Semantic Ambiguities by Subject Area

# General Discussion

Math notation is often reused in different subject areas. Most of the time, the meaning are related. For example “+” might mean addition of numbers or addition of matrices. However, there are times when a notation has a different meaning based on the subject area. For example, “(*a*, *b*)” might mean a point in the plane or an open interval. This discussion focuses on resolving the meaning in these later cases. In most cases, the subject area removes the ambiguity and resolves the meaning. However, that is not always the case and examples where that is true are noted below.

## Adding semantics to presentation MathML

The current working idea for resolving ambiguities in presentation MathML is to add attributes to specify a subject area and another to specify a semantics. For the sake of discussion, I will call these attributes “subject” and “mathrole”. “subject” is an attribute of the math tag and “mathrole” can go on any MathML element. Some examples:



<math **subject=”geometry”** xmlns="<http://www.w3.org/1998/Math/MathML>">

<mover>

<mrow><mi>A</mi><mi>B</mi></mrow>

<mo>&#xAF;</mo>

</mover>

<mo>&#x2225;</mo>

<mover>

<mrow><mi>C</mi><mi>D</mi></mrow>

<mo>&#xAF;</mo>

</mover>

</math>

 (should the mathrole should be on the <msup> instead?)

<math xmlns="<http://www.w3.org/1998/Math/MathML>">

<mi>M</mi>

<mo>&#xB7;</mo>

<msup>

<mi>M</mi>

<mi mathrole=”transpose”>T</mi>

</msup>

</math>



<math xmlns="<http://www.w3.org/1998/Math/MathML>">

<mo>&#xA0;</mo>

<mrow mathrole=”binomial-coef”>

<mo>(</mo>

<mfrac linethickness=”0”>

<mi>m</mi>

<mi>n</mi>

</mfrac>

<mo>)</mo>

</mrow>

</math>

*mathrole* should be able to appear anywhere, not just on leaf nodes. This is illustrated in the binomial coefficient example; there is not a natural place to put the role on a leaf.

## Multiple notations for a single semantics

It is not unusual that multiple notations are used for the same meaning. For example:

division can be represented with “/” or using a horizontal line for a more vertical layout

inverse trig functions can be given as (for example) asin(*x*) or sin-1(*x*) or sin-1*x*.

Multiplication can be represented with horizontal juxtaposition, using “·”, or “×” among other options. Because *mathrole* is used with presentation MathML, mathrole is not is not relevant for issues that might arise from multiple ways of representing the same semantics.

## Internationalization

Some differences in notation are based on language/custom in that country. E.g., “tg” is sometimes used instead of “tan”. Other times, notations are different. In French, “]*a*,*b*[“ is used for an open interval; in English, “(*a*,*b*)” is used. MathML support for long division includes 10 different layouts.

The focus here is on English language use. However, for a full solution, how notations differ in other languages should be added to this discussion.

### Units of Measure

A source of ambiguity can be units. These can be confused with variables, particularly when the unit is only a single letter. For example, ‘l’ is used for liters but can also be a variable; ‘ml’ is probably almost always milliliters. Unit ambiguity cuts across subject areas, so units are independent of the subject area. There is a [MathML note](https://www.w3.org/TR/mathml-units/) that suggests using class=”MathML-Unit” to indicate a unit. With the addition of a mathrole attribute, using mathrole for units should be suggested instead.

### Canonical MathML

The MathML 3 spec is very lenient as to what is valid MathML markup. Other than requiring it to be well formed XML (relaxed as part of HTML 5) and a few requirements on the number of children of elements such as <mfrac>, the spec only makes some suggestions. This leads to many complications with trying to figure out semantics because the same piece of semantics may be presented visually (mostly) the same but have very different markup. For example, here are some ways that “-2 + 3” might be represented:

|  |  |  |
| --- | --- | --- |
| <mrow>  <mrow>  <mo>-</mo>  <mn>2</mn>  </mrow>  <mo>+</mo>  <mn>3</mn>  </mrow> | <mrow>  <mo>-</mo>  <mn>2</mn>  <mo>+</mo>  <mn>3</mn>  </mrow> | <mrow>  <mn>-2</mn>  <mo>+</mo>  <mn>3</mn>  </mrow> |

The first column is the prefered method because the mrow structure reflects the syntax tree of the underlying math. The spec specifically says that negative numbers should not include the negative sign, but it is not called out as invalid and some software generates this. None of these examples benefit from the use of mathrole, but the work needed for interpretation is probably three times what it should be due to the multiple representations. Other examples are multiple characters that might be used for the same semantics (e.g., “-” and “\_” and others might be used in the accent position for an overbar; ‘ or the Unicode prime character might be used for first or higher order derivatives.

Over the years, there have been several papers written about the need for a canonical MathML representation. I believe all differed on some details and often lacked some areas that would benefit from being canonicalized.

HTML faced a similar issue in its early days: “bad” HTML would be corrected by different browsers differently. That led to authors looking at a page in one browser thinking everything was fine, but readers using a different browser seeing a page that looked very poor. Typically, the reader and the author would blame the browser where it looked bad. HTML defines the fix up rules (which are quite complicated) and now browsers agree on how all pages are interpreted.

One potential solution is to do the same for MathML -- define a canonical MathML that all renderers need to produce (in their DOM). That would likely involve changing the HTML parser and is pretty much a non-starter. Another possibility is to develop a shared library that does the same thing -- any MathML passed to a semantic interpreter would work of this canonical MathML. A big problem with either solution is that the syntax tree depends on the semantic interpretation. Here are some examples:

* 3 *f*(*x*+*y*)2: the syntax tree is very different depending on whether this is multiplication or function call
* *h* -2 cos *y* -- might look like an infix minus but consider the similar case “sin -2 cos *y*”. Contrast this to “(sin-cos)(*x*)”.
* |x|y|z| -- can be '|x| y |z|' or '|x |y| z|', or even | (x|y)|z |'.
* At the token level, it is not uncommon to find numbers like 2,000 being broken up into pieces (<mrow> <mn>2</mn> <mo>,</mo> <mn>000</mn> </mrow>) because they are interpreted incorrectly by editor. This is particularly hard to deal with when inside of parens as in “(1,234)” [point?, big number? Open interval? ...\]

In addition to different ways of generating MathML, there a many generators that produce incorrect MathML. For example, ∠ABC might use &InvisibleTimes; between the letters. Other generators I’ve seen use incorrect leaf tagging (e.g., using <mtext> for identifiers with more than one letter. Although it is tempting to write these off as bad MathML, a good semantic renderer needs to be able to deal with common errors, and any error correction should be uniform across semantic interpreters for the reasons mentioned above.

### When to use *mathrole*?

*mathrole* is meant to be used to resolve ambiguity. But when is ambiguity ambiguous 😊? For example, in the following, we have a subscript: log2x. Subscripts can mean many things, but in context, there is no question in this case the subscript is the base of the logarithm. Similarly, the superscript in sin-1*x* is not a power, but when used with a trig function, everyone knows it means inverse. Limits are another example. They often involve the use of right arrow, and right arrows have many meanings. In the context of a limit notation lim*x*→0 sin *x* / *x*, the arrow is unambiguous.

A question that needs to be answered, however, is how much context should be considered to resolve ambiguity? In the equation H + Cl = HCl, it is clear that the “H” is hydrogen, but that is only understood because of the other terms in the equation. Here’s an example that is slightly more ambiguous: 3m/s. Although m and s could be variables, when together like this, it is likely they stand for meters and seconds, respectively.

*This is an area that requires more discussion*.

# Subject Areas

To be done: See what SRE does

## General

Probably the most common place for ambiguity is horizontal juxtaposition, which typically means either function call or multiplication. Because there is no character present, using mathrole would require putting it on an enclosing <mrow>. However, there is a better solution. If someone is going to the trouble of marking up function call or multiplication, they can add the Unicode characters U+2061 or U+2062, which are function application and invisible times.

One place where horizontal juxtaposition means something else is in subscripts such as M12, which might be a double index (the first row, second column of the matrix M). Unicode solves this problem also with the InvisibleComma glyph (2063).

A final place where an invisble Unicode char is useful for mixed fraction. For this, the InvisiblePlus glyph (2064) should be used.

## Elementary Math

Roman numerals is not marked up inside of an <mn>

## Pre-algebra, algebra, algebra II

Another way of writing a function -- f:→xy -- “:” and “→” only unambiguous in full context

"|" -- “such that” when used inside of {}s

double struck capital Z, N, R, Q, or C -- always interpret as Integers, etc? What about if there is a superscript “+” or “-” or a pos Integer?

{... (x,y) …} -- set of points?

| … | -- absolute value unless “...” is an mtable (then determinant); could also be cardinality of a set

From davidf: ordered pair vs cartesian coordinate

## Geometry

Overscripts: segments(\_, 005F), rays(→, 2192), lines(↔, 2194), arc (⌢, 2322) [AB and ABC]

Operators: arc

? degrees (°, 00B0)

∠ angle (2220)

m∠ measure of angle

△ triangle (25B3), also △ABC

? ▱ parallelogram (25B1)

∼ similar [triangles] (233c)

≅ congruent (2245) -- also not congruent, but no unicode for that [menclose or combing slash]

⊥ perpendicular (22A5) also not perpendicular , but no unicode for that [menclose or combing slash]

### Probability

P(A|B) or Pb(A)conditional probability

## Statistics

Overbar: mean

μ -- mean

σ -- standard deviation

N(...) -- Normal distribution

***χ*2**(k) or ***χk*2**-- Chi-squared distribution (Χ, 0397)

F(a,b) or F*α*(a,b) or Fa,b sometimes with ‘f’

H0 null hypothesis

H1  or Ha  alternative hypothesis

E[x] expected value

Var(x) or var(x) -- variance

Cov(X, Y) or cov(X, Y) -- covariance

P(A ∩ B) -- intersection, but spoken “and” in this context

P(A ∪ B) -- union, but spoken “or” in this context

P(A') -- probability of ‘not’ A

P(A) -- probability

## Pre-calculus/Trigonometry

## Calculus

Overscripts: vectors (⇀, 21C0)

Operators:

Gradiant (∇ 2208)

Cross product (⨯ 2A2F, maybe also ×, 00DF)

Dot product (⋅, 22C5)

Notations:

Leibniz: d/dx, dn/dxn, also dy/dx, dny/dxn [d interpreted differently]

Also with partial (∂, 2202)

Lagrange: f’ f’’, f’’’ and f(n)) [can be single quote or prime chars: 0x2032, 0x2033, 0x2034, 0x2057

Newton: dots above -- probably only one, two, or three dots

Euler: Dn f

Also with partial (∂, 2202)

Partial derivative (images from wikipedia):

 

Limits:

With x → a+ or x↗a (2197)

With x → a- or x↘a (2198)

| with subscript(s) -- evaluated at/between

## Linear Algebra

[ mtable ] or ( mtable ) -- matrix or vector (vector ambigious -- matters for speech)

Also used around an <mi> like “M” -- ambigious with absolute value in that case

| mtable | -- determinant

"‖" mtable "‖" -- norm or magnitude -- can also be around a variable

## Combinatorics

P with pre and post subscripts -- permutation

C with pre and post subscripts -- “choose”

(..) binonomial

From davidf: (seem more like graph theory)

*φ*(*n*) -- Euler's totient function

χ(G) -- chromatic number

ω(*G*) -- clique number

d(u,v) -- distance in a graph

∼ -- equivalence relation

Pn  -- path on *n* vertices

Cn -- cycle on *n* vertices

Kn -- complete graph on *n* vertices

**K**m,n -- complete bipartite graph

## Chemistry

Chemical Elements -- might just say letters (without capital indication) “N a” or say the name “Sodium”

Maybe not exceptional, but some chemical formulas have common names that don’t follow standard chemical naming. E.g, H20 is water, not “dihydrogen monoxide”. Ammonia is another example.

Polyatomic Ions also have special speaking rules. Examples

SO3 -> sulfite

PO3 -> phosphite

Don’t speak parens in some cases. Example in compounds like

Al(OH)3

State symbols such as HCl(aq)

Concentrations ([...]): Examples:

Example

[Cu] -> the concentration of copper

[Cu][He] -> the concentration of copper \*times\* the concentration of helium

Equilibrium Constant: Keq or Kc

Ions: Cu2+ is spoken as “copper 2 plus”, no superscript is read

Subscripts: not read as a subscript -- H20

Arrows: → (2192), ⇄ (21C4), ⇌ (21CC)

Could be “yields”, “decays to”, or “is in equilibrium with”

Isotopes:  (Uranium with mass number 235 and atomic number 92)  (Uranium 235)

Decay particles: 

# Acknowledgement

Some of the examples come from the [extensive work that David Farmer has been doing](https://docs.google.com/document/d/1cZnff5_fi_ucNyZ1ex2msmJLE55FAZD-QInkLYe8xiE/edit) looking for ambiguities in math textbooks.