Requirements for MathML markup in EPUB files for NLB

Per Sennels, NLB
Gaute Rønningen, NLB

December 2, 2019

Version history:

- 20181220 / 1.0: First version
- 20190130 / 1.1: Second version: Many new topics added, and there are smaller edits of the original text.
- 20190926 / 1.2: Third version: Minor changes and additions.
- 20191129 / 1.3: Fourth version (this version): Minor changes.

Changes and additions printed in dark red, and with date of change in margin.

Contents

1 The purpose of MathML in NLB's EPUB files **3**

2 The fundamental requirements **3**

3 Additional requirements and examples **7**

3.1 Use of invisible operators 7

3.2 Markup of less than, less than or equal, greater than and greater than or equal 7

3.3 Markup of parenthesis 8

3.4 Markup of the absolute value 10

3.5 Markup of number of degrees 11

3.6 Markup of square roots and higher-order roots 15

3.7 Markup of exponentiation 16

3.8 Markup of vectors 16

3.9 Markup of fractions 18

3.10 Lower indices 19

3.11 Markup of functions with one argument 21

3.12 Markup of functions with two or more arguments 22

3.13 Markup of named functions 24

3.14 Markup of limits and the derivative 25

3.14.1 The *Lagrange notation* 28

3.14.2 The *Leibniz notation* 31

3.15 Markup of the integral 35

3.16 Markup of chemistry 39

3.17 Markup of physics 40

1 The purpose of MathML in NLB's EPUB files

NLB will use MathML in the EPUB files to present mathematical expressions in a variety of ways in the distributed versions of the content:

- In a talking book version, we may synchronize a narrated version of the expression to the standard mathematical notation, the latter being generated automatically from MathML by the browser displaying the math as well as the text content of the talking book.
- In an e-book version we may present a MathML expression in the normal way; as a standard mathematical expression, using standard mathematical notation. In addition, we may present an automatically generated textual representation of the mathematical expression, as an alternative or a supplement to those who prefer to use local synthetic speech or refreshable Braille to digest the content.
- In a TTS based talking book version, we may automatically generate a string that represents a verbal interpretation of the MathML. This string can then be used as a basis for TTS generation of an audio segment that represents the mathematical expression.
- The MathML version of a mathematical expression can be used as a basis for AsciiMath and printed Braille. Once again, the MathML may be converted to either AsciiMath or text, and this version may then be refined and converted to Braille.

Except for the first bullet point, which relies on the math knowledge of a human narrator, production of the distributed version involve some kind of automatic transformation of the MathML markup into some other kind of textual representation, typically a pure text string containing a verbal representation of the math in question.

2 The fundamental requirements

- All mathematical expressions – both inline and block – shall be marked up, along with the general EPUB markup, using *MathML Presentation Markup*, as specified in "Mathematical Markup Language (MathML) Version 3.0 2nd Edition"[1].
- Whenever applicable, the requirements in this document must be respected.

  For mathematical expressions that are not covered in this document, the personnel involved in the markup process is encouraged to do MathML markup based on a good understanding of mathematics, combined with a solid knowledge of the set of MathML elements and attributes.
- Unless MathML markup of a certain type of mathematical expression is specified in this document, the markup should be based on a good understanding of mathematics, combined with a solid knowledge of the set of MathML elements and attributes.
- Use the numeric XML annotation of an entity, for example: instead of `&ApplyFunction;` or `&af;`, use `&#8289;`

[1] See `https://www.w3.org/TR/MathML3/`

- All MathML markup must be annotated with an AsciiMath expression that represents the mathematical expression printed in the book. Thus, the complete markup of any given mathematical expression must be coded according to the following scheme:

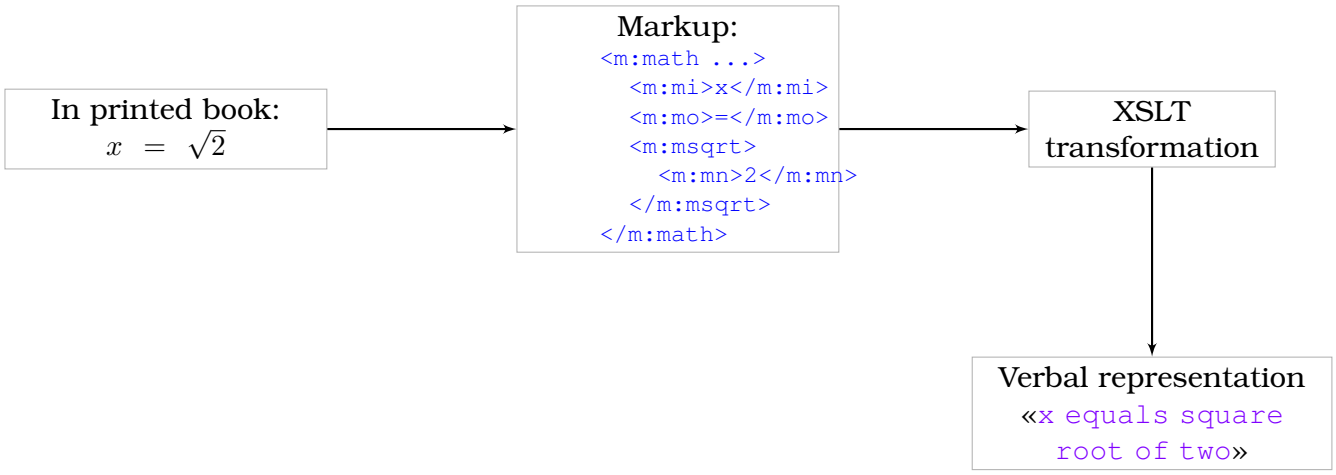
```
<m:math xmlns:m="http://www.w3.org/1998/Math/MathML"
      alttext="[AsciiMath markup]"
      altimg="[Image path]"
      display="[value]">
   <m:semantics>
      <m:mrow>
         [MathML markup]
      </m:mrow>
   </m:semantics>
</m:math>
```

  The value of the `display` attribute shall always be either `block` or `inline`, depending on the placement of the expression in the printed book. The `[MathML markup]` is, of course, the MathML markup that represents the mathematical expression in question, and `[AsciiMath markup]` is the AciiMath version of the same expression.

  Observe that the MathML markup will always be a child of an `mrow` element, which again is a child of an `semantics` element, which again is a child of an `math` element, which constitute the frame around the complete MathML markup.

  Please consult `https://www.w3.org/TR/MathML3/chapter5.html#mixing.semantic.annotations` for further information about annotation of MathML markup.

Please note that, in all of the examples to follow, the markup discussed is the one represented by the `[MathML markup]` part. The outer `math`, `semantics` and `mrow` framework is omitted, to put focus on the matter in question. These elements are of course always required as a container for each of the mathematical expressions in the EPUB file.



The quality of the generated text string relies heavily on the quality of the MathML markup, and care must be taken to create MathML markup that . . .

- . . . correctly represents the printed mathematical expressions;
- . . . is compliant with the W3C recommendations for use of MathML markup;
- . . . respects the NLB specific requirements stated in this document;
- . . . ensures that the mathematical information that can be detected from the markup, is as unambiguous as possible.

The last point is very important, as the interpretation of a mathematical expression often relies on the context the expression is placed in, and also on the interpreter's (e.g. the student's) understanding of that context.

As an example, we can investigate the following expression:

$$a(t+\varphi)$$

One interpretation of this expression is that it represents a variable, a, that is to be multiplied by the sum of two other variables, $t+\varphi$.

A completely different interpretation is that the expression represents a function, a, with one argument, namely the sum of t and φ.

It is extremely important that the interpretation is clearly indicated in the MathML markup. If the expression represents a multiplication operation, this should be clarified by use of the MathML operator `<m:mo>&#8290;</m:mo>`. And similar, the representation of a function must be clarified by using `<m:mo>&#8289;</m:mo>`.

So, even though this:

```
<m:math xmlns="http://www.w3.org/1998/Math/MathML" display="block">
   <m:mi>a</m:mi>
   <m:mo>(</m:mo>
   <m:mi>t</m:mi>
   <m:mo>+</m:mo>
   <m:mi>&phi;</m:mi>
   <m:mo>)</m:mo>
</m:math>
```

in many cases would be perfectly good markup of the expression above, NLB require the markup to be either

```
<m:math xmlns="http://www.w3.org/1998/Math/MathML" display="block">
        <m:mi>a</m:mi>
        <m:mo>&#8290;</m:mo>
        <m:mfenced open="(" close=")">
                <m:mrow>
                        <m:mi>t</m:mi>
                        <m:mo>+</m:mo>
                        <m:mi>&phi;</m:mi>
                </m:mrow>
        </m:mfenced>
</m:math>
```

for the "multiplication interpretation", or

```
1  <m:math xmlns="http://www.w3.org/1998/Math/MathML" display="block">
2          <m:mi>a</m:mi>
3          <m:mo>&#8289;</m:mo>
4          <m:mfenced open="(" close=")">
5                  <m:mrow>
6                          <m:mi>t</m:mi>
7                          <m:mo>+</m:mo>
8                          <m:mi>&phi;</m:mi>
9                  </m:mrow>
10         </m:mfenced>
11 </m:math>
```

for the ”function interpretation”. Note that the only difference is the choice of entity in line 3.

It should be quite clear by now that, to ensure the expected quality of the MathML markup, personnel with solid mathematical skills, as well as the ability to focus on important markup details, must be assigned to this kind of work.

If the expression above stands completely alone, it is not possible to decide the correct representation. However, if there is an overlaying context, perhaps if the expression above is part of a larger mathematical expression, the representation should be clear enough.

This equation

$$a(t+\varphi) = a \cdot t + a \cdot \varphi$$

clearly indicates that we are talking about multiplication, while

$$a(t+\varphi) = \frac{\partial^2 x(t+\varphi)}{\partial t^2}$$

indicates that we are talking about functions, perhaps related to acceleration and position.

This document aims to specify the exact markup we want, when there seems to be different ways to use MathML to represent the mathematical expression. If you cannot find a description on a specific notation in this document, please refer to the Mathematical Markup Language (MathML) Version 3.0 2nd Edition of 10th of April, 2014 `https://www.w3.org/TR/MathML3/`.

Note that this is a work in progress; the scope of mathematical topics to be covered will certainly be widened in the future. You should also expect changes in established requirements as we gain experience with the automatic transformation of MathML to alternative formats.

3 Additional requirements and examples

3.1 Use of invisible operators

If there is any risk of ambiguity, the following operators **must** be used as entities in the markup:

Entity	Numeric	Comment
&ApplyFunction;	&#8289;	Function application
&InvisibleTimes;	&#8290;	Invisible multiplcation
&InvisibleComma;	&#8291;	Invisible separator
	&#8292;	Invisible addition

This means that, even for expressions such as $(x+y)\,(x-y)$ or $2\,\sin\alpha$, where it is quite clear from the context that multiplication is involved, we *require* that these multiplications are added to the markup. Thus, the first of these two expressions must be marked up as

```
1   <m:mfenced open="(" close=")">
2           <m:mrow>
3                   <m:mi>x</m:mi>
4                   <m:mo>+</m:mo>
5                   <m:mi>y</m:mi>
6           </m:mrow>
7   </m:mfenced>
8   <m:mo>&#8290;</m:mo>
9   <m:mfenced open="(" close=")">
10          <m:mrow>
11                  <m:mi>x</m:mi>
12                  <m:mo>-</m:mo>
13                  <m:mi>y</m:mi>
14          </m:mrow>
15  </m:mfenced>
```

while the second expression must be represented with the following markup:

```
1   <m:mn>2</m:mn>
2   <m:mo>&#8290;</m:mo>
3   <m:mrow>
4           <m:mi>sin</m:mi>
5           <m:mo>&#8289;</m:mo>
6           <m:mi>&alpha;</m:mi>
7   </m:mrow>
```

Note the use of `&#8290;` in line 8 and line 2 respectively in these markup snippets.

Multiple examples of use of invisible operators are given in the following sections.

3.2 Markup of less than, less than or equal, greater than and greater than or equal

HTML entities would generally be able to be used for this, but to remove the risk of ambiguity, we will need them to be coded in the following manner:

Short	Numeric	Comment
&lt;	&#60;	Less than
&gt;	&#62;	Greater than
	&#8804;	Less than or equal
	&#8805;	Greater than or equal

$9 < 10$

```
<m:mrow>
   <m:mn>9</m:mn>
   <m:mo>&#60;</m:mo>
   <m:mi>10</m:mi>
</m:mrow>
```

$10 > 9$

```
<m:mrow>
   <m:mn>10</m:mn>
   <m:mo>&#62;</m:mo>
   <m:mi>9</m:mi>
</m:mrow>
```

$9 \leq 9$

```
<m:mrow>
   <m:mn>9</m:mn>
   <m:mo>&#8804;</m:mo>
   <m:mi>9</m:mi>
</m:mrow>
```

$9 \geq 9$

```
<m:mrow>
   <m:mn>9</m:mn>
   <m:mo>&#8805;</m:mo>
   <m:mi>9</m:mi>
</m:mrow>
```

3.3 Markup of parenthesis

Parenthesis shall not be marked up using `<m:mo>(</m:mo>` and `<m:mo>)</m:mo>`. Rather the MathML element `mfenced` must be used. The `mfenced` should contain attribute defining what type of parenthesis it, as follows:

```
<m:mfenced open="(" close=")">
   <m:mi>x</m:mi>
</m:mfenced>
```

This is done to clarify what parenthesis we are dealing with.

There are other parentesis to be aware of:

Square bracket

```
<m:mfenced open="[" close="]">
   <m:mi>x</m:mi>
</m:mfenced>
```

This is trivial when there is only one element inside the parenthesis, such as $g(x)$, which could be marked up as

```
<m:mrow>
   <m:mi>g</m:mi>
   <m:mo>&#8289;</m:mo>
   <m:mfenced open="(" close=")">
      <m:mi>x</m:mi>
   </m:mfenced>
</m:morw>
```

However, when the content of the parenthesis consists of multiple parts, such as $3 \cdot (4 + 9)$, the content of the `mfenced` element must be placed inside an `mrow` element:

```
<m:mn>3</m:mn>
<m:mo>&sdot;</m:mo>
<m:mfenced open="(" close=")">
   <m:mrow>
      <m:mn>4</m:mn>
      <m:mo>+</m:mo>
      <m:mn>9</m:mn>
   </m:mrow>
</m:mfenced>
```

A more complex example, involving nested parenthesis, is

$$4 \cdot (x + x \cdot (x + a)) \cdot (x - x \cdot (x - b))$$

The correct markup of this expression would be

```
<m:mn>4</m:mn>
<m:mo>&sdot;</m:mo>
<m:mfenced open="(" close=")">
   <m:mrow>
      <m:mi>x</m:mi>
      <m:mo>+</m:mo>
      <m:mi>x</m:mi>
      <m:mo>&sdot;</m:mo>
      <m:mfenced open="(" close=")">
         <m:mrow>
            <m:mi>x</m:mi>
            <m:mo>+</m:mo>
            <m:mi>a</m:mi>
```

```
            </m:mrow>
         </m:mfenced>
      </m:mrow>
   </m:mfenced>
   <m:mo>&sdot;</m:mo>
   <m:mfenced open="(" close=")">
      <m:mrow>
      <m:mi>x</m:mi>
      <m:mo>-</m:mo>
      <m:mi>x</m:mi>
      <m:mo>&sdot;</m:mo>
      <m:mfenced open="(" close=")">
         <m:mrow>
            <m:mi>x</m:mi>
            <m:mo>-</m:mo>
            <m:mi>b</m:mi>
         </m:mrow>
      </m:mfenced>
      </m:mrow>
   </m:mfenced>
```

Consult `https://www.w3.org/TR/MathML3/chapter3.html#presm.mfenced` for further information about use of the `mfenced` element.

3.4 Markup of the absolute value

The absolute value of a number, symbol or expression shall be done using the MathML element `mfenced` together with the `open` and `close` attributes. Both attributes must have the value `|`.

The expression

$$|-2| = 2$$

must be marked up as

```
<m:mfenced open="|" close="|">
   <m:mrow>
      <m:mo>-</m:mo>
      <m:mn>2</m:mn>
   </m:mrow>
</m:mfenced>
<m:mo>=</m:mo>
<m:mn>2</m:mn>
```

while the more complex expression

$$\left|\cos\frac{x}{2}\right| = \sqrt{\frac{1+\cos x}{2}}$$

must be marked up as follows:

```
<m:mfenced open="|" close="|">
   <m:mrow>
      <m:mi>cos</m:mi>
      <m:mo>&#8289;</m:mo>
      <m:mfrac>
         <m:mi>x</m:mi>
         <m:mn>2</m:mn>
      </m:mfrac>
   </m:mrow>
</m:mfenced>
<m:mo>=</m:mo>
<m:msqrt>
   <m:mrow>
      <m:mfrac>
         <m:mrow>
            <m:mn>1</m:mn>
            <m:mo>+</m:mo>
            <m:mrow>
               <m:mi>cos</m:mi>
               <m:mo>&#8289;</m:mo>
               <m:mi>x</m:mi>
            </m:mrow>
         </m:mrow>
         <m:mn>2</m:mn>
      </m:mfrac>
   </m:mrow>
</m:msqrt>
```

3.5 Markup of number of degrees

How to mark up a number of degrees, such as such as $360°$ or $-273.15°$ C, depends on whether the value is positive or negative.

For a positive number, the required markup is simple:

```
<m:mrow>
   <m:mn>[numeric value]</m:mn>
   <m:mi>&#176;</m:mi>
</m:mrow>
```

Please note the following:

- The markup must be placed inside an `mrow` element. This `mrow` must contain exactly two children.
- The first child of the `mrow` element must be an `mn` element, containing the relevant value.
- The second child of the `mrow` must be an `mo` element, containing the degree symbol, represented by the numeric entity `&#176;`.

Note also that the msup element must not be used, as the degree symbol by itself represents a raised ring.

For a negative value, the required markup is a bit more complex:

```
<m:mrow>
   <m:mrow>
      <m:mo>-</m:mo>
      <m:mn>[numeric value]</m:mn>
   </m:mrow>
   <m:mi>&#176;</m:mi>
</m:mrow>
```

Please note the following:

- Once again the markup must be placed inside an mrow element with exactly two children.
- However, this time the first child of the mrow element must be another mrow element:
  - The first child of this mrow element must be an mo element containing the normal hyphen sign.
  - The second child of this mrow element must be an mn element, containing the relevant absolute value.
- The second child of the containing mrow element must be an mo element, containing the degree symbol, represented by the numeric entity °.

This kind of markup is relevant both for temperature and for angular measurements. But for angular measurements there is – in addition to the normal 360 degree division of a circle – the *gradian measure* where the circle is divided into 400 gon, so that $360^\circ = 400^{\mathrm{g}}$.

The required markup for this kind of angular measurement is either:

```
<m:msup>
   <m:mn>[numeric value]</m:mn>
   <m:mtext>g</m:mtext>
</m:msup>
```

or

```
<m:msup>
   <m:mrow>
      <m:mo>-</m:mo>
      <m:mn>[numeric value]</m:mn>
   </m:mrow>
   <m:mtext>g</m:mtext>
</m:msup>
```

depending on the value, as described above.

Note the use of the `msup` element this time, in order to raise the unit g.

Examples:

The markup for the expression

$$0^\circ\ \mathrm{C} = 32^\circ\ \mathrm{F} = 273\ \mathrm{K}$$

is

```
<m:mrow>
   <m:mn>0</m:mn>
   <m:mi>&#176;</m:mi>
</m:mrow>
<m:mspace width="0.25em"/>
<m:mtext>C</m:mtext>
<m:mo>=</m:mo>
<m:mrow>
   <m:mn>32</m:mn>
   <m:mi>&#176;</m:mi>
</m:mrow>
<m:mspace width="0.25em"/>
<m:mtext>F</m:mtext>
<m:mo>=</m:mo>
<m:mn>273</m:mn>
<m:mspace width="0.25em"/>
<m:mtext>K</m:mtext>
```

Note the use of the `mspace` element to insert proper visual spacing between values and the relevant unit.

For clarity, one could use the following markup instead:

```
<m:mrow>
   <m:mrow>
      <m:mn>0</m:mn>
      <m:mi>&#176;</m:mi>
   </m:mrow>
   <m:mspace width="0.25em"/>
   <m:mtext>C</m:mtext>
</m:mrow>
<m:mo>=</m:mo>
<m:mrow>
   <m:mrow>
      <m:mn>32</m:mn>
      <m:mi>&#176;</m:mi>
   </m:mrow>
   <m:mspace width="0.25em"/>
   <m:mtext>F</m:mtext>
</m:mrow>
<m:mrow>
   <m:mo>=</m:mo>
```

```
    <m:mn>273</m:mn>
    <m:mspace width="0.25em"/>
    <m:mtext>K</m:mtext>
</m:mrow>
```

The only difference is the extra `mrow` elements that are used to group together the different parts of the expression.

The expression $T_0 = -273.15^\circ$ C must be represented by the following markup:

```
<m:msub>
    <m:mi>T</m:mi>
    <m:mn>0</m:mn>
</m:msub>
<m:mo>=</m:mo>
<m:mrow>
    <m:mrow>
        <m:mo>-</m:mo>
        <m:mn>273,15</m:mn>
    </m:mrow>
    <m:mi>&#176;</m:mi>
</m:mrow>
<m:mspace width="0.25em"/>
<m:mtext>C</m:mtext>
```

As a final example, the expression

$$\sin 45^\circ = \sin 50^\mathrm{g} = \sin \frac{\pi}{2} = \frac{1}{\sqrt{2}}$$

must be marked up as

```
<m:mrow>
    <m:mi>sin</m:mi>
    <m:mo>&#8289;</m:mo>
    <m:mrow>
        <m:mn>45</m:mn>
        <m:mi>&#176;</m:mi>
    </m:mrow>
</m:mrow>
<m:mo>=</m:mo>
<m:mrow>
    <m:mi>sin</m:mi>
    <m:mo>&#8289;</m:mo>
    <m:msup>
        <m:mn>50</m:mn>
        <m:mtext>g</m:mtext>
    </m:msup>
</m:mrow>
<m:mo>=</m:mo>
<m:mrow>
    <m:mi>sin</m:mi>
    <m:mo>&#8289;</m:mo>
```

```
    <m:mfrac>
       <m:mi>&pi;</m:mi>
       <m:mn>4</m:mn>
    </m:mfrac>
</m:mrow>
<m:mo>=</m:mo>
<m:mfrac>
    <m:mn>1</m:mn>
    <m:msqrt>
       <m:mn>2</m:mn>
    </m:msqrt>
</m:mfrac>
```

3.6 Markup of square roots and higher-order roots

The square root must be marked up using the MathML element `msqrt`. As the square root of a mathematical expression can be looked upon as a function with *one* argument, one could expect that the `sqrt` element always should have *exactly one* child. This would require that $\sqrt{a+b}$ should be marked up as

```
<m:msqrt>
    <m:mrow>
       <m:mi>a</m:mi>
       <m:mo>+</m:mo>
       <m:mi>b</m:mi>
    </m:mrow>
</m:msqrt>
```

And this is indeed perfectly good markup, but the simpler form

```
<m:msqrt>
    <m:mi>a</m:mi>
    <m:mo>+</m:mo>
    <m:mi>b</m:mi>
</m:msqrt>
```

would work just as well.

For root expressions other than the square root, the MathML element `mroot` must be used, and this time with the requirement that only two children are allowed. The first child must be the expression of which one wants to find the radical, and the second child must be the order of the root.

Thus, the expression

$$\sqrt[3]{8} = 2$$

must be marked up as

```
<m:mroot>
   <m:mn>8</m:mn>
   <m:mn>3</m:mn>
</m:mroot>
<m:mo>=</m:mo>
<m:mn>2</m:mn>
```

and the expression

$$\sqrt[n]{x} = x^{1/n}$$

must be marked up as

```
<m:mroot>
   <m:mi>x</m:mi>
   <m:mn>n</m:mn>
</m:mroot>
<m:mo>=</m:mo>
<m:msup>
   <m:mi>x</m:mi>
   <m:mfrac bevelled="true">
      <m:mn>1</m:mn>
      <m:mi>n</m:mi>
   </m:mfrac>
</m:msup>
```

3.7 Markup of exponentiation

We do not have any special requirements related to exponentiation, except that markup must be based on information given in
https://www.w3.org/TR/MathML3/chapter3.html#presm.msup.

3.8 Markup of vectors

There are several ways to specify a vector in mathematical notation. One way is to place a right arrow over the symbol representing the variable, such as

$$\vec{v} \quad \text{and} \quad \overrightarrow{AB}$$

Another way is to present the vector in some kind of bold font, as in

$$\mathbf{v} = v_x\,\mathbf{i} + v_y\,\mathbf{j} + v_z\,\mathbf{k}$$

For vector notation with arrows we require the following markup:

```
<m:mover>
   [a single element representing the vector]
   <m:mo>&#8594;</m:mo>
</m:mover>
```

Examples:

The very simple expression $\vec{v}$ must be represented by

```
<m:mover>
   <m:mo>v</m:mo>
   <m:mo>&#8594;</m:mo>
</m:mover>
```

Note that the placeholder `[a single element representing the vector]` may represent more complex notation. This means that $\vec{a} = \vec{a_1} + \vec{a_2}$ must be marked up as:

```
<m:mover>
   <m:mi>a</m:mi>
   <m:mo>&#8594;</m:mo>
</m:mover>
<m:mo>=</m:mo>
<m:mover>
   <m:msub>
      <m:mi>a</m:mi>
      <m:mn>1</m:mn>
   </m:msub>
   <m:mo>&#8594;</m:mo>
</m:mover>
<m:mo>+</m:mo>
<m:mover>
   <m:msub>
      <m:mi>a</m:mi>
      <m:mn>2</m:mn>
   </m:msub>
   <m:mo>&#8594;</m:mo>
</m:mover>
```

The required markup for vectors represented by a bold font, is to add the `mathvariant` attribute to the `mi` element representing the vector. The attribute value *must* be a string containing the substring `bold`, This will typically mean one of `bold`, `bold-italic`, `bold-sans-serif` or `sans-serif-bold-italic`:

```
<m:mi mathvariant="[string containing the substring 'bold']">
   [symbol representing the vector]
</m:mi>
```

This means that the expression

$$\mathbf{v} = v_x\,\mathbf{i} + v_y\,\mathbf{j} + v_z\,\mathbf{k}$$

can be marked up as

```
<m:mi mathvariant="bold">v</m:mi>
<m:mo>=</m:mo>
<m:msub>
   <m:mi>v</m:mi>
```

```
    <m:mi>x</m:mi>
</m:msub>
<m:mo>&#8290;</m:mo>
<m:mi mathvariant="bold">i</m:mi>
<m:mo>+</m:mo>
<m:msub>
    <m:mi>v</m:mi>
    <m:mi>y</m:mi>
</m:msub>
<m:mo>&#8290;</m:mo>
<m:mi mathvariant="bold">j</m:mi>
<m:mo>+</m:mo>
<m:msub>
    <m:mi>v</m:mi>
    <m:mi>z</m:mi>
</m:msub>
<m:mo>&#8290;</m:mo>
<m:mi mathvariant="bold">k</m:mi>
```

Note that another attribute value than `bold` could be used, in order to represent different types of bold font.

3.9 Markup of fractions

The MathML element `mfrac` must always be used for markup of fractions.

Even though the expression $1/2 + 1/2 = 1$ could be marked up as

```
<m:mn>1</m:mn>
<m:mo>/</m:mo>
<m:mn>2</m:mn>
<m:mo>+</m:mo>
<m:mn>1</m:mn>
<m:mo>/</m:mo>
<m:mn>2</m:mn>
<m:mo>=</m:mo>
<m:mn>1</m:mn>
```

we require the use of `mfrac` to represent the fractions:

```
<m:mfrac bevelled="true">
    <m:mn>1</m:mn>
    <m:mn>2</m:mn>
</m:mfrac>
<m:mo>+</m:mo>
<m:mfrac bevelled="true">
    <m:mn>1</m:mn>
    <m:mn>2</m:mn>
</m:mfrac>
<m:mo>=</m:mo>
<m:mn>1</m:mn>
```

Note the use of the `bevelled` attribute to separate the numerator and denominator with a slash rather than with a horizontal line.

Of course, if a horizontal line *is* required, as in this expression:

$$x = \frac{1}{a+b}$$

then the `bevelled` attribute should not be used:

```
<m:mi>x</m:mi>
<m:mo>=</m:mo>
<m:mfrac>
   <m:mn>1</m:mn>
   <m:mrow>
      <m:mi>a</m:mi>
      <m:mo>+</m:mo>
      <m:mi>b</m:mi>
   </m:mrow>
</m:mfrac>
```

Consult `https://www.w3.org/TR/MathML3/chapter3.html#presm.mfrac` for further information about use of the `mfrac` element.

3.10 Lower indices

For lower indices, as in $A_T = A_1 + A_2$ we use the MathML element `msub`.

For a numeric index, the required markup is

```
<m:msub>
   <m:mrow>
      <m:mi>
         [a single greek letter
         or
         a single letter in the reqions a-z or A-Z]
      </m:mi>
   </m:mrow>
   <m:mn>[one or more integers]</m:mn>
</m:msub>
```

and for a symbolic index, the required markup is

```
<m:msub>
   <m:mrow>
      <m:mi>
         [a single greek letter
         or
         a single letter in the reqions a-z or A-Z]
      </m:mi>
   </m:mrow>
   <m:mrow>
      <m:mi>
```

```
            [a single greek letter
            or
            a single letter in the reqions a-z or A-Z]
         </m:mi>
      </m:mrow>
   </m:msub>
```

Examples:

Based on this,

$$A_T = A_1 + A_2$$

must be marked up as

```
<m:msub>
   <m:mrow>
      <m:mi>A</m:mi>
   </m:mrow>
   <m:mi>T</m:mi>
</m:msub>
<m:mo>=</m:mo>
<m:msub>
   <m:mrow>
      <m:mi>A</m:mi>
   </m:mrow>
   <m:mn>1</m:mn>
</m:msub>
<m:mo>+</m:mo>
<m:msub>
   <m:mrow>
      <m:mi>A</m:mi>
   </m:mrow>
   <m:mn>2</m:mn>
</m:msub>
```

and

$$I_\alpha = \frac{I_\beta - I_\gamma}{2}$$

must be marked up as

```
<m:msub>
   <m:mrow>
      <m:mi>I</m:mi>
   </m:mrow>
   <m:mi>&alpha;</m:mi>
</m:msub>
<m:mo>=</m:mo>
<m:mfrac>
   <m:mrow>
      <m:msub>
         <m:mrow>
            <m:mi>I</m:mi>
```

```
            </m:mrow>
            <m:mi>&beta;</m:mi>
        </m:msub>
        <m:mo>-</m:mo>
        <m:msub>
            <m:mrow>
                <m:mi>I</m:mi>
            </m:mrow>
            <m:mi>&gamma;</m:mi>
        </m:msub>
    </m:mrow>
    <m:mn>2</m:mn>
</m:mfrac>
```

3.11 Markup of functions with one argument

A mathematical function, such as $f(x)$, $x(t)$, $F(x)$, $\psi(t)$ and similar, must be marked up as follows:

```
<m:mrow>
    <m:mi>
        [a single greek letter
        or
        a single letter in the reqions a-z or A-Z]
    </m:mi>
    <m:mo>&#8289;</m:mo>
    <m:mfenced open="(" close=")">[any one child]</m:mfenced>
</m:mrow>
```

Please note the following:

- The markup of the function must be placed inside an `mrow` element. This `mrow` must contain exactly three children.
- The first child of the `mrow` must be an `mi` element, containing one single Greek or single English letter, in upper or lower case.
- The second child of the `mrow` must be an `mo` element, containing the *Function Application Entity* `&#8289;`.
- The third child of the `mrow` must be an `mfenced` element, containing the argument to the function. The `mfenced` element must have exactly one child. Apart from that, there are no requirements on the content of the `mfenced` element.

Examples:

The expression

$$g(x)$$

shall be marked up as

```
<m:mrow>
   <m:mi>g</m:mi>
   <m:mo>&#8289;</m:mo>
   <m:mfenced open="(" close=")">
      <m:mi>x</m:mi>
   </m:mfenced>
</m:mrow>
```

And similar, the expression

$$\psi(t)$$

shall be marked up as

```
<m:mrow>
   <m:mi>&psi;</m:mi>
   <m:mo>&#8289;</m:mo>
   <m:mfenced open="(" close=")">
      <m:mi>t</m:mi>
   </m:mfenced>
</m:mrow>
```

If the argument to the function is more complicated, such as in

$$f(x + \Delta x)$$

the corresponding markup will also be more complicated:

```
<m:mrow>
   <m:mi>f</m:mi>
   <m:mo>&#8289;</m:mo>
   <m:mfenced open="(" close=")">
      <m:mrow>
         <m:mi>x</m:mi>
         <m:mo>+</m:mo>
         <m:mi>&Delta;</m:mi>
         <m:mi>x</m:mi>
      </m:mrow>
   </m:mfenced>
</m:mrow>
```

3.12 Markup of functions with two or more arguments

A mathematical function with two or more arguments, such as $f(x, y, z)$, $F(x, t)$, $\psi(r, \theta)$ and similar, must be marked up as follows:

```
<m:mrow>
   <m:mi>
      [a single greek letter
      or
      a single letter in the reqions a-z or A-Z]
   </m:mi>
```

```
    <m:mo>&#8289;</m:mo>
    <m:mfenced open="(" close=")">[two or more children]</m:mfenced>
</m:mrow>
```

Please note the following:

- The markup of the function must be placed inside an `mrow` element. This `mrow` element must contain exactly three children.
- The first child of the `mrow` element must be an `mi` element, containing one single Greek or single English letter, in upper or lower case.
- The second child of the `mrow` element must be an `mo` element, containing the *Function Application Entity* `&#8289;`.
- The third child of the `mrow` element must be an `mfenced` element, containing the arguments to the function. The number of children of the `mfenced` element must be equal to the number of arguments to the function. Apart from that, there are no requirements on the content of the `mfenced` element.

  Please observe that, when the `mfenced` element is rendered correctly, the comma separators are automatically inserted.

Examples:

The expression

$$f(x, y, z)$$

shall be marked up as

```
<m:mrow>
    <m:mi>f</m:mi>
    <m:mo>&#8289;</m:mo>
    <m:mfenced open="(" close=")">
        <m:mi>x</m:mi>
        <m:mi>y</m:mi>
        <m:mi>z</m:mi>
    </m:mfenced>
</m:mrow>
```

And the two-argument function $\psi(r, \theta)$ shall be marked up as

```
<m:mrow>
    <m:mi>&psi;</m:mi>
    <m:mo>&#8289;</m:mo>
    <m:mfenced open="(" close=")">
        <m:mi>r</m:mi>
        <m:mi>&theta;</m:mi>
    </m:mfenced>
</m:mrow>
```

3.13 Markup of named functions

A known mathematical function, such as $\sin\alpha$, $\ln x$, $\arccos(x)$ and similar, must be marked up as follows:

```
<m:mrow>
   <m:mi>[function name]</m:mi>
   <m:mo>&#8289;</m:mo>
   [any one element that represents the argument(s) to the function]
</m:mrow>
```

Please note the following:

- The markup of the function must be placed inside an `mrow` element. This `mrow` must contain exactly three children.
- The first child of the `mrow` must be an `mi` element, containing the name of the function.
- The second child of the `mrow` must be an `mo` element, containing the *Function Application Entity* `&#8289;`.
- There are no particular requirements to the last element, except that it must correctly represent the argument(s) to the function.

Examples:

The expression

$$g(\alpha) = \sin\alpha$$

shall be marked up as

```
<m:mrow>
   <m:mi>g</m:mi>
   <m:mo>&#8289;</m:mo>
   <m:mfenced open="(" close=")">
      <m:mi>&alpha;</m:mi>
   </m:mfenced>
</m:mrow>
<m:mo>=</m:mo>
<m:mrow>
   <m:mi>sin</m:mi>
   <m:mo>&#8289;</m:mo>
   <m:mi>&alpha;</m:mi>
</m:mrow>
```

And, the expression

$$\ln(x\,y) = \ln x + \ln y$$

shall be marked up as

```
<m:mrow>
   <m:mi>ln</m:mi>
   <m:mo>&#8289;</m:mo>
   <m:mfenced open="(" close=")">
      <m:mrow>
         <m:mi>x</m:mi>
         <m:mo>&InvisibleTimes;</m:mo>
         <m:mi>y</m:mi>
      </m:mrow>
   </m:mfenced>
</m:mrow>
<m:mo>=</m:mo>
<m:mrow>
   <m:mi>ln</m:mi>
   <m:mo>&#8289;</m:mo>
   <m:mi>x</m:mi>
</m:mrow>
<m:mo>+</m:mo>
<m:mrow>
   <m:mi>ln</m:mi>
   <m:mo>&#8289;</m:mo>
   <m:mi>y</m:mi>
</m:mrow>
```

Note that, for an expression on the form $\cos 2x$, the following markup is **NOT** correct:

```
<m:mrow>
   <m:mi>cos</m:mi>
   <m:mo>&#8289;</m:mo>
   <m:mn>2</m:mn>
   <m:mo>&#8290;</m:mo>
   <m:mi>x</m:mi>
</m:mrow>
```

This markup breaks the requirement that the `mrow` element must contain exactly three children. Instead, the markup must be as follows:

```
<m:mrow>
   <m:mi>cos</m:mi>
   <m:mo>&#8289;</m:mo>
   <m:mrow>
      <m:mn>2</m:mn>
      <m:mo>&#8290;</m:mo>
      <m:mi>x</m:mi>
   </m:mrow>
</m:mrow>
```

3.14 Markup of limits and the derivative

The limit of a mathematical expression must generically be marked up as follows:

```
<m:mrow>
   <m:munder>
      <m:mo>lim</m:mo>
      [element representing the conditions]
   </m:munder>
   <m:mo>&#8289;</m:mo>
   [one element representing some mathematcal function or expression]
</m:mrow>
```

Please note the following:

- The construction must be placed inside an `mrow` element. This `mrow` must contain exactly three children.
- The first child of the `mrow` element must be an `munder` element. This `munder` element must contain exactly two elements.
  - The first child of the `munder` element must be an `mo` element containing the text `lim`.
  - The second child of the `munder` element must represent the condition for finding the limit. If necessary, place the condition inside an `mrow` element to represent it as one element.
- The second child of the `mrow` must be an `mo` element, containing the *Function Application Entity* `&#8289;`.
- There are no particular requirements to the last element, except that it must correctly represent the expression to find the limit of. If necessary, place the expression inside an `mrow` element to represent it as one element.

Examples:

The expression

$$g(x) = \lim_{t \to T} f(x,t)$$

shall be marked up as

```
<m:mrow>
   <m:mi>g</m:mi>
   <m:mo>&#8289;</m:mo>
   <m:mfenced open="(" close=")">
      <m:mi>x</m:mi>
   </m:mfenced>
</m:mrow>
<m:mo>=</m:mo>
<m:mrow>
   <m:munder>
      <m:mo>lim</m:mo>
      <m:mrow>
         <m:mi>t</m:mi>
         <m:mo>&rarr;</m:mo>
         <m:mi>T</m:mi>
```

```
        </m:mrow>
    </m:munder>
    <m:mo>&#8289;</m:mo>
    <m:mrow>
        <m:mi>f</m:mi>
        <m:mo>&#8289;</m:mo>
        <m:mfenced open="(" close=")">
            <m:mi>x</m:mi>
            <m:mi>t</m:mi>
        </m:mfenced>
    </m:mrow>
</m:mrow>
```

And, the expression

$$\lim_{x \to 0} \frac{1}{x} = \infty$$

shall be marked up as

```
<m:mrow>
    <m:munder>
        <m:mo>lim</m:mo>
        <m:mrow>
            <m:mi>x</m:mi>
            <m:mo>&rarr;</m:mo>
            <m:mn>0</m:mn>
        </m:mrow>
    </m:munder>
    <m:mo>&#8289;</m:mo>
    <m:mfrac>
        <m:mn>1</m:mn>
        <m:mi>x</m:mi>
    </m:mfrac>
</m:mrow>
<m:mo>=</m:mo>
<m:mi>&infin;</m:mi>
```

There are several ways to write the derivative of a mathematical expression. One is to add a prime symbol after the function or expression, as in

$$f'(x) \quad \text{and} \quad (\sin x)'$$

This is *Lagrange's notation*. An alternative is to use *Leibniz's notation*:

$$\frac{\mathrm{d}f(x)}{\mathrm{d}x} \quad \text{and} \quad \frac{\mathrm{d}\sin x}{\mathrm{d}x}$$

There are several more ways to write the derivative, but these are the most common, so we will focus on these two.

3.14.1 The *Lagrange notation*

When we want to write the derivative of a function using *Lagrange's notation*, as in $f'(x)$, the required markup is

```
<m:mrow>
   <m:msup>
      <m:mi>
         [a single greek letter
         or
         a single letter in the reqions a-z or A-Z]
      </m:mi>
      <m:mo>&#8242;</m:mo>
   </m:msup>
   <m:mo>&#8289;</m:mo>
   <m:mfenced open="(" close=")">[markup of one or more arguments]</m:mfenced>
</m:mrow>
```

Please note the following:

- The markup of the derivative of a function must be placed inside an `mrow` element. This `mrow` must contain exactly three children.
- The first child of the `mrow` element must be an `msup` element.
  - The first child of the `msup` element must be an `mi` element, containing one single Greek or single English letter, in upper or lower case.
  - The second child of the `msup` element must be an `mo` element, containing the entity for the prime symbol, `&#8242;`.
    Note: As an alternative to the `&#8242;` entity, we also allow use of the `&#8243;` entity to represent the second derivative, and the `&#8244;` to represent the third derivative.
- The second child of the `mrow` element must be an `mo` element, containing the *Function Application Entity* `&#8289;`.
- The third child of the `mrow` element must be an `mfenced` element, containing the argument(s) to the function.

Examples:

The definition of the derivative of a function f is often given as

$$f'(x) = \lim_{h \to 0} \frac{f(x+h) - f(h)}{h}$$

The correct markup of this expression is

```
<m:mrow>
   <m:msup>
      <m:mi>f</m:mi>
      <m:mo>&#8242;</m:mo>
```

```
      </m:msup>
      <m:mo>&#8289;</m:mo>
      <m:mfenced open="(" close=")">
         <m:mi>x</m:mi>
      </m:mfenced>
   </m:mrow>
   <m:mo>=</m:mo>
   <m:mrow>
      <m:munder>
         <m:mo>lim</m:mo>
         <m:mrow>
            <m:mi>h</m:mi>
            <m:mo>&rarr;</m:mo>
            <m:mn>0</m:mn>
         </m:mrow>
      </m:munder>
      <m:mo>&#8289;</m:mo>
      <m:mfrac>
         <m:mrow>
            <m:mrow>
               <m:mi>f</m:mi>
               <m:mo>&#8289;</m:mo>
               <m:mfenced open="(" close=")">
                  <m:mrow>
                     <m:mi>x</m:mi>
                     <m:mo>+</m:mo>
                     <m:mi>h</m:mi>
                  </m:mrow>
               </m:mfenced>
            </m:mrow>
            <m:mo>-</m:mo>
            <m:mrow>
               <m:mi>f</m:mi>
               <m:mo>&#8289;</m:mo>
               <m:mfenced open="(" close=")">
                  <m:mi>x</m:mi>
               </m:mfenced>
            </m:mrow>
         </m:mrow>
         <m:mi>h</m:mi>
      </m:mfrac>
   </m:mrow>
```

And

$$f''(x) = f'(f'(x))$$

must be marked up as

```
   <m:mrow>
      <m:msup>
         <m:mi>f</m:mi>
         <m:mo>&#8243;</m:mo>
```

```
    </m:msup>
    <m:mo>&#8289;</m:mo>
    <m:mfenced open="(" close=")">
       <m:mi>x</m:mi>
    </m:mfenced>
</m:mrow>
<m:mo>=</m:mo>
<m:mrow>
    <m:msup>
       <m:mi>f</m:mi>
       <m:mo>&#8242;</m:mo>
    </m:msup>
    <m:mo>&#8289;</m:mo>
    <m:mfenced open="(" close=")">
       <m:mrow>
          <m:msup>
             <m:mi>f</m:mi>
             <m:mo>&#8242;</m:mo>
          </m:msup>
          <m:mo>&#8289;</m:mo>
          <m:mfenced open="(" close=")">
             <m:mi>x</m:mi>
          </m:mfenced>
       </m:mrow>
    </m:mfenced>
</m:mrow>
```

When we want to write the derivative of a named function using *Lagrange's notation*, as in $(\sin x)'$, the required markup is

```
<m:msup>
    <m:mfenced open="(" close=")">
       [markup representing the function]
    </m:mfenced>
    <m:mo>&#8242;</m:mo>
</m:msup>
```

Please note the following:

- The markup of the derivative of a named function must be placed inside an `msup` element.
- The first child of the `msup` element must be an `mfenced` element, and that `mfenced` element must contain the markup of the function.
- The second child of the `msup` element must be an `mo` element, containing the entity for the prime symbol, `&prime;`.

  Note: As an alternative to the `&#8242;` entity, we also allow use of the `&#8243;` entity to represent the second derivative, and the `&#8244;` to represent the third derivative.

Examples:

The expression

$$(3ax^3)'' = (9ax^2)' = 18ax$$

must be marked up as

```
<m:msup>
   <m:mfenced open="(" close=")">
      <m:mrow>
         <m:mn>3</m:mn>
         <m:mo>&#8290;</m:mo>
         <m:mi>a</m:mi>
         <m:mo>&#8290;</m:mo>
         <m:msup>
            <m:mi>x</m:mi>
            <m:mn>3</m:mn>
         </m:msup>
      </m:mrow>
   </m:mfenced>
   <m:mo>&#8243;</m:mo>
</m:msup>
<m:mo>=</m:mo>
<m:msup>
   <m:mfenced open="(" close=")">
      <m:mrow>
         <m:mn>9</m:mn>
         <m:mo>&#8290;</m:mo>
         <m:mi>a</m:mi>
         <m:mo>&#8290;</m:mo>
         <m:msup>
            <m:mi>x</m:mi>
            <m:mn>2</m:mn>
         </m:msup>
      </m:mrow>
   </m:mfenced>
   <m:mo>&#8242;</m:mo>
</m:msup>
<m:mo>=</m:mo>
<m:mn>18</m:mn>
<m:mo>&#8290;</m:mo>
<m:mi>a</m:mi>
<m:mo>&#8290;</m:mo>
<m:mi>x</m:mi>
```

3.14.2 The *Leibniz notation*

When we want to write the derivative of a function using *Leibniz's notation*, as in

$$\frac{\mathrm{d}f(x)}{\mathrm{d}x}$$

the required markup is

```
<m:mfrac>
   <m:mrow>
      <m:mo>&#8518;</m:mo>
      [markup required to represent the function]
   </m:mrow>
   <m:mrow>
      <m:mo>&#8518;</m:mo>
      <m:mi>
         [a letter in the region a-z]
      </m:mi>
   </m:mrow>
</m:mfrac>
```

Please note the following:

- The markup of the derivative of a function, using *Leibniz's notation,* must be placed inside an `mfrac` element.
- The first child of the `mfrac` element must be an `mrow` element.
  - The first child of this `mrow` element must be an `mo` element, containing the entity for the *differential d* symbol, `&#8518;`.
  - The rest of this `mrow` element must be filled up with the necessary markup to represent the function.
- The second child of the `mfrac` element must also be an `mrow` element, with only two children.
  - The first child of this second `mrow` element must again be an `mo` element, also this one containing the entity for the *differential d* symbol, `&#8518;`.
  - The second (and last) child of the `mrow` element must be an `mi` element, containg a single letter in the region "a to 'z".

Examples:

The expression

$$f'(x) = \frac{\mathrm{d}f(x)}{\mathrm{d}x}$$

must be marked up as

```
<m:mrow>
   <m:msup>
      <m:mi>f</m:mi>
      <m:mo>&#8242;</m:mo>
   </m:msup>
   <m:mo>&#8289;</m:mo>
   <m:mfenced open="(" close=")">
      <m:mi>x</m:mi>
   </m:mfenced>
</m:mrow>
<m:mo>=</m:mo>
```

```
<m:mfrac>
   <m:mrow>
      <m:mo>&#8518;</m:mo>
      <m:mrow>
         <m:mi>f</m:mi>
         <m:mo>&#8289;</m:mo>
         <m:mfenced open="(" close=")">
            <m:mi>x</m:mi>
         </m:mfenced>
      </m:mrow>
   </m:mrow>
   <m:mrow>
      <m:mo>&#8518;</m:mo>
      <m:mi>x</m:mi>
   </m:mrow>
</m:mfrac>
```

Markup of higher order derivatives of a function, based on *Leibniz's notation*, is just an extension of the markup of the first derivative:

```
<m:mfrac>
   <m:mrow>
      <m:msup>
         <m:mo>&#8518;</m:mo>
         <m:mn>
            [order of differentiation]
         </m:mn>
      </m:msup>
      [markup required to represent the function]
   </m:mrow>
   <m:mrow>
      <m:mo>&dd;</m:mo>
      <m:msup>
         <m:mi>
            [a letter in the region a-z]
         </m:mi>
         <m:mn>
            [order of differentiation]
         </m:mn>
      </m:msup>
   </m:mrow>
</m:mfrac>
```

Please note the following:

- The markup of the second derivative of a function, using *Leibniz's notation*, must be placed inside an `mfrac` element.
- The first child of the `mfrac` element must be an `mrow` element.
  - The first child of this `mrow` element must be an `msup` element

    - The first child of this `msup` element must be an `mo` element, containing the entity for the *differential d* symbol, `&#8518;`.
    - The second child of this `msup` element must be an `mn` element, containing the number representing the order of differentiation.
  - The rest of this `mrow` element must be filled up with the necessary markup to represent the function.

- The second child of the `mfrac` element must also be an `mrow` element, with only two children.
  - The first child of this second `mrow` element must again be an `mo` element, also this one containing the entity for the *differential d* symbol, `&#8518;`.
  - The second (and last) child of the `mrow` element must be an `msup` element
    - The first child of this `msup` element must be an `mi` element, containg a single letter in the region "a to "z".
    - The second child of this `msup` element must be an `mn` element, containing the number representing the order of differentiation.

Examples:

The expression

$$g''(x) = \frac{\mathrm{d}^2 \sin x}{\mathrm{d}x^2}$$

must be marked up as

```
<m:mrow>
   <m:msup>
      <m:mi>g</m:mi>
      <m:mo>&#8243;</m:mo>
   </m:msup>
   <m:mo>&#8289;</m:mo>
   <m:mfenced open="(" close=")">
      <m:mi>x</m:mi>
   </m:mfenced>
</m:mrow>
<m:mo>=</m:mo>
<m:mfrac>
   <m:mrow>
      <m:msup>
         <m:mo>&#8518;</m:mo>
         <m:mn>2</m:mn>
      </m:msup>
      <m:mrow>
         <m:mi>sin</m:mi>
         <m:mo>&#8289;</m:mo>
         <m:mi>x</m:mi>
      </m:mrow>
   </m:mrow>
   <m:mrow>
      <m:mo>&#8518;</m:mo>
```

```
        <m:msup>
           <m:mi>x</m:mi>
           <m:mn>2</m:mn>
        </m:msup>
     </m:mrow>
  </m:mfrac>
```

3.15 Markup of the integral

The following entities are relevant for markup of integrals:

Symbol	Entity	Numeric	Description
$\int$	∫	∫	Integral
$\iint$	∬	∬	Double integral
$\iiint$	∭	∭	Triple integral
$\oint$	∮	∮	Contour integral

For the *indefinite integral*, with a very simple argument, such as $\int f = g + C$ and $T = \oint f$ and even $a(x) = \iint \cos x$, the following markup will be sufficient:

```
<m:mo>&int;</m:mo>
<m:mi>f</m:mi>
<m:mo>=</m:mo>
<m:mi>g</m:mi>
<m:mo>+</m:mo>
<m:mi>C</m:mi>
```

and

```
<m:mi>T</m:mi>
<m:mo>=</m:mo>
<m:mo>&#8750;</m:mo>
<m:mi>g</m:mi>
```

and

```
<m:mrow>
   <m:mi>a</m:mi>
   <m:mo>&#8289;</m:mo>
   <m:mfenced open="(" close=")">
      <m:mi>x</m:mi>
   </m:mfenced>
</m:mrow>
<m:mo>=</m:mo>
<m:mo>&#8748;</m:mo>
<m:mrow>
   <m:mi>cos</m:mi>
   <m:mo>&#8289;</m:mo>
   <m:mi>x</m:mi>
</m:mrow>
```

respectively.

However, once the expressions get a little more complex, the requirements get more strict. Just the need to specify the integration variable, as in

$$\int f(x)\mathrm{d}x$$

will require a very specific markup:

```
<m:mo>&int;</m:mo>
[markup of function to be integrated
<m:mrow>
   <m:mo>&dd;</m:mo>
   <m:mi>[integration variable]</m:mi>
</m:mrow>
```

Note that the `&int;` entity in the first `mo` element could just as well be any of the other integral entities listed on the preceding page.

So the complete markup of

$$C(z) = \oint \frac{1}{z}\mathrm{d}z$$

would be

```
<m:mrow>
   <m:mi>C</m:mi>
   <m:mo>&#8289;</m:mo>
   <m:mfenced open="(" close=")">
      <m:mi>z</m:mi>
   </m:mfenced>
</m:mrow>
<m:mo>=</m:mo>
<m:mo>&oint;</m:mo>
<m:mfrac>
   <m:mn>1</m:mn>
   <m:mi>z</m:mi>
</m:mfrac>
<m:mrow>
   <m:mo>&dd;</m:mo>
   <m:mi>z</m:mi>
</m:mrow>
```

For *definite integrals*, where the interval of the integration is specified, there are further requirements to the markup.

```
<m:mrow>
   <m:munderover>
      <m:mo>[integration sign]</m:mo>
      [upper limit]
      [lower limit]
   </m:munderover>
```

```
    [any markup representing the expression to be integrated]
    <m:mrow>
       <m:mo>&dd;</m:mo>
       <m:mi>[a letter in the region a-z]</m:mi>
    </m:mrow>
</m:mrow>
```

Please note the following:

- The whole expression must be placed inside an `mrow` element.
- This `mrow` must have three children:
  - First, an `munderover` element, with exactly three children:
    - The first child of this `munderover` element must be an `mo` element containing one of the entities listed on page 35.
    - The second child must be an element containing the necessary markup to represent the lower limit of the integration interval.
    - The last child must be an element containing the necessary markup to represent the upper limit of the integration interval.
  - Second, any element containing the markup of the expression to be integrated.
  - Third, an `mrow` element, containing exactly two children:
    - The first child of this `mrow` element must be an `mo` element, also this one containing the entity for the *differential d* symbol, `&DifferentialD;`.
    - The second child of this `mrow` element must be an `mi` element, containing a single letter in the region ”a to ”z”.

Examples:

The expression

$$F(x) = \int_{t=a}^{t=x} f(t)\mathrm{d}t$$

must be marked up as

```
<m:mrow>
   <m:mi>F</m:mi>
   <m:mo>&#8289;</m:mo>
   <m:mfenced open="(" close=")">
      <m:mi>x</m:mi>
   </m:mfenced>
</m:mrow>
<m:mo>=</m:mo>
<m:mrow>
   <m:munderover>
      <m:mo>&int;</m:mo>
      <m:mrow>
         <m:mi>t</m:mi>
         <m:mo>=</m:mo>
```

```
            <m:mi>a</m:mi>
         </m:mrow>
         <m:mrow>
            <m:mi>t</m:mi>
            <m:mo>=</m:mo>
            <m:mi>x</m:mi>
         </m:mrow>
      </m:munderover>
      <m:mrow>
         <m:mi>f</m:mi>
         <m:mo>&#8289;</m:mo>
         <m:mfenced open="(" close=")">
            <m:mi>t</m:mi>
         </m:mfenced>
      </m:mrow>
      <m:mrow>
         <m:mo>&dd;</m:mo>
         <m:mi>t</m:mi>
      </m:mrow>
   </m:mrow>
```

Another example:

$$\oint \frac{1}{z}\mathrm{d}z = \int_0^{2\pi} \frac{1}{e^{it}}\, i\, e^{it}\mathrm{d}t$$

must be represented by the following markup:

```
<m:mo>&oint;</m:mo>
<m:mfrac>
   <m:mn>1</m:mn>
   <m:mi>z</m:mi>
</m:mfrac>
<m:mrow>
   <m:mo>&dd;</m:mo>
   <m:mi>z</m:mi>
</m:mrow>
<m:mo>=</m:mo>
<m:mrow>
   <m:munderover>
      <m:mo>&int;</m:mo>
      <m:mn>0</m:mn>
      <m:mrow>
         <m:mn>2</m:mn>
         <m:mo>&#8290;</m:mo>
         <m:mi>&pi;</m:mi>
      </m:mrow>
   </m:munderover>
   <m:mrow>
      <m:mfrac>
         <m:mn>1</m:mn>
         <m:msup>
            <m:mi>e</m:mi>
```

```
            <m:mrow>
               <m:mi>i</m:mi>
               <m:mo>&#8290;</m:mo>
               <m:mi>t</m:mi>
            </m:mrow>
         </m:msup>
      </m:mfrac>
      <m:mo>&#8290;</m:mo>
      <m:mi>i</m:mi>
      <m:mo>&#8290;</m:mo>
      <m:msup>
         <m:mi>e</m:mi>
         <m:mrow>
            <m:mi>i</m:mi>
            <m:mo>&#8290;</m:mo>
            <m:mi>t</m:mi>
         </m:mrow>
      </m:msup>
   </m:mrow>
   <m:mrow>
      <m:mo>&dd;</m:mo>
      <m:mi>t</m:mi>
   </m:mrow>
</m:mrow>
```

3.16 Markup of chemistry

Every `math` element that is a chemical formula, will have to have added the class "chemistry" to them to stand out from normal MathML notation - because chemical formulas might have a different meaning in math. We will need to use the `mtext` element to define them. Each of these `mtext` elements must have a `class="asciimath"` attribute.

For the following chemical formula for photosynthesis:

$$6\,CO_2 + 6\,H_2O + \text{light} \longrightarrow C_6H_{12}O_6 + 6\,O_2 \tag{1}$$

The markup would be as follows:

```
<m:math
    xmlns:m="http://www.w3.org/1998/Math/MathML"
    alttext="6 CO_2 + 6 H_2O + light rarr C_6H_12O_6 + 6 O_2"
    display="block"
    class="chemistry">
   <m:mn>6</m:mn>
   <m:mtext class="asciimath">C O_2</m:mtext>
   <m:mo>+</m:mo>
   <m:mn>6</m:mn>
   <m:mtext class="asciimath">H_2 O</m:mtext>
   <m:mo>+</m:mo>
   <m:mtext>light</m:mtext>
```

```
    <m:mo>&#8594;</m:mo>
    <m:mtext class="asciimath">C_6 H_12 O_6</m:mtext>
    <m:mo>+</m:mo>
    <m:mn>6</m:mn>
    <m:mtext class="asciimath">O_2</m:mtext>
</m:math>
```

Note: This markup will most likely be changed in the near future, and we're open to suggestions on how to improve it.

3.17 Markup of physics

Every `math` element that is a physics formula, will have to have added the class "physics" to them to stand out from normal MathML notation - because physics formulas might have a different meaning in math.

The markup would be as follows:

```
<m:math
    xmlns:m="http://www.w3.org/1998/Math/MathML"
    alttext="..."
    display="block"
   class="physics">
    ...
</m:math>
```

Note: This markup will most likely be changed in the near future, and we're open to suggestions on how to improve it.