your rule: no parentheses

> > > Left to right doesn't work.
> > >
> > > b+c*a = a*b... and we're stuck
> > >
> > > Right to left doesn't work.
> > >
> > > b+c*a = ...b*c and we're stuck.
> > >
> > > Multiplication has precedence over addition doesn't work.
> > >
> > > a*... and we're stuck
> > >
> > > Addition has precedence over multiplication doesn't work.
> > >
> > > a*b+c = a*b+... and we're stuck
> > >
> > > Accordingly, trying to point to this case as a motivation for some
> > > particular choice of operator precedence seems ill conceived.
> > >
> > > According to your argument, it follows that we are all using either
> > > Polish (prefix) or Reverse Polish (postfix) notation.
>
> > Sorry, you've lost me. I was agreeing with you that even without any precedence convention we could still represent the distributive law as a*(b + c) = (a*b) + (a*c). However, the convention makes the parentheses redundant, because a*b + a*c is understood to mean (a*b) + (a*c).
>
> Convention makes _SOME OF THE_ parentheses redundant. Your claim is that _ALL OF THE_ parentheses are redundant.

No, only the right hand side. It's just a trivial observation. With no convention for precedence we would have to write a*(b + c) = (a*b) + (a*c), but given the usual precedence convention we can just write a*(b + c) = a*b + a*c. That's it, there's nothing more to this part, honest.

More interesting is the reason why the distributive property should be relevant to the original question, which did not come over at all in my original post, I regret to say. Somehow I feel that this at least part of the reason why we naturally want to bind * more tightly than +. I feel that if the operators did *not* have this property then there would be less reason for the convention.

Please respond to the challenge above. Try to phrase the distributive law without using parentheses. If you have to resort to prefix or postfix notation, my case is made. If you can't do it at all, your case is lost.

.