11-2016

Why Multiplication Has Higher Priority than Addition: A Pedagogical Remark

Olga Kosheleva
University of Texas at El Paso, olgak@utep.edu

Vladik Kreinovich
University of Texas at El Paso, vladik@utep.edu

Comments:
Technical Report: UTEP-CS-16-81

Why Multiplication Has Higher Priority than Addition: A Pedagogical Remark

Olga Kosheleva[1] and Vladik Kreinovich[2]
[1]Department of Teacher Education
[2]Department of Computer Science
University of Texas at El Paso
500 W. University
El Paso, TX 79968, USA
olgak@utep.edu, vladik@utep.edu

Abstract

Traditionally, multiplication has higher priority over addition; this means that there is no need to add parentheses if we want to perform multiplication first, and we need to explicitly add parentheses if we want addition to be performed first. Why not use an alternative arrangement, in which addition has higher priority? In this paper, we explain the traditional priority arrangement by showing that in the general case, the traditional arrangement allows us to use fewer parentheses than the alternative one.

1 Why Multiplication Has Higher Priority than Addition: Formulation of the Problem

Multiplication has higher priority than addition: a reminder. In our usual arithmetic notations, multiplication has priority over addition. This means that if the arithmetic expression has no parentheses, e.g., has the type $a + b \cdot c$, then we:

- first multiply b and c, and then
- add a to the resulting product.

If we want to perform addition first, then we have to add parentheses. For example, if we want to first add a and b, and then multiply the sum by c, then we have to write an expression $(a + b) \cdot c$.

Why? A natural question is: why does multiplication have higher priority than addition? Why not consider addition a higher-priority operation, so that:

- the expression $a + b \cdot c$ would mean $(a + b) \cdot c$, and
- to describe the case when we want b and c to be multiplied first, we should write $a + (b \cdot c)$?

Is the current priority arrangement due solely to historical reasons or there is a more fundamental reason behind this arrangement?

What we do in this paper. In this paper, we provide a computation-based explanation of why multiplication has higher priority than addition.

2 Why Multiplication Has Higher Priority than Addition: Computation-Based Explanation

Main idea behind our explanation. We could add parentheses for all the operations. In this case, $a+b\cdot c$ would have the form $a+(b\cdot c)$. The reason why priority is introduced in the first place is to decrease the number of parentheses as much as possible.

It is therefore reasonable to analyze which of two possible arrangements requires fewer parentheses:

- the usual arrangement in which multiplication has higher priority than addition, or
- an alternative arrangement in which addition has higher priority than multiplication.

What can we compute based on addition and multiplication: case of one variable. To analyze which of two operations – multiplication or addition – should have higher priority when computing complex expressions, let us consider what expressions we can get by using addition and multiplication.

Let us consider the simplest case when we have a single variable x. In addition to this variable, we can use several constants. We form expressions by applying addition and multiplication to this variable x and to different constants.

One can easily prove, by induction, that as a result, we get polynomials. Indeed:

- the variable x itself and all constants are particular cases of polynomials, and
- when we add or multiply two polynomials, we always get a polynomial.

Vice versa, each polynomial $f(x)$ of one variable can be described as

$$f(x) = a_0 + a_1 \cdot x + a_2 \cdot x^2 + \ldots + a_n \cdot x^n, \tag{1}$$

and thus, can be obtained from the variable x and from the constants a_i by using addition and multiplication, as

$$f(x) = a_0 + a_1 \cdot x + a_2 \cdot x \cdot x + \ldots + a_n \cdot x \cdot \ldots \cdot x \ (n \text{ times}). \tag{2}$$

For the canonical polynomial expression, it makes computational sense to assign higher priority to multiplication. The canonical expression (1)-(2) for a polynomial, when expressed in terms of traditional priority arrangement, does not need any parentheses.

On the other hand, if we try to describe the same expression (1) in an arrangement in which addition has higher priority than multiplication, then we need to add a large number of parentheses:

$$f(x) = a_0 + (a_1 \cdot x) + (a_2 \cdot x \cdot x) + \ldots + (a_n \cdot x \cdot \ldots \cdot x) \ (n \text{ times}). \qquad (3)$$

Actually, we need to add as many parentheses as we would need if there was no priority arrangement at all.

Thus, while considering multiplication as a higher-priority operation does not require any parentheses at all, the alternative arrangement – when addition has higher priority – would require a large number of parentheses, actually, the same number that we would need if we did not have any priority arrangement at all. From this viewpoint, assigning higher priority to multiplication is definitely a preferable option.

What if we take into account how polynomials are actually computed? The above explanation may not be fully convincing: it is based on the way we write down the polynomial expression, but this is *not* the way polynomial expressions are actually computed.

It is known that the most efficient way to compute the value of a polynomial of one variable is to use the Horner scheme; see, e.g., [1]. In the usual notations, with multiplication having higher priority, the Horner scheme takes the form

$$f(x) = a_0 + x \cdot (a_1 + x \cdot (a_2 + \ldots + x \cdot (a_{n-1} + x \cdot a_n) \ldots)). \qquad (4)$$

For $n = 1$, we have

$$f(x) = a_0 + a_1 \cdot x \qquad (5)$$

with no need for parentheses. For $n = 2$, we have

$$f(x) = a_0 + x \cdot (a_1 + x \cdot a_2), \qquad (5)$$

with one pair of parentheses. For $n = 3$, we have

$$f(x) = a_0 + x \cdot (a_1 + x \cdot (a_2 + x \cdot a_3)), \qquad (6)$$

with two pairs of parentheses. In the general case, for a polynomial of n-th degree, we need an expression (4), with $n - 1$ pairs of parentheses.

What will happen if we consider addition to be a higher-priority operation? In this case, for $n = 1$, we will need an expression

$$f(x) = a_0 + (x \cdot a_1), \qquad (7)$$

with one pair of parentheses. For $n = 2$, we will need an expression

$$f(x) = a_0 + (x \cdot a_1 + (x \cdot a_2)), \tag{8}$$

with two pairs of parentheses. For $n = 3$, we will need an expression

$$f(x) = a_0 + (x \cdot a_1 + (x \cdot a_2 + (x \cdot a_3))), \tag{9}$$

with three pairs of parentheses. In the general case, for a polynomial of n-th degree, we need n pairs of parentheses:

$$f(x) = a_0 + (x \cdot a_1 + (x \cdot a_2 + \ldots + (x \cdot a_{n-1} + (x \cdot a_n)) \ldots)), \tag{10}$$

with n pairs of parentheses.

So, for every n:

- the usual arrangement, when multiplication has higher priority, requires $n - 1$ pairs of parentheses, while
- the alternative arrangement, when addition has higher priority, requires n pairs of parentheses.

Since $n > n - 1$ for all n, the traditional arrangement requires fewer parentheses and is, thus, indeed preferable.

3 Conclusions

In this paper, we explain the traditional priority arrangement – when multiplication has higher priority than addition – by showing that, in the general case, this arrangement requires fewer parentheses than the alternative arrangement, when addition is given higher priority than multiplication.

Acknowledgments

This work was supported in part by the National Science Foundation grants HRD-0734825 and HRD-1242122 (Cyber-ShARE Center of Excellence) and DUE-0926721, and by an award "UTEP and Prudential Actuarial Science Academy and Pipeline Initiative" from Prudential Foundation.

References

[1] D. E. Knuth, *The Art of Computer Programming*, Addison-Wesley, 2011.