Syntax and Semantics of Numerical Language for Elementary School

Diego Pareja-Heredia. ***Universidad del Quindío***

Abstract

Numerical language begins its development at early stages of infancy. This development goes parallel with colloquial language. However, traditional math instruction does not give enough relevance to the syntax and semantics of numerical language.

In this paper we want to show, through some examples, how syntax and semantics are important for the comprehension of numbers, their properties and the operations defined on them.

Introduction. Along these lines, the word number will be used to mean *natural number*, unless we say otherwise.

Numerical language has a structure with its own syntax, as ordinary language does. The alphabet for this language, in the case of decimal representation, is the set {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}. With this alphabet we can form strings, or numerals, say, $(a_n a_{n-1} \ldots a_1 a_0)$, each of them represents a particular number. These strings of digits can be seen as polynomials of one variable x:

$$(a_n a_{n-1} \ldots a_1 a_0) = a_n x^n + a_{n-1}x^{n-1} + \ldots + a_0 x^0 = \sum_{j=0}^{j=n} a_{n-j}x^{n-j} \qquad [\ 1\ ]$$

With the syntactic property

$$x^n = 10x^{n-1}. \qquad [\ 2\ ]$$

We note that if $n = 1$, then, $x = 10$. This just means that our numerical language is decimal.

Property [2], above, gives our language the characteristic of being a positional system. This means that the n-position in the numeral, counted from right to left (from 0 to n) has a value 10 times the value of $(n - 1)$-position. We name those positions: units, tens, hundreds, thousands and so on.

Early as kids, we discover that *number words* give us some information about each number. However, we didn't note that this information could be used to go inside the properties which characterize that number or to a set of numbers. When we heard, for instance, the number words: *two hundred fifty six* and one *hundred twenty eight*, we just translate these words to written language as 256 and 128. However, with some syntax knowledge from our numerical language, we could immediately translate them to:

$$P(x) = 2x^2 + 5x + 6 = 2x^2 + 4x + (x + 6) = 2x^2 + 4x + 16 = 2(x^2 + 2x + 8) = 2Q(x),$$

$$Q(x) = x^2 + 2x + 8 = 10x + 2x + 6 = 12x + 8 = 2(6x + 4) = 2^2(3x + 2) = 2^2(32) = 2^2 2^5 = 2^7.$$

And then we can conclude, among other things: 1) both numbers 256 and 128 are even; 2) the first number is twice the second and 3) $256 = 2\times128 = 2^8$.

Syntax and Semantics for a Numerical Language.

Inside the above paragraph we can find some typical mathematical sentences, like

a) $P(x) = 2Q(x)$

b) $Q(x) = 2^7$

c) 256 and 128 are even and $256 = 2^8$.

d) From the truth of *a*) and *b*) we conclude *c*).

These sentences together conform, a simple mathematical argumentation, or a proof of the type: $p \wedge q \rightarrow r$.

While in sentences *a*), *b*) and *c*) it is prevalent the syntax of the language, in *d*) the predominant feature is semantics. Semantics here, gives meaning to sentences when we are speaking about truth.

In math language syntax is the set of rules governing the well formation of numbers and sentences related to them.

Besides the numerical alphabet mentioned above, the numerical language requires:

1. – A list of variables and a list of symbols for logic connectives. Along this short mathematical discourse we have used so far, variables like *P, Q, x, p, q, r*, and symbols for connectives such as: $\wedge$ and $\rightarrow$, meaning conjunction and implication.

2. – The equality sign (=). This sign let us create chains of symbols, such as: A = B = … = Z . When in A we define one or more arithmetical operations, as in **[1]**, the symbols B, …, Z, could be the sequence of steps to arrive at the end of the process. If we borrow some terms from the computer language, we can say that A receives the input and Z gives the output.

3. – Grouping signs. Grouping signs like (), [], {} are used in mathematics sentences, to join together symbols or math expressions.

4. – Quantifiers. [Universal (∀) and Existential (∃)]. Quantifiers in a language limit the variability of variables.

In the case of $P(x)$ and $Q(x)$, mentioned above, variable x could take any value in the set of real numbers, symbolically: $\forall(x)P(x)$, and, $\forall(x)Q(x)$. However, these polynomials would not always represent a natural number. For number representation would be better use $\exists(x)P(x)$, and, $\exists(x)Q(x)$, since we can choose $x = 10$, to get the required numbers.

5. – Primitive Parameters. In formalized number theory is usual to have some primitive parameters, or undefined elements such as: 0, 1, +, ×, to design operations and important arithmetic elements[1]. We have used expressions like

$$P(x) = 2x^2 + 5x + 6, \text{ and, } Q(x) = x^2 + 2x + 8$$

Where all primitive parameters are implicitly or explicitly included. For instance, $P(x)$ and $Q(x)$ really means:

$$P(x) = (1 + 1)\times x\times x + 5\times x + 6\times x^0 = 2\times x\times x + 5\times x + 6\times 1, \text{ and, } Q(x) = x\times x + (1 + 1)\times x + 8\times x^0.$$

In both cases all primitive parameters are present explicitly.

Example. Let us recall that Decimal Polynomials are those which all their coefficients are taken from the set {0,1,2,3,4,5,6,7,8,9}. These polynomials uniquely represent a number. However, we can find families of *no decimal polynomials* equivalent to the same number.

Let's take the number $s = 53$. The associate decimal polynomial for s is $S(x) = 5x + 3$. Others no decimal polynomials representing s, are: $4x + 13$, $3x + 23$, $2x + 33$, $x + 43$ and also 53. All of them are different polynomials, nevertheless when we change x for 10 we find, in all cases, the number 53.

And let's try now with $t = 2647$ to write its decimal polynomial and find some other polynomials equivalent to t.

The only decimal polynomial associated with t is: $T(x) = 2x^3 + 6x^2 + 4x + 7$.

Applying the syntactic property **[2]** we can replace x^3 by $10x^2$ and we get:

$2647 = T(x) = 2x^3 + 6x^2 + 4x + 7 = x^3 + x^3 + 6x^2 + 4x + 7 = x^3 + 10x^2 + 6x^2 + 4x + 7 = x^3 + 16x^2 + 4x + 7$. This last polynomial is numerically equivalent to $T(x)$.

It follows that: $2647 = 10^3 + 16\times 10^2 + 4\times 10 + 7 = 1000 + 16\times 100 + 4\times 10 + 7 = 1000 + 1600 + 40 + 7$.

[1] See my Epistemology notes in: http://www.matematicasyfilosofiaenelaula.info/Epistemologia%202009/David%20HIlbert%20y%20el%20Formalismo.pdf

Also we can express $T(x) = 2x^3 + 6x^2 + 4x + 7$, as, $2x^3 + 5x^2 + x^2 + 4x + 7 = 2x^3 + 5x^2 + 10x + 4x + 7 = 2x^3 + 5x^2 + 14x + 7$ and taking, $x = 10$, we show that: 2647 = 2000 + 500 + 140 + 7.

Therefore we can practice addition and multiplication without using the traditional algorithms, but transforming the decimal polynomial with the help of the basic syntactic property [2].

The Inherent Structure inside Decimal Representation. Instead of engaging in repeating arithmetical algorithms over and over again, it could be more rewarding for students to understand the structure which support decimal representation of numbers, whose core is condensed in formula [1].

In the expression

$$(a_n a_{n-1} \ldots a_1 a_0) = a_n x^n + a_{n-1} x^{n-1} + \ldots + a_0 x^0 = a_n 10^n + a_{n-1} 10^{n-1} + \ldots + a_0 10^0 \qquad [\ 3\ ]$$

We are defining the number whose digits are $a_n a_{n-1} \ldots a_1 a_0$, in terms of a sequence of products and sums. The "products" here are the most simple of all: the multiples of powers of ten. Consequently, the "sum" will be easy to calculate. The *number words* that names the number, will give us the answer.

Let's take 479, whose number words are: *Four Hundred Seventy Nine*.

$479 = 4 \times 10^2 + 7 \times 10^1 + 9 \times 10^0 = 4 \times 100 + 7 \times 10 + 9 \times 1 = 400 + 70 + 9 =$ (four hundred, seventy, nine; omitting the plus sign).

According to our syntax, the *digits string 479* has a value of: 400 + 70 + 9 and according to our semantics we receive a meaning of *Four Hundred Seventy Nine*.

Inner Product. In [3] we observe some regularity around the composition of digits and powers of ten. The first digit is associated with the highest power of ten and so on, until the last one, with $1 = 10^0$. In other words, the arrangement $(a_n, a_{n-1}, \ldots, a_1, a_0)$ is associated to the arrangement $(10^n, 10^{n-1}, \ldots, 10^1, 10^0)$ and the result of this association is the long sum at the final part of **[3]**.

The way we write the above arrangements reminds us vector notation. The long sum above, is called inner product (or dot product) of the two vectors. Explicitly:

$$(a_n, a_{n-1}, \ldots, a_1, a_0) \bullet (10^n, 10^{n-1}, \ldots, 10^1, 10^0) = a_n \times 10^n + a_{n-1} \times 10^{n-1} + , \ldots , + a_1 \times 10^1 + a_0 \times 10^0$$

Inner product, whose notation is "•", has a very important role in Hilbert Spaces.

This product would be first taught at elementary math students before traditional algorithms (inherited from medieval times). Inner product gives naturally, the exact value to digits in the numeral.

For example, the numeral 5384 (the string of digits) comes from the inner product:

$(5, 3, 8, 4) \bullet (10^3, 10^2, 10^1, 10^0) = (5, 3, 8, 4) \bullet (1000, 100, 10, 1) = 5\times1000 + 3\times100 + 8\times10 + 4 = 5000 + 300 + 80 + 4$.

As a result, 5384 is expressed in number words as: *Five Thousand Three Hundred Eighty Four.*

As we have noted, number words give us the sum of all the multiples of ten powers at the right hand of **[3]**. Then, we want now to know something more about these multiples.

Let us again use vector notation to represent the successive powers of ten, as an (n+1)-th dimensional vector:

$$(10^n, 10^{n-1}, \dots, 10^1, 10^0)$$

If we define vector addition as the vector formed by the sum of vector components in successive order, then we can express the above vector as

$$(10^n, 10^{n-1}, \dots, 10^1, 10^0) = (10^n, 0, \dots, 0, 0) + (0, 10^{n-1}, \dots, 0, 0) + \dots + (0, 0, \dots, 10^1, 0) + (0, 0, \dots, 0, 10^0)$$

The set of vectors at right of the equal sign is a rather special one and is called an orthogonal set. This set can be written:

$$\mathbf{S} = \{(10^n, 0, \dots, 0, 0), (0, 10^{n-1}, \dots, 0, 0), \dots, (0, 0, \dots, 10^1, 0), (0, 0, \dots, 0, 1)\} \qquad \textbf{[4]}$$

The last vector counts the units; next vector the tens, then hundreds and so on.

We say that each pair of two different vectors in the set is orthogonal since the inner product (dot product) is zero. This set permits us to know the value of each digit in the numeral. For instance, if we have the number 437, the orthogonal set for this case is $(10^2,0,0)$; (0,10,0), (0,0,1). Using the dot product of the vector representing 437 by each of them, we find:

$(4,3,7) \bullet (100,0,0) = 4\times100 + 3\times0 + 7\times0 = 400$. This means that the first digit has a value of 400.

$(4,3,7) \bullet (0,10,0) = 4\times0 + 3\times10 + 7\times0 = 30$. Then the second digit has a value of 30, and finally,

$(4,3,7) \bullet (0,0,1) = 4\times0 + 3\times0 + 7\times1 = 7$. This means that the last digit (units) has a value of 7.

As mentioned earlier, addition of two vectors is the vector that results from the addition of respective components orderly. Let us identify numbers with vectors. If one of them has fewer digits than the other, we place zeros as the first entries of the shorter. For example let us identify 325 and 1246 with the vectors (0,3,2,5) and (1,2,4,6). Then 325 + 1246 can be associated to the sum.

(0,3,2,5) + (1,2,4,6) = (0+1,3+2,2+4,5+6) = (1,5,6,11) = (1,5,6,10+1) = (1,5,6+1,1) = (1,5,7,1)

In the final steps we applied fundamental formula [2], which is the basis for handle the carries in traditional algorithms. From here we conclude that 325 + 1246 = 1571.

The above addition procedure is justified by the polynomial representation of the given numbers.

$(3x^2+2x+5) + (1x^3+2x^2+4x +6) = x^3+5x^2+6x +11 = x^3+5x^2+6x +10 +1 = x^3+5x^2+6x+x +1 = x^3+5x^2+7x+1$.

The last decimal polynomial corresponds to 1571.

Through last examples we have shown some logical rigor, step by step, in order to keep a sketch of a proof appropriated for making a program useful for coding. Some steps, however, can be omitted to save time and space. For example the sum 325 + 1246 can be reduced, either to:

(1,2,4,6) + (0,3,2,5) = (1,5,6,11) = (1,5,7,1).

Or to: 325 + 1246 = 1571, if we have enough ability to handle the carry away mentally in the last step above.

Changing numerals for vectors. The called "vectors" here are joined to our syntactic formula [2], in the sense that the n – th coordinate has ten times the value of the $(n-1)$ – th coordinate. Nevertheless, our vectors are different to the classical ones, although, operations and properties are essentially the same. Syntactic property [2], give us the possibility to transform naturally, a numeral into a vector.

The procedure is as follows: 1) we start with the numeral (string of digits) as a one-dimensional vector, then we left units apart to get a two dimensional vector, after that, we go to a tridimensional vector with tens and units at last. Continuing with this process we get:

$(a_n a_{n-1}\ldots a_1 a_0) = (a_n a_{n-1}\ldots a_1 ,a_0) = (a_n a_{n-1}\ldots ,a_1 ,a_0) =\ldots= (a_n a_{n-1},\ldots,a_1 ,a_0) = (a_n , a_{n-1},\ldots,a_1,a_0)$

All the process reduces to putting commas between digits. Let see how this process works with the numeral 3678:

(3678) = (367,8) = (36,7,8) = (3,6,7,8)

The important thing here is that the sequence of equalities above, gives us the possibility to represent 3678 in different forms as a sum of ten power multiples, namely:

$$3678 = 367\times10 + 8 = 36\times10^2 + 7\times10 + 8 = 3\times10^3 + 6\times10^2 + 7\times10 + 8.$$

In the same way, we can reverse the process going from right to left. This action gives us a reasonable meaning to carries' handling, as in the following example.

Example. Let us use vector notation to add 9993 + 7. First we put the two numerals in the same dimension, i. e. (9,9,9,3) +(0,0,0,7). Then we proceed with the addition:

(9,9,9,3) + (0,0,0,7) = (9+0,9+0,9+0,3+7) = (9,9,9,10) = (9,9,9+1,0) = (9,9,10,0) = (9,10,0,0) = (10,0,0,0).

Using the mentioned property above, the last vector is (1,0,0,0,0). We now conclude that 9993 + 7 = 10000.

Multiplication by scalar. The orthogonal set shown in [4] gives us the chance to introduce another type of product called multiplication by scalar, where a number multiplying a vector gives another vector where all the components or coordinates are multiplied by that number. The sign for this type of multiplication is "·". If k is a number and $(a_n, a_{n-1},\ldots,a_1,a_0)$ is a vector, we define this product as:

$$k\cdot(a_n, a_{n-1},\ldots,a_1,a_0) = (k\times a_n, k\times a_{n-1},\ldots, k\times a_1, k\times a_0). \qquad [\mathbf{5}]$$

In particular, when k is a digit in {0,1,2,3,4,5,6,7,8,9} and "×" is the sign for multiplication among digits, we find an easy way to multiply some numbers. Take, for example, k =7 and the number 348. The product of these numbers is:

7×348 = 7 · (3,4,8) = (21,28,56) = (21,28+5,6) = (21,33,6) = (21+3,3,6) = (24,3,6) = (2,4,3,6) =2436.

Maybe this procedure could looks weird, but, it is a reasoned way to explain why, 7×348 = 2436 without getting out of the context, going straight ahead and directly from input to output.

Through this paper we have mentioned three types of multiplication. I think we can try to teach elementary arithmetic starting with the inner product. After that, we can follow with product by scalar in order to perform simple multiplications by one digit. At last, we could try with the regular multiplication taught for centuries in the same way.

In another paper[2] we have explained how to use vectors and polynomials in the process of teaching multiplication algorithms and number factoring.

---

[2] Pareja-Heredia, D. *Beginning Abstract Algebra at Elementary Schoo1*, at: http://www.matematicasyfilosofiaenelaula.info/articulos/Beginning_Abstract_Algebra_at_Elementary_School.pdf

The same way as we introduce a factor into a vector, we can reverse this action putting a common factor out from the components of a vector. As an example of this case, we can put the powers of ten out of the parenthesis in the orthogonal set [**4**], and get something like this:

$10^n \cdot (1 , \dots , 0 , 0)$, $10^{n\text{-}1} \cdot (0 , 1, \dots , 0 , 0)$, … , $10^1 \cdot (0, 0, \dots ,1 , 0)$, and, $10^0 \cdot (0 , 0 , \dots , 0 ,1)$.

The above vectors with coordinates 1 in one place and zeros elsewhere, are called unity vectors and may be used to span all the $(n+1)$-th dimensional vector space[3]. This vector set is an example of an orthonormal set, which means, an orthogonal set where all vectors have length one.

Conclusion. Through the above lines we have presented some different forms to introduce numbers and algorithms for sum and product. We believe that with this methodology, elementary arithmetic can be connected to advance mathematics in a reasonable and systematic way.

First version, Armenia, July 2014

[3] The study of vector spaces was initiated by Hermann Grassmann in 1832. Linear algebra, a very important subject in advanced mathematics, is a product of the futuristic vision of Grassmann.