Annales Mathematicae et Informaticae
Manuscript
http://ami.ektf.hu

What does Mathematical Notation actually mean, and how can computers process it?*

James Davenport[a]

[a]University of Bath (U.K.)
J.H.Davenport@bath.ac.uk

January 26, 2015

Abstract

Mathematical Notation is generally though of as universal and constant. This is not as true as the layman thinks, and Notation is in fact an evolving, subject-specific, collection of sub-notations, where the same symbol can mean different things in different parts of the same sentence. This paper surveys the various ways computers process, and help humans to process, the varieties of notation.

Keywords: Mathematical Notation, MathML, OpenMath

MSC: 00A35, 68A30, 68C20

1. Notation: the perception and the reality

The outsiders' perception of mathematical notation is that it is unambiguous, unchanging, precise, and world-wide (or more so[1]). One need merely Google for the phrase "mathematically precise" to see many instances of this view. And indeed there is a lot of truth in this belief: the author has seen mathematicians be unable to speak to each other, having no human language in common, but be able to communicate by writing mathematics.

This is not just a popular belief: the computing discipline of "Formal Methods", which employs tens of thousands of people in industry, as well as many academics,

*Thanks to many people: typesetters, editors, OpenMath and MathML colleagues, TEXnicians.

[1]Witness various science-fiction stories where, e.g., Pythagoras' Theorem is used as a demonstration of intelligence.

Table 1: Cultural Notation Differences

Idea	Anglo-Saxon	French	German
half-open interval	$(0, 1]$	$]0, 1]$	varies
single-valued function	arctan	Arctan	arctan
multi-valued function	Arctan	arctan	Arctan
$\{0, 1, 2, \ldots\}$	$\mathbb{N}$	$\mathbb{N}$	$\mathbb{N} \cup \{0\}$
$\{1, 2, 3, \ldots\}$	$\mathbb{N} \setminus \{0\}$	$\mathbb{N} \setminus \{0\}$	$\mathbb{N}$

tries to reduce computer programming to mathematics/logic, and has substantial success in doing so.

However, mathematical notation is certainly not unchanging. Few except the scholars of the history of notation would recognise in

$$1cu.m.6ce.p.11co. \text{ equale } 6n^i \tag{1.1}$$

[Bab30, attributed to [Pac94]] the modern $x^3 - 6x^2 + 11x = 6$.

The + sign is less than 500 years old [Sti44] (this text also introduced $-$ and $\sqrt{}$).

The = sign is slightly younger [Rec57].

Recorde [Rec57] wrote $2\overline{a+b}$: $2(a+b)$ is later, but the parenthetic notation won because it is (much![2]) easier for manual typesetting.

Calculus had from the origin, and still has, two very different notations for ordinary differentiation: $\dot{x}$ versus $\frac{\mathrm{d}x}{\mathrm{d}t}$, which have given rise to u_{xxt} versus $\frac{\partial^3 u}{\partial^2 x \partial t}$.

Relativity introduced the summation convention [Ein16]: $\sum_{i=1}^{3} c_i x^i$ is abbreviated as $c_i x^i$ (but $c_\mu x^\mu$ is short for $\sum_{\mu=0}^{3} c_\mu x^\mu$, i.e. the range of summation depends on the alphabet from which the index is drawn).

Mathematical notation is also not quite as international as the layman believes: see Table 1. The examples there are drawn from relatively advanced mathematics, but the differences can be more basic — [Lop08, p. 2] lists five different forms of writing 9,435,671 found in Houston schools. These issues can spread to the description of algorithms such as division, as in Figure 1. Indeed MathML [Wor10] recognises 10 such formats, such as `stackedleftlinetop`: see `http://www.w3.org/Math/draft-spec/mathml.html#chapter3_presm.mlongdiv.ex`.

Mathematical notation is also subject-specific: while the mathematician writes i for $\sqrt{-1}$, the electrical engineer writes j, reserving i for current. A more chaotic

[2] The author, in the 1960s, used to typeset mathematics using "cold lead" technology: $2(a+b)$ involved selecting six characters from the cases of symbols, while $2\overline{a+b}$ would have involved cutting a raised piece of lead to form the overline *and* two "sleepers" — unraised pieces of lead — to sit either side of it to ensure that the overline was over the right characters. Furthermore, any change in the paragraph which moved the $\overline{a+b}$ horizontally would involve cutting new sleepers.

Figure 1: Division (from [Lop08, p. 7])

Format 1	Format 2	
$\begin{array}{r} 2 \\ 3\overline{)74} \\ 1 \end{array}$	$\begin{array}{r\|l} 74 & 3 \\ 1 & 2 \end{array}$	▪ Students typically begin to formulate and answer questions such as: How many times can 3 go into 7? Another way of asking is if we divide 70 into 3 sets, how many are in each set. ▪ Students write the 2 in the tens place, above the 7, on Format 1, but the 2 goes below the divisor when written in Format 2 style. Notice the placement of the quotient on each format. ▪ The next step is done mentally. Students multiply 3 x2 or (3 sets of 20) and then subtract. The only part that is written on paper is the remainder, 1 ten. Notice its location on both formats.
$\begin{array}{r} 2 \\ 3\overline{)74} \\ 14 \end{array}$	$\begin{array}{r\|l} 74 & 3 \\ 14 & 2 \end{array}$	▪ The 4 is brought down and students consider 14 next. ▪ Notice where the 14 is written on both formats.
$\begin{array}{r} 24 \\ 3\overline{)74} \\ 14 \end{array}$	$\begin{array}{r\|l} 74 & 3 \\ 14 & 24 \end{array}$	▪ Students now find that 3 will go into 14 three (3) times. They write 4 in the quotient's place.
$\begin{array}{r} 24 \\ 3\overline{)74} \\ 14 \\ 2 \end{array}$	$\begin{array}{r\|l} 74 & 3 \\ 14 & 24 \\ 2 & \end{array}$	▪ Students again mentally subtract 12 from 14 and write only the remainder: 2.

example of notational clash between subjects can be seen in [BDD+91], where the algebraist uses [...] to indicate a polynomial ring extension, and the biochemist uses [...] to indicate "concentration of". Hence computations were being conducted in $\mathbb{C}[[P][S][E]]$. The fact that "reaction scheme" notation uses + to indicate combination of reagents rather than mathematical addition is a further complication for the reader.

The mathematician also knows (without, possibly, having articulated it) that notation is area-specific within mathematics. For example $(2, 4)$ might be, depending on the area, any of:

Set Theory The ordered pair "first 2, then 4";

(Geometry) The point $x = 2, y = 4$;

(Vectors) The 2-vector of 2 and 4;

Calculus Open interval from 2 to 4;

Group Theory The transposition that swaps 2 and 4;

Number Theory The greatest common divisor of 2 and 4;

In general, these expressions, whilst written identically, are *spoken* differently by the mathematician: the written text "we draw a line from (2,4) to (3,5)" is spoken "we draw a line from the point (2,4) to the point (3,5)". This can apply even within a given sentence[3]: every group theorist would read

$$\text{Since } H_i \leq G \text{ for } i \leq n \tag{1.2}$$

as "Since H sub i is a subgroup of G for i less than or equal to n" without, probably, even noticing that the two instances of $\leq$ had been pronounced very differently. This issue is a major challenge for mathematics "text-to-speech" renderers.

2. Imperfections in notation

Mathematical notation has evolved over the centuries, and some innovations were, with hindsight, less than ideal.

2.1. "Landau" Notation

This notation, apparently actually due to Bachmann [Bac94], has two components. The first is not controversial: we use $O(f(n))$ to denote those functions that "grow no faster than $f(n)$" — formally (though rarely stated as such)

$$O(f(n)) = \{g(n) | \exists N, A : \forall n > N \ |g(n)| < Af(n)\}, \tag{2.1}$$

[3] I owe this example to Ieuan Evans of Bath.

and similarly with o, Ω, ω and Θ. The second component of this notation is the use of "=" with this, as in $\log_2 n = O(\log n)$. This is not the traditional use of the $=$ sign, as the relation is not symmetric: we can't write $O(\log n) = \log_2 n$, for example, and while we might stretch the notation to $O(n^2) = O(n^3)$, it is certainly not the case that $O(n^3) = O(n^2)$. Again, the spoken language gives a clue: the (English-speaking) mathematician would say "is" not "equals".

If we were honest with (2.1), we would write $O(\log n) \in \log_2 n$, but to the best of the author's knowledge, [Lev07] is the only textbook to be consistently honest in this area using $\in$, though [Het14] does refer to $O(f)$ as a set, and is careful to use neither $=$ nor $\in$. Being honest with (2.1) has another advantage: we can write $\Theta(f(n)) = O(f(n)) \cap \Omega(f(n))$, as [Het14] does.

2.2. Iterated Functions

No-one could quarrel with any of the following:

$\sin(x^2)$ square x, then apply sin

$(\sin x)^2$ apply sin to x, then square the result

$\sin(\sin(x))$ apply sin to x, then apply sin again

The problem comes with $\sin^2 x$, which is generally used to mean $(\sin x)^2$, whereas, if anything, it should mean $\sin(\sin(x))$, since this is the sense in which we write $\sin^{-1}(x)$ — apply the inverse operation of sin, not $1/\sin(x)$. The author is not the first to object to this notation: "[This] is by far the most objectionable of any" [Bab30]. The author has not encountered a definitive explanation of the origin of this notation, which was clearly common by the time of Babbage, but his experience of manual printing leads him to believe that it was economy of printing:

$$\sin^2 \frac{\theta+\phi}{2} \text{ versus } \left(\sin \frac{\theta+\phi}{2}\right)^2 \tag{2.2}$$

obviates searching for the very large brackets, and building up an exponent to a non-standard height.

2.3. Continued Fractions

The "correct" notation for continued fractions, as in

$$\pi = 3 + \cfrac{1}{7 + \cfrac{1}{15 + \cfrac{1}{1 + \cfrac{1}{292 + \ddots}}}} \tag{2.3}$$

is nearly always reduced to

$$\pi = 3 + \frac{1}{7+}\,\frac{1}{15+}\,\frac{1}{1+}\,\frac{1}{292+}\cdots, \tag{2.4}$$

which is much easier for (manual) typesetting[4], and uses less space — still a relevant consideration. Furthermore, if the individual terms of the continued fraction are complicated, as in

$$\alpha = a_0 + \cfrac{1}{a_1 + \cfrac{1}{a_2 + \cfrac{1}{a_3 + \cfrac{1}{a_4 + \ddots}}}}, \tag{2.5}$$

the alternative notation is probably more readable, at least when the reader is used to it.

2.4. Conclusion

We actually see that the same printed notation can mean very different mathematical objects, and that the same mathematical object can be displayed in many different styles. This has led to a conceptual split between the computerisation of the *presentation*, how the mathematics looks, and the computerisation of the *content* (or semantics), i.e. what the mathematics means. This is formalised in the MathML standard, which has different chapters, and even different basic tokens, for the two approaches.

3. Computer Displays of Mathematics

Let us first look at how computers mediate the presentation of mathematical formulae.

3.1. Display of Mathematics

We can distinguish various (overlapping) periods in the computer display of mathematics.

1. Images — generally GIF or JPEG formats, though others have been used, and SVG has become more desirable [SW14]. The fundamental problem with an image is that it is precisely an image — all machine-processable information has been lost. In HTML, it is possible to include an `ALT`ernative representation, and this might be the LaTeX source, which at least conveys some information to a text-to-speech renderer.

2. Computer processing — as photocomposition replaced "hot metal" technology in typesetting shops, so these photocomposers became computer-controlled. Various programs, notably `troff` [Oss76] and the associated mathematics preprocessor `eqn` [KC75], were developed to take advantage of this capability, and the author's PhD thesis was ported as [Dav81] to the IBM equivalent program — YFL [Gru79].

[4] As we (LaTeX) have written it (2.3) uses three sizes of digits, while (2.4) only uses one. Most "Hot metal" printers only had two available, so the result would not be as attractive as (2.3).

3. A major breakthrough came with Knuth's TEX [Knu84]. One fundamental development here over its predecessors was the principle of boxes with width, height and depth. The requirement to know the explicit depth of a box is fundamental, as in (2.3). This has become the *de facto* gold standard for mathematical typesetting.

4. The original HTML did not support mathematics, much to its designer's regret, and MathML–Presentation 1.0 [Wor97] soon appeared to fill this gap. However, the browsers of the period did not support the concept of 'depth' for boxes, and this can still be a problem today (Chrome's support for MathML has been intermittent, largely for this reason). A further challenge with many browsers[5] is the lack of fonts available.

5. MathJax [Mat11] has emerged as a pragmatic solution to the vagaries of browsers, and is discussed further in [SW14].

3.2. Line Breaking

All systems the author knows of make a fundamental distinction between "in-line" and "display" mathematics, and the user has to state which is required, e.g. `$...$` versus `$$...$$` in TEX. TEX and its derivatives provide so support for automatic breaking of lines if a displayed formula overflows the line width, and not much support for in-line formulae[6]. In the author's experience, a significant fraction of the effort in converting a paper from one format to another is in reflowing the equations, and maybe converting from display to in-line or *vice versa*.

However, the author of a web page has no control over the width within which it is displayed, and hence the browser must do *something* about linebreaking. This is also a problem for the various kinds of e-book readers, and partially accounts for the relative difficulty of handling mathematics, or technical text in general, on these devices. The MathML standard [Wor14, §3.1.7] provides a suggested algorithm, but, as it says there:

> This algorithm takes time proportional to the number of token elements times the number of lines.

This problem, with its blend of algorithmics and aesthetics, is at least as difficult as, but less-researched than, the problem of table layout, as discussed in [MMH13]

3.3. MathML-Presentation

While it is possible to regard MathML-Presentation as "LATEX with pointy brackets", this view in fact does it a disservice. While $f(x)$, written as `f(x)` in TEX, could be written as

[5] And other software: PowerPoint has often given users problems here.

[6] TEXnically speaking, the mathematics has been converted into a list of boxes by the time it is realised that line-breaking is needed, hence rules like "break at the outermost operator" no longer make sense.

```
<mrow> <mi> f </mi> <mo> ( </mo>  <mi> x </mi>  <mo> ) </mo> </mrow>
```

it would best be represented in MathML as

```
<mrow>
  <mi> f </mi>
  <mo> &ApplyFunction; </mo>
  <mrow>
    <mo> ( </mo>
    <mi> x </mi>
    <mo> ) </mo>
  </mrow>
</mrow>
```

In this representation, the function application, and precisely what the argument is, are clearly apparent. This matters for speech rendering — "f of x", as well as semantic analysis. However, it is still presentation, and cannot solve the sort of problem seen in (1.2).

4. Computer Representation of Mathematical Content

Originally, there were two different approaches to the description of mathematical content: OpenMath and MathML-Content. We describe each, and then the convergence process.

4.1. OpenMath

The OpenMath movement grew out of the Computer Algebra community's wish to move formulae between systems. An early document is [ADS96], which emphasises the importance of extensibility. Indeed, OpenMath is not so much an encoding as a framework for encoding, and the Standard [BCC+04] does not of itself specify how to transmit anything more complicated than integers. In fact it defines only a few basic concepts, listed here as their XML encodings.

`OMOBJ` The basic constructor, whose argument is an OpenMath objects. This exists so that OpenMath can be embedded in other documents, as formulae are in text. It's opposite is `OMFOREIGN`, indicating that we have some non-OpenMath constructs (such as Presentation MathML) embedded in an OpenMath object.

`OMS` This indicates an "OpenMath Symbol", an object to which the OpenMath process assigns a definite meaning. The arguments are the **name** of the symbol, e.g. `sin`, and the location of the "Content Dictionary" in which that definition can be found. This location can be either a simple name (`transc1`

would indicate the standard Content Dictionary for basic transcendental functions) or a complete URL.

`OMA` This indicates an "OpenMath Application", where the first argument is to be considered an operator applied to the remaining arguments.

`OMBIND` This indicates an "OpenMath Binding", where the first argument is some operator to bind the variables specified in the second argument in the use of the third argument. A typical first argument would be `<OMS name="forall" cd="quant1"/>` to indicate $\forall$.

`OME` This indicates an "OpenMath Error Object" (such as "divide by zero"): the first argument is the 'name' of the error, as an `OMS`, and the rest are additional arguments depending on the error.

`OMATTR` This indicates an "OpenMath Attribution": the first argument has various attributes, such as `color` being `red`.

`OMR` This indicates an "OpenMath Reference" and allows us to build directed acyclic[7] graphs, rather than just trees.

Basic objects are encoded by any of `OMV` (variables), `OMI` (integers), `OMB` (byte arrays), `OMSTR` (Unicode strings) or `OMF` (IEEE floating-point numbers).

4.2. MathML-Content

This was introduced at the start of the MathML process, with a view to being "an explicit encoding of the underlying mathematical meaning of an expression, rather than any particular rendering for the expression" [Wor14]. Equally, as have we have seen, renderings can be ambiguous, and one aim of MathML-Content is to remove this ambiguity. Consider $(F + G)x$: this could be either multiplication or function application: see Figure 2.

The original aim in MathML (version 1) was to handle "school" mathematics, otherwise "K–12", or Kindergarten to 12th-grade. However, this became a moving target, as constructs like `<div>` were introduced.

4.3. Convergence

The reader will have noticed that there is a strong similarity between OpenMath and MathML-Content, with `<apply>` corresponding to `<OMA>`, and `<ci>` constructs corresponding to `<OMV name=` constructs. The difference is that `<plus/>` is part of the MathML specification, whereas `<OMS name="plus" cd="arith1"/>` is just a symbol in an OpenMath content dictionary. This difference is also the source of the greater expressivity of OpenMath: MathML needed to charge to accommodate `<div/>`, whereas OpenMath just added the `veccalc1` content dictionary.

[7] There is an explicit ban on cycles in the OpenMath standard.

Figure 2: Alternative MathML-Content for $(F + G)x$

```
<apply><times/>          <apply>
  <apply><plus/>           <apply><plus/>
    <ci>F</ci>               <ci>F</ci>
    <ci>G</ci>               <ci>G</ci>
  </apply>                 </apply>
  <ci>x</ci>               <ci>x</ci>
</apply>                 </apply>
```

We note that there is no need for brackets, as `<apply>...</apply>` groups, and the meaning is explicit: in the first we have an application of `<times/>` while in the second we are applying $F + G$.

MathML version 2 therefore added the ability to use OpenMath symbols, thus buying into the expressivity of OpenMath. In MathML version 3, the authors went further, and defined MathML-Content in terms of OpenMath as follows.

> [In §4.2] a core collection of elements comprising Strict Content Markup are described. Strict Content Markup is sufficient to encode general expression trees in a semantically rigorous way. It is in one-to-one correspondence with OpenMath element set. OpenMath is a standard for representing formal mathematical objects and semantics through the use of extensible Content Dictionaries. [Wor14, §4.1.1].

`<plus/>` is then defined to be a shorthand for `<OMS name="plus" cd="arith1"/>`, etc.

5. Conclusion

In terms of reproducing via computers the intricate two-dimensional layouts of mathematical notation, created (at considerable expense) by cold-metal printers, the TEX engine [Knu84] has no equal. However, all it does is express how to lay out the symbols, and says nothing about their meaning. The invisible operator after `)` in the LATEX `(F+G)x` could be either function application or multiplication.

Although it is possible to write MathML-presentation that conveys no more information than the LATEX, well-written MathML-presentation can convey far more, as the invisible operator should be either `&ApplyFunction;` or `&InvisibleTimes;`.

However, presentation MathML can only go so far in encoding meaning, and is still unable to resolve the two uses of $\leq$ in (1.2) for example. For this, we need a representation of the semantics ,either OpenMath or MathML-Content. Fortunately, the two have converged so much that they are essentially isomorphic structures, and we can look forward to greater convergence in the future.

References

[ADS96] J.A. Abbott, A. Díaz, and R.S. Sutor. OpenMath: A Protocol for the Exchange of Mathematical Information. *SIGSAM Bulletin 1*, 30:21–24, 1996.

[Bab30] C. Babbage. Article "Notation". *Edinburgh Encyclopaedia*, 15:394–399, 1830.

[Bac94] P. Bachmann. *Die analytische Zahlentheorie*. Teubner, 1894.

[BCC+04] S. Buswell, O. Caprotti, D.P. Carlisle, M.C. Dewar, M. Gaëtano, and M. Kohlhase. The OpenMath Standard 2.0. `http://www.openmath.org`, 2004.

[BDD+91] J.P. Bennett, J.H. Davenport, M.C. Dewar, D.L. Fisher, M. Grinfeld, and H.M. Sauro. Computer algebra approaches to enzyme kinetics. In Gérard Jacob and Françoise Lamnabhi-Lagarrigue, editors, *Algebraic Computing in Control*, volume 165 of *Lecture Notes in Control and Information Sciences*, pages 23–30. Springer Berlin Heidelberg, 1991.

[Dav81] J.H. Davenport. *On the Integration of Algebraic Functions*, volume 102 of *Springer Lecture Notes in Computer Science*. Springer Berlin Heidelberg New York (Russian ed. MIR Moscow 1985), 1981.

[Ein16] A. Einstein. Die Grundlage der allgemeinen Relativitaetstheorie (The Foundation of the General Theory of Relativity). *Annalen der Physik Fourth Ser.*, 49:284–339, 1916.

[Gru79] A.M. Gruhn. The Yorktown Formatting Language: User Guide. Technical Report RC 6994 IBM Research, 1979.

[Het14] M. Hetland. *Python algorithms: Mastering Basic Algorithms in the Python Language (2nd ed.)*. Apress, 2014.

[KC75] B.W. Kernighan and L.L. Cherry. A System for Typesetting Mathematics. *Comm. ACM*, 18:151–157, 1975.

[Knu84] D.E. Knuth. *The TEXbook: Computers and Typesetting Vol. A*. Addison–Wesley, 1984.

[Lev07] A. Levitin. *Introduction to the design and analysis of algorithms*. Pearson Addison–Wesley, 2007.

[Lop08] N.R. Lopez. Mathematical Notation Comparisons between U.S. and Latin American Countries. `https://sites.google.com/site/algorithmcollectionproject/mathematical-notation-comparisons-between-u-s-and-latin-american-countries`, 2008.

[Mat11] MathJax Consortium. MathJax: Beautiful math in all browsers. `http://www.mathjax.org/`, 2011.

[MMH13] Kim Marriott, Peter Moulder, and Nathan Hurst. Html automatic table layout. *ACM Trans. Web*, 7(1):4:1–4:27, March 2013.

[Oss76] J.F. Ossanna. Nroff/Troff User's Manual. Technical Report 54 Bell Labs, 1976.

[Pac94] Luca Pacioli. *Summa de arithmetica, geometria, proportioni et proportionalita*. Venice, 1494.

[Rec57] R. Recorde. *The Whetstone of Witte*. J. Kyngstone, London, 1557.

[Sti44] Stifelius [Michael Stifel]. *Arithmetica Integra*. Iohan Petreius, Norimberg, 1544.

[SW14] M. Schubotz and G. Wicke. Mathoid: Robust, Scalable, Fast and Accessible Math Rendering for Wikipedia. http://arxiv.org/abs/1404.6179, 2014.

[Wor97] World-Wide Web Consortium. Mathematical Markup Language: First Public Draft. http://www.w3.org/TR/WD-math-970515/, 1997.

[Wor10] World-Wide Web Consortium. Mathematical Markup Language (MathML) Version 3.0: W3C Recommendation 21 October 2010. http://www.w3.org/TR/2010/REC-MathML3-20101021/, 2010.

[Wor14] World-Wide Web Consortium. Mathematical Markup Language (MathML) Version 3.0: second edition. http://www.w3.org/TR/2014/REC-MathML3-20140410/, 2014.