BODMAS, BOMDAS and DAMNUS ... the sequel

Greg Gamble
Curtin University
G.Gamble@curtin.edu.au

This article is a “reaction” to the Jane Watson article of a similar name in the last Cross Section issue.

Introduction

Firstly, a general comment with regard to mnemonics. Our brains find it a little difficult, at least in the short term, to remember more than 6 or 7 items. Mnemonics can help by bundling several items into one, which, of course is much easier to remember. However, many, if not most, mnemonics also require some sort of “support” information, without which the mnemonic is of little benefit, and so it is with the BODMAS mnemonic and its variants, but more on that later.

Purpose of the BODMAS mnemonic

Why have rules for order of operations at all, if we can just add brackets to show the order we intend? Essentially we have these rules to avoid clutter. An expression only needs to be mildly complicated before nesting of brackets, and therefore matching of brackets becomes a bit of an issue, (and note that square brackets and braces are not used for this sort of grouping by calculators and computers).

The variants and what they mean

My first issue with the Jane Watson article is in what the O of BODMAS stands for. Rather than “Of” (as, implied Multiplication), which apparently it was originally, these days, the improved interpretation is “Orders” as a synonym for “Indices”. So BODMAS abbreviates:

> Brackets, Orders, Division, Multiplication, Addition, Subtraction

but then some use BOMDAS which has Division and Multiplication in the opposite order. Then there is BIMDAS which is the same as BOMDAS with the word “Indices” instead of “Orders“ (this is my preferred version). Then the Americans have the phrase:

> Please Excuse My Dear Aunt Sally

which very cutely suggests

> Parentheses, Exponents, Multiplication, Division, Addition, Subtraction

where “Parentheses” is the more correct term for round brackets and “Exponents” is yet another synonym for “Indices”.

But hang on, how can it be that some people use an acronym with Multiplication and Division in one order, and others use one in the opposite order. They can't both be right, can they? Well, this brings me to my introductory comment, that a mnemonic without its “support” information is of little benefit:

> Multiplication and Division have the **same** precedence!

and despite the fact that all the acronyms end in AS (and not sometimes SA - around the other way, doesn't exactly roll off the tongue),

> Addition and Subtraction also have the **same** precedence!

So BIMDAS (my preferred version, as I said before) should really be recalled as saying:

Brackets, Indices, Multiplication and Division, Addition and Subtraction

where each comma is read as a "then", and

when the above doesn't tell you what to do first evaluate from left to right.

As I said before, without the "support" information the mnemonic is pretty much useless.

I'm delighted to say that when I googled for: `YouTube BODMAS` that I got the link:

http://www.youtube.com/watch?v=Z-FKjqL6NyQ

and the people involved do a wonderful job of singing (tunefully) the explanation above correctly, and all in 1 minute and 27 seconds.

Multiplication and Division, same precedence? ... Explain!

Now we seem to have opened a can of worms. How can it be that Multiplication and Division have the same precedence? Well ... if one understands Division as a sort of Multiplication in disguise, it starts to make sense. Whenever we divide by x, equivalently we can multiply by its reciprocal $1/x$. Thus dividing by 2 is the same as multiplying by $1/2$.

Similarly, subtracting x is equivalent to adding its negative $-x$. So subtracting 2 is the same as adding -2.

So instead of having the four operations: $\times, \div, +, -$ we could say we just have the two: $\times, +$.

Thinking this way has some side benefits, since both these operations have the properties of being commutative and being associative. These operations being commutative means that:

$$a \times b = b \times a$$
and $$a + b = b + a$$

and their being associative means that:

$$(a \times b) \times c = a \times (b \times c)$$
and $$(a + b) + c = a + (b + c)$$

so that even without having a BIMDAS (or variant) rule, brackets are not needed for telling us how to evaluate

$$a \times b \times c$$
and $$a + b + c$$

Calculators

The Jane Watson article claimed that different calculators would apply different rules of precedence. This was quite possibly true at the time the article was written, but today you would be hard-pressed finding even a $20 calculator that doesn't have the BIMDAS rules built in, which is a bit of a shame, really; I'd rather the little kiddie-winkles were forced to think about how they entered expressions into their calculators, rather than leave things to blind chance, as some seem to! At this point I think I'd like to relate a history of calculators from a personal experience point of view.

Quite a few years ago, I was working full-time as an Engineering Assistant (I was at uni. part-time) and Barry, a somewhat older Technical Officer, came up to me with a calculator that was apparently giving him the wrong answer. I suppose I was perceived as the young bright spark who would be able to explain it, and as it happened I could. Anyway, Barry keyed in:

$3 + 4 \times 5 =$

and the calculator showed 35. Barry, knowing his order of operations, “knew” this should have been 23. Anyway, I explained that the calculator didn't know order of operations rules, and so to enforce the rules he would need to insert the brackets manually; even the early calculators had those! So he needed to enter:

$3 + (4 \times 5) =$

or, alternatively, he could rearrange the expression first using the commutativity of addition and enter:

$4 \times 5 + 3 =$

I further explained what he was seeing in the calculator's display as he was pressing the keys. The early calculators did have a memory key which when used put what was currently in display into a memory where it could be recalled later using the RM (Recall Memory) key and usually also had a capability for up to 6 levels of brackets - but maybe that was a fairly high-end model. Otherwise, they only used the display and an unseen register - a stack - for calculations. The keys Barry pressed caused the following

3 ... 3 appears in the display
\+ ... as a side-effect signals the number in the display is complete
(no more digits for it are coming)
4 ... 3 gets moved to the stack and 4 appears in the display
$\times$... as a side-effect signals the number in the display is complete
so that the calculation of 3 + 4 can proceed, and does, so that
7 appears in the display
5 ... 7 is moved to the stack and 5 appears in the display
= ... signals the number in the display is complete and
evaluation of the stack number times the display number,
i.e. 7×5, can proceed, and 35 appears in the display.

Only a few years later, Texas Instruments and Hewlett Packard were the leaders in producing calculators, and anyone doing Engineering had one or the other. The TI calculators touted AOS - Algebraic Order of Simplification, i.e. it had BIMDAS built in. HP, on the other hand, had Reverse Polish, explaining that with this, one didn't need any brackets. You will find any HP calculator you get today, still has a mode for Reverse Polish, but also applies BIMDAS outside this mode.

At the time Jane wrote her article, I imagine only $5 wafer thin solar-powered (and hence batteryless) calculators might not have had BIMDAS built in.

So what's Reverse Polish? Well ... most of the operations we use are binary, that is to say they operate on two operands, and, in fact there are three ways we could define binary operations: *prefix*, *infix*, or *postfix*.
For *prefix*, the operation comes first, then the operands, e.g. $+(a,b)$, the "+" of a and b.
For *infix*, the way we usually use, the operation is "in between", e.g. $a+b$.
For *postfix*, the operation comes at the end, e.g. $(a,b)+$, a and b are plussed.
Essentially, Reverse Polish is postfix. “But hang on”, you say, “that has brackets all over the place”.

What I haven't told you yet, is that you don't actually enter the brackets, but you may have to use the `Enter` key as a separator, i.e. as the comma, between the arguments. So, 4 × 5 + 3 (infix) would be entered:

4 `Enter` 5 × 3 + ... which in postfix is: ((4,5) ×,3)+

with no need for an "=" key. I think HP had a good idea, because it made the user think about what they were entering. I fear that the modern calculator, allows students to think they don't need to know the order of operations, because the calculator will do it all for them.

A common error

In the past, we used whitespace for grouping a lot. The trouble with this, is that electronic technology largely ignores whitespace. So when we write

$$\frac{a}{b \times c}$$

and then gradually make the vinculum (the line between numerator and denominator) more and more oblique we may arrive at

$$a / b \times c$$

and still expect that $b \times c$ is in the denominator, but it isn't! (Of course, students make a similar error when × is replaced by +.)

Now, according to BIMDAS, we evaluate from left to right, because / and × have the same precedence. So, in fact, $a / b \times c$ is the same as

$$\frac{a \times c}{b}.$$

Here's a detailed analysis, which you may like to review after reading the next couple of paragraphs.

$a / b \times c = a \times (1/b) \times c$ ..... dividing by b is the same as multiplying by $1/b$
$= a \times c \times (1/b)$ ..... multiplication is associative and commutative
$= (a \times c) \times (1/b)$ .. multiplication is associative
$= (a \times c)/b$ .... dividing by b is the same as multiplying by $1/b$

Factors and Terms

In an expression of form

$$a \times b \times c$$

a, b, c are called factors. Recall we said before that Division is just Multiplication in disguise. So, therefore an expression of form

$$a \times b / c$$

can also be said to have factors and they are $a, b, 1/c$, since the expression written with only multiplications would be

$$a \times b \times (1/c).$$

Also, in an expression of form

$$a+b+c$$

a,b,c are called terms. Again, recall that we said before that Subtraction is just Addition in disguise. So, therefore an expression of form

$$a-b+c$$

can also be said to have terms, and they are $a,-b,c,$ since the expression written with only additions would be

$$a+(-b)+c$$

So, after we have dealt with the Brackets and Indices we are left with a sum of terms, where each term is a product of factors.

An exercise

Enter the following expression as you would need to into a calculator, and try to do so, with as few brackets as possible:

$$\frac{u+\dfrac{u}{u+1}}{\dfrac{1}{u+1}+1}+\frac{1}{u}$$

This is a problem I give university students as an introductory exercise for our on-line quiz system, and most students find it hard.

The line between a numerator and a denominator is called a vinculum. Essentially, what we are doing is replacing horizontal vinculums which imply a certain grouping with oblique ones (/) which don't.

The key idea is that it follows from the BIMDAS rules, that if a numerator or denominator consists of more than one term it needs to be bracketed, or otherwise they are superfluous (when we use / rather than a horizontal vinculum).

So we see

$$\frac{u}{u+1}=u/(u+1)$$

and

$$\frac{1}{u+1}=1/(u+1).$$

This reduces the problem to the sum of two fractions where the left fraction consists of a numerator and a denominator, each consisting of two terms, and therefore needing bracketing:

$$\frac{u+\dfrac{u}{u+1}}{\dfrac{1}{u+1}+1}+\frac{1}{u}=\frac{u+u/(u+1)}{1/(u+1)+1}+\frac{1}{u}$$

$$=(u+u/(u+1))/(1/(u+1)+1)+1/u$$

Computer programming languages

Computer programming languages have to deal with a vast number more operations than what is encountered in an algebraic expression. In particular, there is the `mod` (or remainder) operation, logical `and` and `or`, but, in my experience they all agree with respect to the operations mentioned above. With respect to the extra operations, there are differences, and so, in these cases, one should err on the side of safety and use brackets when in doubt, or for portability of your computer program.

Is this the end of the story?

The short answer is "No". Where do functions fit in? (e.g. $\sin(x)^2$ - Maple treats it as: $(\sin(x))^2$, which is sensible, since the other interpretation - which I have also seen - can be written: $\sin(x^2)$.) And, there is a special rule for multiple levels of indices, not covered by BIMDAS. For

a^{b^C} … evaluation does not proceed from left to right!

Why not? Essentially,

$$(a^b)^c = a^{bc}$$

by the Index Laws, and the righthand expression is surely the way we would prefer to write $(a^b)^c$.

On the other hand,

$$a^{b^C}$$

doesn't have a nice other way to write it, except for leaving the brackets out, if indeed that is the convention adopted (and it is).

Wikipedia identifies two calculators that differ in regard to evaluation of a^{b^C} . So, to be safe, in this case, one should insert brackets when entering into a calculator.

Other reading

Wikipedia is a wonderful resource, and, particularly for mathematics, can generally be relied upon for accuracy. When I visit wikipedia, as I do often, I get a surge of feeling that altruism is alive and well. Check out: http://en.wikipedia.org/wiki/Order_of_operations
It covers everything I have mentioned and more.

Conclusion

I'm hoping after you've read this, you agree that BIMDAS (or BODMAS or whatever) is a good start, which with associative links to quite a lot of "support" information, helps you get closer to the full story. And the "word" to think about in association with calculators is GIGO (Garbage In Garbage Out); always have some sort of check, so that you have some confidence of the correctness of the output, e.g. rounding off your numbers and doing a quick mental check is good for checking your answer is about the right size.

References

Watson, J.M. (2010), BODMAS, BOMDAS and DAMNUS, *Cross Section* Vol. 20 No. 4, 6-9, (reprinted from a 1995 article).
Wikipedia: http://en.wikipedia.org/wiki/Order_of_operations
YouTube: http://www.youtube.com/watch?v=Z-FKjqL6NyQ