MathML

HANS HAGEN

```
math> <apply> <eq/> <apply> <diff/> <bvar> <ci> x </ci> </bvar> <ci> a </ci> </apply> <ci> 0 </ci> </apply> </math> <math> <apply> <eq/> <apply>
<diff/> <bvar> <ci> x </ci> </bvar> <ci> x </ci> </apply> <cn> 1 </cn> </apply> </math> <math> <apply> <eq/> <apply> <diff/> <bvar> <ci> x </ci>
</bvar> <apply> <times/> <ci> a </ci> <ci> u </ci> </apply> </apply> <apply> <times/> <ci> a </ci> <apply> <diff/> <bvar> <ci> x </ci> </bvar> <ci>
u </ci> </apply> </apply> </apply> </math> <math> <apply> <eq/> <apply> <diff/> <bvar> <ci> x </ci> </bvar> <apply> <plus/> <ci> u </ci> <ci> v
</ci> <ci> w </ci> </apply> </apply> <apply> <plus/> <apply> <diff/> <bvar> <ci> x </ci> </bvar> <ci> u </ci> </apply> <apply> <diff/> <bvar> <ci>
x </ci> </bvar> <ci> v </ci> </apply> <apply> <diff/> <bvar> <ci> x </ci> </bvar> <ci> w </ci> </apply> </apply> </apply> </math> <math> <apply>
<eq/> <apply> <diff/> <bvar> <ci> x </ci> </bvar> <apply> <times/> <ci> u </ci> <ci> v </ci> </apply> </apply> <apply> <plus/> <apply> <times/> <ci>
u </ci> <apply> <diff/> <bvar> <ci> x </ci> </bvar> <ci> u </ci> </apply> </apply> <apply> <times/> <ci> v </ci> <apply> <diff/> <bvar> <ci> x </ci>
</bvar> <ci> v </ci> </apply> </apply> </apply> </apply> </math> <math> <apply> <eq/> <apply> <diff/> <bvar> <ci> x </ci> </bvar> <apply> <times/>
<ci> u </ci> <ci> v </ci> <ci> w </ci> </apply> </apply> <apply> <plus/> <apply> <times/> <ci> v </ci> <ci> w </ci> <apply> <diff/> <bvar> <ci> x </ci>
</bvar> <ci> u </ci> </apply> </apply> <apply> <times/> <ci> u </ci> <ci> w </ci> <apply> <diff/> <bvar> <ci> x </ci> </bvar> <ci> v </ci> </apply>
</apply> <apply> <times/> <ci> u </ci> <ci> v </ci> <apply> <diff/> <bvar> <ci> x </ci> </bvar> <ci> w </ci> </apply> </apply> </apply> </apply>
</math> <math> <apply> <eq/> <apply> <diff/> <bvar> <ci> x </ci> </bvar> <apply> <divide/> <ci> u </ci> <ci> v </ci> </apply> </apply> <apply>
<divide/> <apply> <minus/> <apply> <times/> <ci> v </ci> <apply> <diff/> <bvar> <ci> x </ci> </bvar> <ci> u </ci> </apply> </apply> <apply> <times/>
<ci> u </ci> <apply> <diff/> <bvar> <ci> x </ci> </bvar> <ci> v </ci> </apply> </apply> </apply> <apply> <power/> <ci> v </ci> <cn> 2 </cn> </apply>
</apply> <apply> <minus/> <apply> <times/> <apply> <divide/> <cn> 1 </cn> <ci> v </ci> </apply> <apply> <diff/> <bvar> <ci> x </ci> </bvar> <ci>
u </ci> </apply> </apply> <apply> <times/> <apply> <divide/> <cn> u </cn> <apply> <power/> <ci> v </ci> <cn> 2 </cn> </apply> </apply> <apply>
<diff/> <bvar> <ci> x </ci> </bvar> <ci> v </ci> </apply> </apply> </apply> </apply> </math> <math> <apply> <eq/> <apply> <diff/> <bvar> <ci> x </ci>
</bvar> <apply> <power/> <ci> u </ci> <ci> n </ci> </apply> </apply> <apply> <times/> <ci> n </ci> <apply> <power/> <ci> u </ci> <apply> <minus/>
<ci> n </ci> <cn> 1 </cn> </apply> </apply> <apply> <diff/> <bvar> <ci> x </ci> </bvar> <ci> u </ci> </apply> </apply> </apply> </math> <math>
<apply> <eq/> <apply> <diff/> <bvar> <ci> x </ci> </bvar> <apply> <root/> <ci> u </ci> </apply> </apply> <apply> <times/> <apply> <divide/> <cn>
1 </cn> <apply> <times/> <cn> 2 </cn> <apply> <root/> <ci> u </ci> </apply> </apply> </apply> <apply> <diff/> <bvar> <ci> x </ci> </bvar> <ci> u
</ci> </apply> </apply> </apply> </math> <math> <apply> <eq/> <apply> <diff/> <bvar> <ci> x </ci> </bvar> <apply> <divide/> <cn> 1 </cn> <ci> u
</ci> </apply> </apply> <apply> <times/> <apply> <minus/> <apply> <divide/> <cn> 1 </cn> <apply> <power/> <ci> u </ci> <cn> 2 </cn> </apply>
</apply> </apply> <apply> <diff/> <bvar> <ci> x </ci> </bvar> <ci> u </ci> </apply> </apply> </apply> </math> <math> <apply> <eq/> <apply> <diff/>
<bvar> <ci> x </ci> </bvar> <apply> <divide/> <cn> 1 </cn> <apply> <power/> <ci> u </ci> <cn> n </cn> </apply> </apply> </apply> <apply> <times/>
<apply> <minus/> <apply> <divide/> <ci> n </ci> <apply> <power/> <ci> u </ci> <apply> <plus/> <ci> n </ci> <cn> 1 </cn> </apply> </apply> </apply>
</apply> <apply> <diff/> <bvar> <ci> x </ci> </bvar> <ci> u </ci> </apply> </apply> </apply> </math> <math> <apply> <eq/> <apply> <diff/> <bvar>
<ci> x </ci> </bvar> <apply> <inverse/> <apply> <sinh/> <ci> u </ci> </apply> </apply> </apply> <apply> <diff/> <bvar> <ci> x </ci> </bvar> <apply>
<log/> <apply> <plus/> <ci> u </ci> <apply> <root/> <apply> <plus/> <apply> <power/> <ci> u </ci> <cn> 2 </cn> </apply> <cn> 1 </cn> </apply>
</apply> </apply> </apply> </apply> <apply> <times/> <apply> <divide/> <cn> 1 </cn> <apply> <root/> <apply> <plus/> <apply> <power/> <ci> u
</ci> <cn> 2 </cn> </apply> <cn> 1 </cn> </apply> </apply> </apply> <apply> <diff/> <bvar> <ci> x </ci> </bvar> <ci> u </ci> </apply> </apply>
</apply> </math>  <math> <apply> <eq/> <apply> <int/> <bvar> <ci> x </ci> </bvar> <apply> <divide/> <cn> 1 </cn> <apply> <times/> <ci> x </ci>
<apply> <root/> <apply> <fn> <ci> &plusminus; </ci> </fn> <apply> <power/> <ci> a </ci> <cn> 2 </cn> </apply> <apply> <power/> <ci> x </ci> <cn>
2 </cn> </apply> </apply> </apply> </apply> </apply> </apply> <apply> <minus/> <apply> <times/> <apply> <divide/> <cn> 1 </cn> <ci> a </ci>
</apply> <apply> <log/> <apply> <divide/> <apply> <plus/> <ci> a </ci> <apply> <root/> <apply> <fn> <ci> &plusminus; </ci> </fn> <apply> <power/>
<ci> a </ci> <cn> 2 </cn> </apply> <apply> <power/> <ci> x </ci> <cn> 2 </cn> </apply> </apply> </apply> </apply> <ci> x </ci> </apply> </apply>
</apply> </apply> </apply> </math> <math> <apply> <eq/> <apply> <int/> <bvar> <ci> x </ci> </bvar> <apply> <divide/> <cn> 1 </cn> <apply> <plus/>
<ci> a </ci> <apply> <times/> <ci> b </ci> <apply> <power/> <ci> x </ci> <cn> 2 </cn> </apply> </apply> </apply> </apply> </apply> <apply> <or/>
<apply> <times/> <apply> <divide/> <cn> 1 </cn> <apply> <times/> <cn> 2 </cn> <apply> <root/> <apply> <minus/> <apply> <times/> <ci> a </ci> <ci>
b </ci> </apply> </apply> </apply> </apply> </apply> <apply> <log/> <apply> <divide/> <apply> <plus/> <ci> a </ci> <apply> <times/> <ci> x </ci>
<apply> <root/> <apply> <minus/> <apply> <times/> <ci> a </ci> <ci> b </ci> </apply> </apply> </apply> </apply> </apply> <apply> <minus/> <ci> a
</ci> <apply> <times/> <ci> x </ci> <apply> <root/> <apply> <minus/> <apply> <times/> <ci> a </ci> <ci> b </ci> </apply> </apply> </apply> </apply>
</apply> </apply> </apply> </apply> <apply> <times/> <apply> <divide/> <cn> 1 </cn> <apply> <root/> <apply> <minus/> <apply> <times/> <ci> a </ci>
<ci> b </ci> </apply> </apply> </apply> </apply> <apply> <power/> <apply> <tanh/> <apply> <divide/> <apply> <times/> <ci> x </ci> <apply> <root/>
<apply> <minus/> <apply> <times/> <ci> a </ci> <ci> b </ci> </apply> </apply> </apply> </apply> <ci> a </ci> </apply> </apply> <apply> <minus/> <cn>
1 </cn> </apply> </apply> </apply> </apply> </apply> </math> <math> <apply> <eq/> <apply> <int/> <bvar> <ci> x </ci> </bvar> <apply> <divide/>
<cn> 1 </cn> <apply> <times/> <apply> <cos/> <apply> <times/> <ci> a </ci> <ci> x </ci> </apply> </apply> <apply> <fn> <ci> &plusminus; </ci> </fn>
<cn> 1 </cn> <apply> <sin/> <apply> <times/> <ci> a </ci> <ci> x </ci> </apply> </apply> </apply> </apply> </apply> </apply> <apply> <plus/> <apply>
<fn> <ci> &minusplus; </ci> </fn> <apply> <divide/> <cn> 1 </cn> <apply> <times/> <cn> 2 </cn> <ci> a </ci> <apply> <fn> <ci> &plusminus; </ci> </fn>
<cn> 1 </cn> <apply> <sin/> <apply> <times/> <ci> a </ci> <ci> x </ci> </apply> </apply> </apply> </apply> </apply> </apply> <apply> <times/> <apply>
<divide/> <cn> 1 </cn> <apply> <times/> <cn> 2 </cn> <ci> a </ci> </apply> </apply> <apply> <log/> <apply> <tan/> <apply> <plus/> <apply> <divide/>
<ci> &pi; </ci> <cn> 4 </cn> </apply> <apply> <divide/> <apply> <times/> <ci> a </ci> <ci> x </ci> </apply> <cn> 2 </cn> </apply> </apply> </apply>
</apply> </apply> </apply> </apply> </math>  <math> <apply> <eq/> <apply> <plus/> <cn> 1 </cn> <apply> <minus/> <apply> <divide/> <cn> 1 </cn>
<cn> 3 </cn> </apply> </apply> <apply> <divide/> <cn> 1 </cn> <cn> 5 </cn> </apply> <apply> <minus/> <apply> <divide/> <cn> 1 </cn> <cn> 7 </cn>
</apply> </apply> <ci> &cdots; </ci> </apply> <apply> <divide/> <ci> &pi; </ci> <cn> 4 </cn> </apply> </apply> </math> <math> <apply> <eq/> <apply>
<plus/> <cn> 1 </cn> <apply> <divide/> <cn> 1 </cn> <apply> <power/> <cn> 2 </cn> <cn> 2 </cn> </apply> </apply> <apply> <divide/> <cn> 1 </cn>
<apply> <power/> <cn> 3 </cn> <cn> 2 </cn> </apply> </apply> <apply> <divide/> <cn> 1 </cn> <apply> <power/> <cn> 4 </cn> <cn> 2 </cn> </apply>
</apply> <ci> &cdots; </ci> </apply> <apply> <divide/> <apply> <power/> <ci> &pi; </ci> <cn> 2 </cn> </apply> <cn> 6 </cn> </apply> </apply> </math>
<math> <apply> <eq/> <apply> <plus/> <cn> 1 </cn> <apply> <minus/> <apply> <divide/> <cn> 1 </cn> <apply> <power/> <cn> 2 </cn> <cn> 2 </cn>
</apply> </apply> </apply> <apply> <divide/> <cn> 1 </cn> <apply> <power/> <cn> 3 </cn> <cn> 2 </cn> </apply> </apply> <apply> <minus/> <apply>
<divide/> <cn> 1 </cn> <apply> <power/> <cn> 4 </cn> <cn> 2 </cn> </apply> </apply> </apply> <ci> &cdots; </ci> </apply> <apply> <divide/> <apply>
<power/> <ci> &pi; </ci> <cn> 2 </cn> </apply> <cn> 12 </cn> </apply> </apply> </math> <math> <apply> <forall/> <condition> <apply> <in/> <ci> x
</ci> <ci> &reals; </ci> </apply> </condition> <apply> <eq/> <apply> <power/> <ci> &exponentiale; </ci> <ci> x </ci> </apply> <apply> <plus/> <cn> 1
</cn> <ci> x </ci> <apply> <divide/> <apply> <power/> <ci> x </ci> <cn> 2 </cn> </apply> <apply> <factorial/> <cn> 2 </cn> </apply> </apply> <apply>
<divide/> <apply> <power/> <ci> x </ci> <cn> 3 </cn> </apply> <apply> <factorial/> <cn> 3 </cn> </apply> </apply> <ci> &cdots; </ci> <apply> <divide/>
<apply> <power/> <ci> x </ci> <ci> n </ci> </apply> <apply> <factorial/> <ci> n </ci> </apply> </apply> <ci> &cdots; </ci> </apply> </apply> </apply>
</math> <math> <apply> <forall/> <condition> <apply> <in/> <ci> x </ci> <ci> &reals; </ci> </apply> </condition> <apply> <eq/> <apply> <power/>
<ci> &exponentiale; </ci> <apply> <minus/> <ci> x </ci> </apply> </apply> <apply> <plus/> <cn> 1 </cn> <apply> <minus/> <ci> x </ci> </apply>
<apply> <divide/> <apply> <power/> <ci> x </ci> <cn> 2 </cn> </apply> <apply> <factorial/> <cn> 2 </cn> </apply> </apply> <apply> <minus/> <apply>
<divide/> <apply> <power/> <ci> x </ci> <cn> 3 </cn> </apply> <apply> <factorial/> <cn> 3 </cn> </apply> </apply> </apply> <ci> &cdots; </ci> <apply>
```

Hans Hagen
Hasselt, January 2001 / June 2008 / June 2011
www.pragma-ade.com

More changes and additions can be expected when there is a definitive version of the MathML 3 specification and more correct testsuite. One thing we need to look into is the nesting model dealing with () discussed in the spec.

copyright : PRAGMA ADE, Hasselt, NL
version : June 8, 2011
renderer : version 1 / mkiv

Table of Contents

Introduction

It is a well known fact that TEX can do a pretty good job on typesetting math. This is one reason why many scientific articles, papers and books are typeset using TEX. However, in these days of triumphing angle brackets, coding in TEX looks more and more out of place.

From the point of view of an author, coding in TEX is quite natural, given that some time is spent on reading the manuals. This is because not only the natural flow of the definition suits the way mathematicians think, but also because the author has quite some control over the way his thoughts end up on paper. It will be no surprise that switching to a more restricted way of coding, which also demands more keystrokes, is not on forehand considered to be better.

There are however circumstances that one wants to share formulas (or formula–like specifications) between several applications, one of which is a typesetting engine. In that case, a bit more work now, later saves you some headaches due to keeping the different source documents in sync.

The moment coding math in xml is discussed, those in favour stress that coding can be eased by using appropriate editors. Here we encounter a dilemma. For optimal usage, one should code in terms of content, that is, the principles that are expressed in a formula. Editors are not that strong in this area, and if they would be, editing would be not that much different from traditionally editing formulas: just keying in ideas using code that at first sight looks obscure. A more graphical oriented editor can help authors to compose formulas, but the underlaying coding will mainly be in terms of placing glyphs and boxes, and as a result the code will hardly be usable in other applications.

So either we code in terms of concepts, which permits sharing code among applications, and poses strong limitations on the influence of authors on the visual appearance. Or we use an interactive editor to fine tune the appearance of a formula and take for granted that reuse will be minimal or suboptimal.

In the following chapters we will discuss the mathematical language MathML in the perspective of typography. As a typesetting vehicle, we have used ConTEXt. However, the principles introduced here and the examples that we provide are independent of ConTEXt. For a more formal exploration we recommend the MathML specification.

This document is dedicated to all those ConTEXt users who like typesetting math. I'm sure that my father, who was a math teacher, would have liked proofreading this document. His absence was compensated by Tobias Burnus, Wang Lei, Ton Otten, and members of the ConTEXt mailing list who carefully read the text, corrected the errors in my math, tested the functionality, and made suggestions. Any remaining errors are mine.

This version is produced by ConTEXt MkIV *and is also used as testcase. The* MathML *processing code will be cleaned up which can occasionally result in suboptimal rendering.*

What is MATHML

<>

Backgrounds

MathML showed up in the evolving vacuum between structural sgml markup and presentational html. Both sgml and html can be recognized by angle brackets. The disadvantage of sgml was that it was so open ended, that general tools could hardly be developed. html on the other hand was easy to use and became extremely popular and users as well as software vendors quickly spoiled the original ideas and created a mess. sgml never became really popular, but thanks to html people became accustomed to that kind of notation. So, when xml came around as a more restricted cousin of sgml, the world was kind of ready for it. It cannot be denied that by some clever marketing many of today's users think that they use something new and modern, while we are actually dealing with something from the early days of computing. A main benefit of xml is that it brought the ideas behind sgml (and medium neutral coding in general) to the users and at the same time made a major cleanup of html possible.

About the same time, MathML was defined, both to bring math to the www, and to provide a way of coding math that will stimulate sharing the same code between different applications. At the end of 2000, the MathML version 2 draft became a recommendation. In the process of rewriting the interpreter for ConTeXt MkIV mid 2008 a draft of MathML version 3 has been used.

Now, imagine that we want to present a document on the internet using a format like html, either for viewing or for being spoken. Converting text and graphics is, given proper source coding, seldom a problem, but converting formulas into some angle bracket representation is more tricky. A way out of this is MathML's presentational markup.

$$a + b = c$$

This simple formula, when coded in TeX, looks like:

```
$$ a + b = c $$
```

In presentational MathML we get:

```
<math xmlns="http://www.w3c.org/mathml" version="2.0">
  <mrow>
    <mi> a </mi>
    <mo> + </mo>
    <mi> b </mi>
    <mo> = </mo>
    <mi> c </mi>
  </mrow>
```

```
</math>
```

In presentational MathML, we use mostly begintags (`<mi>`) and end tags (`</mi>`). The *row* element is the basic building block of a formula. The *mi* element specifies a math identifier and *mo* is used for operators. In the process of typesetting, both are subjected to interpretation in order to get the best visualization.

Converting TEX code directly or indirectly, using the dvi output or even in–memory produced math lists, has been one of the driving forces behind presentational MathML and other math related dtd's like EuroMath. One may wonder if there are sound and valid reasons for going the opposite way. You can imagine that a converter from TEX to MathML produces *menclose*, *mspace*, *mstyle* and other elements that can have many spacing related attributes, but I wonder if any author is willing to think in those quantities. Visual editors of course are good candidates for producing presentational MathML.

But wouldn't it be more efficient if we could express ideas and concepts in such a way that they could be handled by a broad range of applications, including a typesetting engine? This is why, in addition to presentational MathML, there is also content MathML. The previous formula, when coded in such a way, looks like:

```
<math xmlns="http://www.w3c.org/mathml" version="2.0">
  <apply> <eq/>
    <apply> <plus/>
      <ci> a </ci>
      <ci> b </ci>
    </apply>
    <ci> c </ci>
  </apply>
</math>
```

This way of defining a formula resembles the so called polish (or stackwise) notation. Opposite to presentational markup, here a typesetting engine has to find out in what order and what way the content has to be presented. This may seem a disadvantage, but in practice implementing content markup is not that complicated. The big advantage is that, once we know how to typeset a concept, TEX can do a good job, while in presentational markup much hard coded spacing can spoil everything. One can of course ignore specific elements, but it is more safe to start from less and enhance, than to leave away something with unknown quantities.

Instead of using hard coded operators as in presentational MathML, content markup uses empty elements like `<plus/>`. Many operators and functions are predefined but one can also define his own; in MathML 3 this is further extended by adopting OpenMath as variant.

Of course the main question to be answered now is to what extent the author can influence the appearance of a formula defined in content markup. Content markup has the advantage that the results can be more consistent, but taking away all control is counterproductive. The MathML level 2 draft mentions that this level covers most of the pre university math. If so, that is a proper starting point, but especially educational math often has to be typeset in such ways that it serves its purpose. Also, (re)using the formulas in other applications (simulators and alike) is useful in an educational setting, so content markup is quite suitable.

How do we combine the advantages of content markup with the wish of an author to control the visual output and at the same time get an as high as possible typeset result. There are several ways to accomplish this. One is to include in the document source both the content markup and the TEX specific code.

```
<math xmlns="http://www.w3c.org/mathml" version="2.0">
  <semantics>
    <apply> <eq/>
      <apply> <plus/>
        <ci> a </ci>
        <ci> b </ci>
      </apply>
    </apply>
    <ci> c </ci>
    <annotation encoding="TeX">a+b=c</annotation>
  </semantics>
</math>
```

The *annotation* element is one of the few that is permitted inside the *math* element. In this example, we embed pure TEX code, which, when enabled is typeset in math mode. It will be clear that for a simple formula like this one, such redundant coding is not needed, but one can imagine more complicated formulas. Because we want to limit the amount of work, we prefer just content markup.

Remark: Some characters, fillers or whatever may not show up. This is due to the fact that the relevant tables for ConTEXt MkIV *are defined stepwise. In due time most relevant symbols will be accessible.*

<> *Two methods* The best way to learn MathML is to key in formulas, so that is what

we did as soon as we started adding MathML support to ConTEXt. In some areas, MathML provides much detail (many functions are represented by elements) while in other areas one has to fall back on the more generic function element or a full description. Compare the following definitions:

```
<document>
  <math xmlns="http://www.w3c.org/mathml" version="2.0">
    <apply> <sin/> <ci> x </ci> </apply>
  </math>
  <math xmlns="http://www.w3c.org/mathml" version="2.0">
    <mrow> <mi> sin </mi> <mi> x </mi> </mrow>
  </math>
</document>
```

We prefer the first definition because it is more structured and gives more control over the result. There is only one ‘unknown’ quantity, x, and from the encapsulating element *ci* we know that it is an identifier.

$$\sin x$$

$$sinx$$

In the content example, from the *apply sin* we can deduce that the following argument is an operand, either an *apply*, or a *ci* or *cn*. In the presentational alternative, the following elements can be braces, a math identifier, a row, a sequence of identifiers and operators, etc. There, the look and feel is hard coded.

```
<?context-mathml-directive function reduction no ?>
```

This directive, either issued in the xml file, or set in the style file, changes the appearance of the function, but only in content markup. It is because of this feature, that we favour content markup.

$$\sin (x)$$

$$sinx$$

Does this mean that we can cover everything with content markup? The answer to this is still unclear. Consider the following definition.

$$\int \left(\frac{1}{\cos (ax) (1 \pm \sin (ax))}\right) \mathrm{d}x = \left(\frac{1}{2a (1 \pm \sin (ax))}\right) + \frac{1}{2a} \log \tan \left(\frac{\pi}{4} + \frac{ax}{2}\right)$$

Here we combine several cases in one formula by using ± and ∓ symbols. Because we only have *plus* and *minus* elements, we have to revert to the generic function element *fn*. We show the complete definition of this formula.

```
<math xmlns='http://www.w3c.org/mathml' version='2.0'>
  <apply> <eq/>
    <apply> <int/>
      <bvar> <ci> x </ci> </bvar>
      <apply> <divide/>
        <cn> 1 </cn>
        <apply> <times/>
          <apply> <cos/>
            <apply> <times/>
              <ci> a </ci>
              <ci> x </ci>
            </apply>
          </apply>
          <apply> <fn> <ci> &plusminus; </ci> </fn>
            <cn> 1 </cn>
            <apply> <sin/>
              <apply> <times/>
                <ci> a </ci>
                <ci> x </ci>
              </apply>
            </apply>
          </apply>
        </apply>
      </apply>
    </apply>
    <apply> <plus/>
      <apply> <fn> <ci> &minusplus; </ci> </fn>
        <apply> <divide/>
          <cn> 1 </cn>
          <apply> <times/>
            <cn> 2 </cn>
            <ci> a </ci>
            <apply> <fn> <ci> &plusminus; </ci> </fn>
              <cn> 1 </cn>
              <apply> <sin/>
                <apply> <times/>
                  <ci> a </ci>
                  <ci> x </ci>
                </apply>
              </apply>
            </apply>
          </apply>
        </apply>
      </apply>
      <apply> <times/>
        <apply> <divide/>
          <cn> 1 </cn>
          <apply> <times/>
            <cn> 2 </cn>
            <ci> a </ci>
          </apply>
        </apply>
        <apply> <log/>
          <apply> <tan/>
            <apply> <plus/>
              <apply> <divide/>
                <ci> &pi; </ci>
                <cn> 4 </cn>
              </apply>
              <apply> <divide/>
                <apply> <times/>
                  <ci> a </ci>
                  <ci> x </ci>
                </apply>
                <cn> 2 </cn>
              </apply>
            </apply>
          </apply>
        </apply>
      </apply>
    </apply>
  </apply>
```

```
</math>
```

The MathML parser and typesetting engine have to know how to handle these special cases, because the visualization depends on the function (or operator). Here both composed signs are treated like the plus and minus signs, but in other cases an embraced argument may be needed. Each special case needs a specific handler.

Presentational markup

If a document contains presentational MathML, there is a good chance that the code is output by an editor. Here we will discuss the presentation elements that make sense for users when they want to manually code presentational MathML. In this chapter we show the default rendering, later we will discuss options.

Although much is permitted, we advise to keep the code as simple as possible, because then TeX can do a rather good job on interpreting and typesetting it. Just let TeX take care of the spacing.

<> *mi, mn, mo* Presentational markup comes down to pasting boxes together in math specific ways. The basic building blocks are these three character elements.

$$x = 5$$

```
<math xmlns="http://www.w3c.org/mathml" version="2.0">
  <mrow>
    <mi> x </mi> <mo> = </mo> <mn> 5 </mn>
  </mrow>
</math>
```

mi	identifier	normally typeset in an italic font
mn	number	normally typeset in a normal font
mo	operator	surrounded by specific spacing

Because numbers are taken from an upright font, special numbers are taken care of automatically. Here are some from the MathML specification:

2 0.123 0,000,000 2.1e10 0xFFeF MCMLXIX twentyone

```
<math xmlns="http://www.w3c.org/mathml" version="2.0">
  <mrow>
    <mn> 2          </mn> <mtext>&nbsp;&nbsp;</mtext>
    <mn> 0.123      </mn> <mtext>&nbsp;&nbsp;</mtext>
    <mn> 0,000,000  </mn> <mtext>&nbsp;&nbsp;</mtext>
    <mn> 2.1e10     </mn> <mtext>&nbsp;&nbsp;</mtext>
    <mn> 0xFFeF     </mn> <mtext>&nbsp;&nbsp;</mtext>
    <mn> MCMLXIX    </mn> <mtext>&nbsp;&nbsp;</mtext>
    <mn> twenty one </mn> <mtext>&nbsp;&nbsp;</mtext>
  </mrow>
```

```
</math>
```

Special characters can be accessed by their Unicode point or by a corresponding entity. For some reason there is quite some duplication in entities, but we don't bother too much about it because after all Unicode math (which has its own peculiarities) is the way to go. The specification has this somewhat strange formula definition:

$$2 + 3\mathrm{i}\frac{1}{2}\pi\mathrm{e}$$

```
<math xmlns="http://www.w3c.org/mathml" version="2.0">
  <mrow>
    <mn> 2 </mn>
    <mo> + </mo>
    <mrow>
      <mn> 3</mn>
      <mo> &InvisibleTimes; </mo>
      <mi> &ImaginaryI; </mi>
    </mrow>
  </mrow>
  <mfrac>
    <mn> 1 </mn>
    <mn> 2 </mn>
  </mfrac>
  <mi> &pi; </mi>
  <mi> &ExponentialE; </mi>
</math>
```

And:

$$\frac{\mathrm{d}}{\mathrm{d}x}$$

```
<math xmlns="http://www.w3c.org/mathml" version="2.0">
  <mfrac>
    <mo> &DifferentialD; </mo>
    <mrow>
      <mo> &DifferentialD; </mo>
      <mi> x </mi>
    </mrow>
  </mfrac>
```

```
</math>
```

Visualizing the *mo* element involved some heuristics. For instance the size of fences depends on what they fence. In the following case you see how we can influence this. For practical pusposes we only support size 1.

$$(x) \text{ or } (x) \text{ or } (\frac{1}{2})$$

```
<math xmlns="http://www.w3c.org/mathml" version="2.0">
  <mrow>
    <mo> ( </mo> <mi> x </mi>  <mo> ) </mo>
  </mrow>
  <mtext> or </mtext>
  <mrow>
    <mo maxsize="1"> ( </mo>  <mi> x </mi>  <mo> ) </mo>
  </mrow>
  <mtext> or </mtext>
  <mrow>
    <mo maxsize="1"     > ( </mo>
        <mfrac> <mn> 1 </mn> <mn> 2 </mn> </mfrac>
    <mo stretchy="false"> ) </mo>
  </mrow>
</math>
```

<> *mrow* The previous example demonstrated the use of *mrow*, the element that is used to communicate the larger building blocks. Although this element from the perspective of typesetting is not always needed, by using it, the structure of the formula in the document source is more clear. There is some messy magic going on when we try to fake fenced expressions.

$$x \geq 2$$

```
<math xmlns="http://www.w3c.org/mathml" version="2.0">
  <mrow> <mi> x </mi> <mo> &geq; </mo> <mn> 2 </mn> </mrow>
</math>
```

$$y > 4$$

```
<math xmlns="http://www.w3c.org/mathml" version="2.0">
  <mrow>
```

```
    <mi> y </mi> <mo> &gt; </mo> <mn> 4 </mn>
  </mrow>
</math>
```

$$<x>$$

```
<math xmlns="http://www.w3c.org/mathml" version="2.0">
  <mrow>
    <mo> &lt; </mo> <mi> x </mi> <mo> &gt; </mo>
  </mrow>
</math>
```

$$a<b<c$$

```
<math xmlns="http://www.w3c.org/mathml" version="2.0">
  <mrow>
    <mi> a </mi> <mo> &lt; </mo> <mi> b </mi> <mo> &lt; </mo> <mi> c </mi>
  </mrow>
</math>
```

Spacing between a sign and the following token is taken care of automatically by TEX:

$$-1-1$$

```
<math xmlns="http://www.w3c.org/mathml" version="2.0">
  <mrow>
    <mo> - </mo>
    <mn> 1 </mn>
    <mo> - </mo>
    <mn> 1 </mn>
  </mrow>
</math>
```

<> ***msup, msub, msubsup*** Where in content markup super and subscript are absent and derived from the context, in presentational markup they are quite present.

$$x_1{}^2$$

```
<math xmlns="http://www.w3c.org/mathml" version="2.0">
  <msup>
    <msub> <mi> x </mi> <mn> 1 </mn> </msub>
    <mn> 2 </mn>
  </msup>
</math>
```

$$x_1^2$$

```
<math xmlns="http://www.w3c.org/mathml" version="2.0">
  <msubsup>
    <mi> x </mi>
    <mn> 1 </mn>
    <mn> 2 </mn>
  </msubsup>
</math>
```

Watch the difference between both definitions and appearances. You can influence the default behaviour with processing instructions.

<> *mfrac* Addition, subtraction and multiplication is hard coded using the *mo* element with $+$, $-$, and $\times$ (or nothing). You can use / for division, but for more complicated formulas you have to fall back on fraction building. This is why MathML provides the *mfrac*.

$$\frac{x+1}{y+1}$$

```
<math xmlns="http://www.w3c.org/mathml" version="2.0">
  <mfrac>
    <mrow> <mi> x </mi> <mo> + </mo> <mn> 1 </mn> </mrow>
    <mrow> <mi> y </mi> <mo> + </mo> <mn> 1 </mn> </mrow>
  </mfrac>
</math>
```

You can change the width of the rule, but this is generally a bad idea. For special purposes you can set the line thickness to zero.

$$\genfrac{}{}{0pt}{}{x \geq 2}{y \leq 4}$$

```
<math xmlns="http://www.w3c.org/mathml" version="2.0">
  <mfrac linethickness="0">
    <mrow> <mi> x </mi> <mo> &geq; </mo> <mn> 2 </mn> </mrow>
    <mrow> <mi> y </mi> <mo> &leq; </mo> <mn> 4 </mn> </mrow>
  </mfrac>
</math>
```

A different kind of rendering is also possible, as shown in the following example.

$$\frac{x}{2}\Big/\frac{x}{2}$$

```
<math xmlns="http://www.w3c.org/mathml" version="2.0">
  <mfrac bevelled="true">
    <mfrac>
      <mi> x </mi> <mn> 2 </mn>
      <mi> y </mi> <mn> 4 </mn>
    </mfrac>
    <mfrac>
      <mi> x </mi> <mn> 2 </mn>
      <mi> y </mi> <mn> 4 </mn>
    </mfrac>
  </mfrac>
</math>
```

<>

mfenced Braces are used to visually group sub–expressions. In presentational MathML you can either hard code braces, or use the *mfenced* element to generate delimiters automatically. In ConTEXt, as much as possible, the operators and identifiers are interpreted, and when recognized treated according to their nature.

$$(a, b, 1)$$

```
<math xmlns="http://www.w3c.org/mathml" version="2.0">
  <mfenced> <mi> a </mi> <mi> b </mi> <mn> 1 </mn> </mfenced>
</math>
```

The fencing symbols adapt their size to the content. Their dimensions also depend on the way math fonts are defined. The standard TEX fonts will give the same height of braces around x and y, but in other fonts the y may invoke slightly larger ones.

$$[0, 1)$$

```
<math xmlns="http://www.w3c.org/mathml" version="2.0">
  <mfenced open="[" close=")" separators=",">
    <mn> 0 </mn> <mn> 1 </mn>
  </mfenced>
</math>
```

The separators adapt their size to the fenced content too, just like the fences.

$$\left[\frac{1}{x}\middle|\frac{1}{y}\middle|\frac{1}{z}\right]$$

```
<math xmlns="http://www.w3c.org/mathml" version="2.0">
  <mfenced open="[" close="]" separators="|">
    <mfrac> <mn> 1 </mn> <mi> x </mi> </mfrac>
    <mfrac> <mn> 1 </mn> <mi> y </mi> </mfrac>
    <mfrac> <mn> 1 </mn> <mi> z </mi> </mfrac>
  </mfenced>
</math>
```

$$(1+x)$$

```
<math xmlns="http://www.w3c.org/mathml" version="2.0">
  <mfenced>
    <mrow> <mn> 1 </mn> <mo> + </mo> <mi> x </mi> </mrow>
  </mfenced>
</math>
```

$$\{1|2+3-4$$

```
<math xmlns="http://www.w3c.org/mathml" version="2.0">
  <mfenced open="{" close="" separators="|+-">
    <mn> 1 </mn> <mn> 2 </mn> <mn> 3 </mn> <mn> 4 </mn>
  </mfenced>
</math>
```

$$a1b2c3d4e$$

```
<math xmlns="http://www.w3c.org/mathml" version="2.0">
  <mfenced open="a" close="e" separators="bcd">
    <mn> 1 </mn> <mn> 2 </mn> <mn> 3 </mn> <mn> 4 </mn>
  </mfenced>
</math>
```

<> *msqrt, mroot* The shape and size of roots, integrals, sums and products can depend on the size of the content.

$$\sqrt{b}$$

```
<math xmlns="http://www.w3c.org/mathml" version="2.0">
  <msqrt>
    <mi> b </mi>
  </msqrt>
</math>
```

$$\sqrt[2]{b}$$

```
<math xmlns="http://www.w3c.org/mathml" version="2.0">
  <mroot>
    <mi> b </mi>
    <mn> 2 </mn>
  </mroot>
</math>
```

$$\sqrt[2]{\frac{1}{b}}$$

```
<math xmlns="http://www.w3c.org/mathml" version="2.0">
  <mroot>
    <mfrac> <mn> 1 </mn> <mi> b </mi> </mfrac>
    <mn> 2 </mn>
  </mroot>
</math>
```

$$\sqrt[3]{\frac{1}{a+b}}$$

```
<math xmlns="http://www.w3c.org/mathml" version="2.0">
  <mroot>
    <mfrac>
      <mn> 1 </mn>
      <mrow> <mi> a </mi> <mo> + </mo> <mi> b </mi> </mrow>
    </mfrac>
    <mn> 3 </mn>
  </mroot>
</math>
```

<> **mtext** If you put text in a *mi* element, it will come out rather ugly. This is due to the fact that identifiers are (at least in TEX) not subjected to the kerning that is normally used in text. Therefore, when you want to add some text to a formula, you should use the *mtext* element.

$$\frac{SomeText}{\text{Some Text}}$$

```
<math xmlns="http://www.w3c.org/mathml" version="2.0">
  <mfrac>
    <mi> Some Text </mi>
    <mtext> Some Text </mtext>
  </mfrac>
</math>
```

As with all elements, leading and trailing spaces are ignored. If you really want a space in front or at the end, you should use one of the space tokens other than the ascii spacing tokens. You can also use entities like `&nbsp;`.

<> **mover, munder, munderover** Not all formulas are math and spacing and font rules may differ per discipline. The following formula reflects a chemical reaction.

$$2\ \mathrm{H}_{2} + \ \mathrm{O}_{2} \underset{\text{explosion}}{\longrightarrow} 2\ \mathrm{H}_{2}\ \mathrm{O}$$

```
<math xmlns="http://www.w3c.org/mathml" version="2.0">
  <mrow>
    <mrow>
      <mn> 2 </mn>
      <msub> <mtext> H </mtext> <mn> 2 </mn> </msub>
    </mrow>
    <mo> + </mo>
    <msub> <mtext> O </mtext> <mn> 2 </mn> </msub>
    <munder>
      <mo> &RightArrow; </mo>
      <mtext> explosion </mtext>
    </munder>
    <mrow>
      <mn> 2 </mn>
      <msub> <mtext> H </mtext> <mn> 2 </mn> </msub>
      <mtext> O </mtext>
    </mrow>
  </mrow>
</math>
```

The *munder*, *mover* and *munderover* elements can be used to put symbols and text or formulas on top of each other. When applicable, the symbols will stretch themselves to span the natural size of the text or formula.

The following examples demonstrate how the relevant components of this threesome are defined.

$$x \underset{\text{maps to}}{\longrightarrow} y$$

```
<math xmlns="http://www.w3c.org/mathml" version="2.0">
  <mrow>
    <mi> x </mi>
    <munder>
      <mo> &RightArrow; </mo>
      <mtext> maps to </mtext>
    </munder>
    <mi> y </mi>
  </mrow>
</math>
```

$$x \underset{\longrightarrow}{\text{maps to}} y$$

```
<math xmlns="http://www.w3c.org/mathml" version="2.0">
  <mrow>
    <mi> x </mi>
    <munder>
      <mtext> maps to </mtext>
      <mo> &RightArrow; </mo>
    </munder>
    <mi> y </mi>
  </mrow>
</math>
```

$$x \overset{\longrightarrow}{\text{maps to}} y$$

```
<math xmlns="http://www.w3c.org/mathml" version="2.0">
```

```
  <mrow>
    <mi> x </mi>
    <mover>
      <mtext> maps to </mtext>
      <mo> &RightArrow; </mo>
    </mover>
    <mi> y </mi>
  </mrow>
</math>
```

$$x \xrightarrow{\text{maps to}} y$$

```
<math xmlns="http://www.w3c.org/mathml" version="2.0">
  <mrow>
    <mi> x </mi>
    <mover>
      <mo> &RightArrow; </mo>
      <mtext> maps to </mtext>
    </mover>
    <mi> y </mi>
  </mrow>
</math>
```

$$\int_1^\infty$$

```
<math xmlns="http://www.w3c.org/mathml" version="2.0">
  <mrow>
    <munderover>
      <mi> &int; </mi>
      <mn> 1 </mn>
      <mi> &infin; </mi>
    </munderover>
  </mrow>
</math>
```

$$x$$

```
<math xmlns="http://www.w3c.org/mathml" version="2.0">
    <mover>
        <mi> x </mi>
        <mo> &Hat; </mo>
    </mover>
</math>
```

<> *ms* This is a bit weird element. It behaves like *mtext* but puts quotes around the text.

$$\frac{\text{“ Some Text ”}}{\text{Some Text}}$$

```
<math xmlns="http://www.w3c.org/mathml" version="2.0">
  <mfrac>
    <ms> Some Text </ms>
    <mtext> Some Text </mtext>
  </mfrac>
</math>
```

You can specify the left and right boundary characters, either directly or (preferably) using entities like `&quot;`.

$$\text{+ A Famous Quotation +}$$

```
<math xmlns="http://www.w3c.org/mathml" version="2.0">
  <ms lquote="+" rquote="+"> A Famous Quotation </ms>
</math>
```

<> *menclose* This element is implemented but it is such a weird element that it's probably seldom used.

$$\overline{)123}$$

```
<math xmlns="http://www.w3c.org/mathml" version="2.0">
  <menclose notation="longdiv"><mn>123</mn></menclose>
</math>
```

$$\overline{123\,|}$$

```
<math xmlns="http://www.w3c.org/mathml" version="2.0">
  <menclose notation="actuarial"><mn>123</mn></menclose>
</math>
```

$$\sqrt{123}$$

```
<math xmlns="http://www.w3c.org/mathml" version="2.0">
  <menclose notation="radical"><mn>123</mn></menclose>
</math>
```

A bit more complex example (taken from the specification) demonstrates where those somewhat strange rendering options are good for:

$$\begin{array}{rr} & 10 \\ 131 & \overline{)1413} \\ & 131 \\ & 103 \end{array}$$

```
<math xmlns="http://www.w3c.org/mathml" version="2.0">
  <mtable columnspacing="0pt" rowspacing="0pt">
    <mtr>
      <mtd></mtd>
      <mtd columnalign="right"><mn>10</mn></mtd>
    </mtr>
    <mtr>
      <mtd columnalign="right"><mn>131</mn></mtd>
      <mtd columnalign="right">
        <menclose notation="longdiv"><mn>1413</mn></menclose>
      </mtd>
    </mtr>
    <mtr>
      <mtd></mtd>
      <mtd columnalign="right">
        <mrow>
          <munder>
            <mn>131</mn>
            <mo>&UnderBar;</mo>
          </munder>
          <mphantom><mn>3</mn></mphantom>
        </mrow>
      </mtd>
    </mtr>
    <mtr>
```

```
      <mtd></mtd>
      <mtd columnalign="right"><mn>103</mn></mtd>
    </mtr>
  </mtable>
</math>
```

In MathML 3 a few more notations showed up and to some extend we support them. We assume that the previously mentioned variants are always applied to the content first.

$$\enclose{box,downdiagonalstrike}{\text{whatever}}$$

```
<math xmlns="http://www.w3c.org/mathml" version="2.0">
  <menclose notation="box downdiagonalstrike">
    <mtext>whatever</mtext>
  </menclose>
</math>
```

$$\enclose{roundedbox,updiagonalstrike}{\text{whatever}}$$

```
<math xmlns="http://www.w3c.org/mathml" version="2.0">
  <menclose notation="roundedbox updiagonalstrike">
    <mtext>whatever</mtext>
  </menclose>
</math>
```

$$\enclose{circle,verticalstrike,horizontalstrike}{\text{whatever}}$$

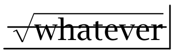
```
<math xmlns="http://www.w3c.org/mathml" version="2.0">
  <menclose notation="circle verticalstrike horizontalstrike">
    <mtext>whatever</mtext>
  </menclose>
</math>
```

$$\enclose{left,top,verticalstrike}{\text{whatever}}$$

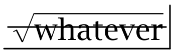
```
<math xmlns="http://www.w3c.org/mathml" version="2.0">
  <menclose notation="left top verticalstrike">
    <mtext>whatever</mtext>
  </menclose>
</math>
```

$$\enclose{right,bottom,horizontalstrike}{\text{whatever}}$$

```
<math xmlns="http://www.w3c.org/mathml" version="2.0">
  <menclose notation="right bottom horizontalstrike">
    <mtext>whatever</mtext>
  </menclose>
</math>
```



```
<math xmlns="http://www.w3c.org/mathml" version="2.0">
  <menclose notation="radical right bottom horizontalstrike">
    <mtext>whatever</mtext>
  </menclose>
</math>
```



```
<math xmlns="http://www.w3c.org/mathml" version="2.0">
  <menclose notation="right bottom horizontalstrike radical">
    <mtext>whatever</mtext>
  </menclose>
</math>
```

The graphics are drawn at runtime by METAPOST. Currently we don't combine them into one which would be more efficient in terms of output (not so much in runtime). You can define additional variants; as an example we show one of the solutions:

```
\startuseMPgraphic{mml:enclose:box}
  draw OverlayBox
    withpen pencircle scaled (ExHeight/10) ;
\stopuseMPgraphic

\defineoverlay [mml:enclose:box] [\useMPgraphic{mml:enclose:box}]
```

You can roll out your own:

```
\startuseMPgraphic{mml:enclose:mybox}
  draw OverlayBox enlarged (ExHeight/5)
    withpen pencircle scaled (ExHeight/10) ;
\stopuseMPgraphic

\defineoverlay [mml:enclose:mybox] [\useMPgraphic{mml:enclose:mybox}]
```

$$\boxed{\text{whatever}}$$

```
<math xmlns="http://www.w3c.org/mathml" version="2.0">
  <menclose notation="mybox">
    <mtext>whatever</mtext>
  </menclose>
</math>
```

<> ***merror*** There is not much chance that this element will end up in a math textbook, unless the typeset output of programs is part of the story.

$$\text{Are you kidding?}\;\; \frac{1+x}{0}$$

```
<math xmlns="http://www.w3c.org/mathml" version="2.0">
  <merror>
    <mtext> Are you kidding? &ThickSpace; </mtext>
    <mfrac>
      <mrow> <mn> 1 </mn> <mo> + </mo> <mi> x </mi> </mrow>
      <mn> 0 </mn>
    </mfrac>
  </merror>
</math>
```

<> ***mmultiscripts, mprescripts*** This element is one of the less obvious ones. The next two examples are taken from the specification. The *multiscripts* element takes an odd number of arguments. The second and successive child elements alternate between sub- and superscript. The empty element *none* —a dedicated element *mnone* would have been a better choice— serves as a placeholder.

$$R_i{}^j{}_{kl}$$

```
<math xmlns="http://www.w3c.org/mathml" version="2.0">
  <mmultiscripts>
    <mi> R </mi>
    <mi> i </mi>
    <none/>
    <none/>
    <mi> j </mi>
    <mi> k </mi>
    <none/>
    <mi> l </mi>
    <none/>
  </mmultiscripts>
</math>
```

The *mmultiscripts* element can also be used to attach prescripts to a symbol. The next example demonstrates this. The empty *prescripts* element signals the start of the prescripts section.

$${}_{427}Qb_4$$

```
<math xmlns="http://www.w3c.org/mathml" version="2.0">
  <mmultiscripts>
    <mi> Qb </mi>
    <mn> 4 </mn>
    <none/>
    <mprescripts/>
    <mn> 427 </mn>
    <none/>
  </mmultiscripts>
</math>
```

<> ***mspace*** Currently not all functionality of the *mspace* element is implemented. Over time we will see what support is needed and makes sense, especially since this command can spoil things. We only support the units that make sense, so units in terms of pixels —a rather persistent oversight in drafts— are kindly ignored.

use   me   with   care

```
<math xmlns="http://www.w3c.org/mathml" version="2.0">
  <mrow>
    <mtext> use  </mtext> <mspace width="1em" />
    <mtext> me   </mtext> <mspace width="1ex" />
    <mtext> with </mtext> <mspace width="10pt"/>
    <mtext> care </mtext>
  </mrow>
</math>
```

You can also pass a sample text:

$$\frac{44}{\frac{112233}{11\qquad 33}}$$

```
<math xmlns="http://www.w3c.org/mathml" version="2.0">
  <mfrac>
     <mi> 44 </mi>
     <mfrac>
       <mrow>
         <mn> 11 </mn> <mn> 22 </mn> <mn> 33 </mn>
       </mrow>
       <mrow>
         <mn> 11 </mn> <mspace spacing="22"/> <mn> 33 </mn>
       </mrow>
     </mfrac>
  </mfrac>
</math>
```

<> *mphantom* A phantom element hides its content but still takes its space. A phantom element can contain other elements.

who is afraid of   elements

```
<math xmlns="http://www.w3c.org/mathml" version="2.0">
  <mrow>
    <mtext>    who is afraid of </mtext>    <mspace width=".5em" />
    <mphantom> phantom          </mphantom> <mspace width=".5em" />
    <mtext>    elements         </mtext>
  </mrow>
</math>
```

<> *mpadded* As with a few other elements, we first have to see some practical usage for this, before we could implement the functionality needed.

<> *mtable, mtr, mtd, mlabeledtr* As soon as you want to represent a matrix or other more complicated composed constructs, you end up with spacing problems. This is when tables come into view. Because presentational elements have no deep knowledge

about their content, tables made with presentational MathML will in most cases look worse than those that result from content markup.

We have implemented tables on top of the normal xml (html) based table support in ConTeXt, also known as natural tables. Depending on the needs, support for the *mtable* element will be extended.

The *mtable* element takes a lot of attributes. When no attributes are given, we assume that a matrix is wanted, and typeset the content accordingly.

$$\begin{pmatrix} x_{1,1} & 1 & 0 \\ 0 & x_{2,2} & 1 \\ 0 & 1 & x_{3,3} \end{pmatrix}$$

```
<math xmlns="http://www.w3c.org/mathml" version="2.0">
  <mrow>
    <mo> ( </mo>
    <mtable>
      <mtr>
        <mtd> <msub> <mi> x </mi> <mn> 1,1 </mn> </msub> </mtd>
        <mtd> <mn> 1 </mn> </mtd>
        <mtd> <mn> 0 </mn> </mtd>
      </mtr>
      <mtr>
        <mtd> <mn> 0 </mn> </mtd>
        <mtd> <msub> <mi> x </mi> <mn> 2,2 </mn> </msub> </mtd>
        <mtd> <mn> 1 </mn> </mtd>
      </mtr>
      <mtr>
        <mtd> <mn> 0 </mn> </mtd>
        <mtd> <mn> 1 </mn> </mtd>
        <mtd> <msub> <mi> x </mi> <mn> 3,3 </mn> </msub> </mtd>
      </mtr>
    </mtable>
    <mo> ) </mo>
  </mrow>
</math>
```

$$\begin{array}{lcr} \boxed{100} & 100 & 100 \\ 10 & \boxed{10} & 10 \\ 1 & 1 & \boxed{1} \end{array}$$

```
<math xmlns="http://www.w3c.org/mathml" version="2.0">
  <mtable columnalign="left center right">
    <mtr>
      <mtd frame="solid"> <mn> 100 </mn> </mtd>
      <mtd              > <mn> 100 </mn> </mtd>
      <mtd              > <mn> 100 </mn> </mtd>
    </mtr>
    <mtr>
      <mtd              > <mn> 10  </mn> </mtd>
      <mtd frame="solid"> <mn> 10  </mn> </mtd>
      <mtd              > <mn> 10  </mn> </mtd>
    </mtr>
    <mtr>
      <mtd              > <mn> 1   </mn> </mtd>
      <mtd              > <mn> 1   </mn> </mtd>
      <mtd frame="solid"> <mn> 1   </mn> </mtd>
    </mtr>
  </mtable>
</math>
```

A special case is the labeled row *mlabeledtr*. This one is meant for numbering equations. However, in a properly formatted document there is probably some encapsulating structure that takes care of this. Therefore we discard the first child element. We show an example taken from the specification.

$$E = mc^2$$

```
<math xmlns="http://www.w3c.org/mathml" version="2.0">
  <mtable>
    <mlabeledtr>
      <mtd>crap</mtd>
      <mtd>
        <mrow>
          <mi>E</mi>
          <mo>=</mo>
          <mrow>
            <mi>m</mi>
            <mi>&it;</mi>
            <msup>
```

```
              <mi>c</mi>
              <mn>2</mn>
            </msup>
          </mrow>
        </mrow>
      </mtd>
    </mlabeledtr>
  </mtable>
</math>
```

Although the underlying table mechanism can provide all the support needed (and even more), not all attributes are yet implemented. We will make a useful selection.

columnalign	keyword: left center (middle) right
columnspacing	a meaningful dimension
rowspacing	a meaningful dimension
frame	keyword: none (off) solid (on)
color	a named color identifier
background	a named color identifier

We only support properly named colors as back- and foreground colors. The normal ConTEXt color mapping mechanism can be used to remap colors. This permits (read: forces) a consistent usage of colors. If you use named backgrounds ... the sky is the limit.

<> *mcolumn* This element is new in MathML 3 and is kind of special in the sense that the content is analyzed. It would have made more sense just to provide some proper structure instead since it's intended use is rather well defined.

Because it is not much fun to implement such a messy element we only support it partially and add what comes on our way. Here are a few examples (more or less taken from the reference).

$$\begin{array}{r} 12 \\ \times 12 \\ \hline 24 \\ 12 \\ \hline 144 \end{array}$$

```
<math xmlns="http://www.w3c.org/mathml" version="2.0">
```

```
  <mcolumn>
    <mn>12</mn>
    <mrow> <mo>&times;</mo> <mn>12</mn> </mrow>
    <mline spacing="000"/>
    <mn>24</mn>
    <mrow> <mn>12</mn> <mspace spacing="0"/> </mrow>
    <mline spacing="000"/>
    <mn>144</mn>
  </mcolumn>
</math>
```

$$\begin{array}{l}123\\ 456+\\ \hline 579\end{array}$$

```
<math xmlns="http://www.w3c.org/mathml" version="2.0">
  <mcolumn>
    <mn>123</mn>
    <mrow> <mn>456</mn> <mo>+</mo> </mrow>
    <mline spacing="000+"/>
    <mn>579</mn>
  </mcolumn>
</math>
```

$$\begin{array}{l}1{,}23\\ 4{,}56+\\ \hline 5{,}79\end{array}$$

```
<math xmlns="http://www.w3c.org/mathml" version="2.0">
  <mcolumn>
    <mn>1,23</mn>
    <mrow> <mn>4,56</mn> <mo>+</mo> </mrow>
    <mline spacing="0,00+"/>
    <mn>5,79</mn>
  </mcolumn>
</math>
```

$$\begin{array}{r}52\\ -7\\ \hline\end{array}$$

45

```
<math xmlns="http://www.w3c.org/mathml" version="2.0">
  <mcolumn>
    <mstyle mathsize="71%">
        <menclose notation="bottom"> <mn>10</mn> </menclose>
    </mstyle>
    <mn>52</mn>
    <mrow> <mo>&minus;</mo> <mn>7</mn> </mrow>
    <mline spacing="45"/>
    <mn>45</mn>
  </mcolumn>
</math>
```

Similar effects can be accomplished with the *mtable* element.

<> **malignmark** This element is used in tables and is not yet implemented, first because I still have to unravel its exact usage, but second, because it is about the ugliest piece of MathML markup you will encounter.

<> **mglyph** This element is for those who want to violate the ideas of general markup by popping in his or her own glyphs. Of course one should use entities, even if they have to be defined.

$$\mathrm{A} + \mathrm{B} = \mathrm{C}$$

```
<math xmlns="http://www.w3c.org/mathml" version="2.0">
  <mrow>
    <mi> <mglyph fontfamily="Serif" index="65" alt="The Letter A"/></mi>
    <mo> + </mo>
    <mi> <mglyph fontfamily="Serif" index="66" alt="The Letter B"/></mi>
    <mo> = </mo>
    <mi> <mglyph fontfamily="Serif" index="67" alt="The Letter C"/></mi>
  </mrow>
</math>
```

<> **mstyle** This element is implemented but not yet discussed since we want more control over its misuse.

<> *afterword* You may have noticed that we prefer content MathML over presentational MathML. So, unless you’re already tired of any math coded in angle brackets, we invite you to read the next chapter too.

Content markup

In this chapter we will discuss the MathML elements from the point of view of typesetting. We will not pay attention to other rendering techniques, like speech generation. Some elements take attributes and those often make more sense for other applications than for a typesetting engine like TEX, which has a strong math engine that knows how to handle math.

One of the most prominent changes in MathML 3 is support for an OpenMath like coding. Here the *csymbol* takes the place of the empty element as first argument of an *apply*. There are more symbols in OpenMath then we supported in the interpreter, but in due time (depending on demand) we will add more. At the time of writing this the draft was really a draft which made it hard to grasp all the implications for rendering so we probably need to overhaul the code sometime in the future.

Another change is the usage of *apply* that has been delegated to *bind*. One may wonder why this hadn't happen before. For the moment we treat the *bind* as if it were an *apply*.

<> *apply* If you are dealing with rather ordinary math, you will only need a subset of content MathML. For this reason we will start with the most common elements. When you key in xml directly, you will encounter the *apply* element quite often, even in a relatively short formula like the following.

$$-1$$

```
<math xmlns="http://www.w3c.org/mathml" version="2.0">
    <apply> <minus/>
        <cn> 1 </cn>
    </apply>
</math>
```

In most cases the *apply* element is followed by a specification disguised as an empty element.

Later we will see more complex examples but here we already show the different ways of encoding. First we show the traditional MathML 2 method:

$$\forall_x : x \geq 4$$

```
<math xmlns="http://www.w3c.org/mathml" version="2.0">
    <apply> <forall/>
        <bvar> <ci>x</ci> </bvar>
        <apply> <geq/>
```

```
            <ci>x</ci>
            <cn>4</cn>
        </apply>
    </apply>
</math>
```

This is now called ‘pragmatic’ MathML. Using symbols and *bind* this becomes ‘strict’ MathML:

$$\forall_x : x \geq 4$$

```
<math xmlns="http://www.w3c.org/mathml" version="3.0">
    <bind> <csymbol cd="quant1">forall</csymbol>
        <bvar> <ci>x</ci> </bvar>
        <apply> <csymbol cd="relation1">geq</csymbol>
            <ci>x</ci>
            <cn>4</cn>
        </apply>
    </bind>
</math>
```

<> *ci, cn, sep* These elements are used to specify identifiers and numbers. Both elements can be made more explicit by using attributes.

`type`	`set`	use a representation appropriate for sets
	`vector`	mark this element as vector
	`function`	consider this element to be a function
	`fn`	idem

When *set* is specified, a blackboard symbol is used when available.

$$x \in \mathbb{N}$$

```
<math xmlns="http://www.w3c.org/mathml" version="2.0">
  <apply> <in/>
    <ci> x </ci>
    <ci type="set"> N </ci>
  </apply>
</math>
```

The *function* specification makes sense when the *ci* element is used in for instance a differential equation.

type	integer	a whole number with an optional base
	logical	a boolean constant
	rational	a real number
	complex-cartesian	a complex number in $x + iy$ notation
	complex	idem
	complex-polar	a complex number in polar notation ...

You're lucky when your document uses decimal notation, otherwise you will end up with long specs if you want to be clear in what numbers are used.

$$1A2C_{16} + 0101_{16} = 1B2D_{16}$$

```
<math xmlns="http://www.w3c.org/mathml" version="2.0">
  <apply> <eq/>
    <apply> <plus/>
      <cn type="integer" base="16"> 1A2C </cn>
      <cn type="integer" base="16"> 0101 </cn>
    </apply>
    <cn type="integer" base="16"> 1B2D </cn>
  </apply>
</math>
```

Complex numbers have two components. These are separated by the *sep* element. In the following example we see that instead of using a *ci* with set specifier, the empty element *complexes* can be used. We will see some more of those later.

$$(2 + 5\,\mathrm{i}) \in \mathbb{C}$$

```
<math xmlns="http://www.w3c.org/mathml" version="2.0">
  <apply> <in/>
    <cn type="complex"> 2 <sep/> 5 </cn>
    <complexes/>
  </apply>
</math>
```

<> ***eq, neq, gt, lt, geq, leq*** Expressions, and especially those with *eq* are typical for math. Because such expressions can be quite large, there are provisions for proper alignment.

lt	$a < b$	leq	$a \leq b$
eq	$a = b$	neq	$a \neq b$
gt	$a > b$	geq	$a \geq b$

$$a \leq b \leq c$$

```
<math xmlns="http://www.w3c.org/mathml" version="2.0">
  <apply> <leq/>
    <ci> a </ci>
    <ci> b </ci>
    <ci> c </ci>
  </apply>
</math>
```

<> *equivalent, approx, implies* Equivalence, approximations, and implications are handled like *eq* and alike and have their own symbols.

$$a + b \equiv b + a$$

```
<math xmlns="http://www.w3c.org/mathml" version="2.0">
  <apply> <equivalent/>
    <apply> <plus/> <ci> a </ci> <ci> b </ci> </apply>
    <apply> <plus/> <ci> b </ci> <ci> a </ci> </apply>
  </apply>
</math>
```

This document is typeset with LuaTeX built upon TeX version 3.14159, and given that TeX is written by a mathematician, it will be no surprise that:

$$3.14159 \approx \pi$$

```
<math xmlns="http://www.w3c.org/mathml" version="2.0">
  <apply> <approx/>
    <cn> 3.14159 </cn>
    <pi/>
  </apply>
</math>
```

$$x + 4 = 9 \Rightarrow x = 5$$

```
<math xmlns="http://www.w3c.org/mathml" version="2.0">
  <apply> <implies/>
    <apply> <eq/>
      <apply> <plus/>
        <ci> x </ci>
        <cn> 4 </cn>
      </apply>
      <cn> 9 </cn>
    </apply>
    <apply> <eq/>
      <ci> x </ci>
      <cn> 5 </cn>
    </apply>
  </apply>
</math>
```

<> *minus, plus* Addition and subtraction are main building blocks of math so you will meet them often.

$$37 - x$$

```
<math xmlns="http://www.w3c.org/mathml" version="2.0">
  <apply> <minus/>
    <cn> 37 </cn>
    <ci> x </ci>
  </apply>
</math>
```

In most cases there will be more than one argument to take care of, but especially *minus* will be used with one argument too. Although `<cn> -37 </cn>` is valid, using *minus* is sometimes more clear.

$$-37$$

```
<math xmlns="http://www.w3c.org/mathml" version="2.0">
  <apply> <minus/>
    <cn> 37 </cn>
  </apply>
</math>
```

You should pay attention to combinations of *plus* and *minus*. Opposite to presentational MathML, in content markup you don't think and code sequential.

$$-x + 37$$

```
<math xmlns="http://www.w3c.org/mathml" version="2.0">
  <apply> <plus/>
    <apply> <minus/>
      <ci> x </ci>
    </apply>
    <cn> 37 </cn>
  </apply>
</math>
```

In MathML 3 we can also be more vebose:

$$a + x$$

```
<math xmlns="http://www.w3c.org/mathml" version="3.0">
    <apply> <csymbol cd="arith1">plus</csymbol>
        <ci>a</ci>
        <ci>x</ci>
    </apply>
</math>
```

<>

times Multiplication is another top ten element. Although `3p` as content of the *ci* element would have rendered the next example as well, you really should split off the number and mark it as *cn*. When this is done consistently, we can comfortably change the font of numbers independent of the font used for displaying identifiers.

$$3p$$

```
<math xmlns="http://www.w3c.org/mathml" version="2.0">
  <apply> <times/>
    <cn> 3 </cn>
    <ci> p </ci>
  </apply>
</math>
```

In a following chapter we will see how we can add multiplication signs between variables and constants.

<> *divide* When typeset, a division is characterized by a horizontal rule. Some elements, like the differential element *diff*, generate their own division.

$$1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \dots = \frac{\pi}{4}$$

This example also demonstrates how to mix *plus* and *minus*.

```
<math xmlns='http://www.w3c.org/mathml' version='2.0'>
  <apply> <eq/>
    <apply> <plus/>
      <cn> 1 </cn>
      <apply> <minus/>
        <apply> <divide/>
          <cn> 1 </cn>
          <cn> 3 </cn>
        </apply>
      </apply>
      <apply> <divide/>
        <cn> 1 </cn>
        <cn> 5 </cn>
      </apply>
      <apply> <minus/>
        <apply> <divide/>
          <cn> 1 </cn>
          <cn> 7 </cn>
        </apply>
      </apply>
      <ci> &cdots; </ci>
    </apply>
    <apply> <divide/>
      <ci> &pi; </ci>
      <cn> 4 </cn>
    </apply>
 </apply>
</math>
```

$$\frac{-b - -b - \sqrt{a}}{(b - b) - -b - \sqrt{a}}$$

```
<math xmlns="http://www.w3c.org/mathml" version="2.0">
```

```
    <apply><divide/>
        <apply><minus/>
            <apply><minus/><ci>b</ci></apply>
            <apply><minus/><ci>b</ci></apply>
            <apply><root/> <ci>a</ci></apply>
        </apply>
        <apply><minus/>
            <apply><minus/><ci>b</ci><ci>b</ci></apply>
            <apply><minus/><ci>b</ci></apply>
            <apply><root/> <ci>a</ci></apply>
        </apply>
    </apply>
</math>
```

<> *power* In presentational MathML you think in super- and subscripts, but in content MathML these elements are not available. There you need to think in terms of *power*.

$$x^2 + \sin^2 x$$

```
<math xmlns="http://www.w3c.org/mathml" version="2.0">
  <apply> <plus/>
    <apply> <power/>
      <ci> x </ci>
      <cn> 2 </cn>
    </apply>
    <apply> <power/>
      <apply> <sin/>
        <ci> x </ci>
      </apply>
      <cn> 2 </cn>
    </apply>
  </apply>
</math>
```

The *power* element is clever enough to determine where the superscript should go. In the case of the sinus function, by default it will go after the function identifier.

<> *root, degree* If you study math related dtd's —this are the formal descriptions for sgml or xml element collections— you will notice that there are not that many elements that

demand a special kind of typography: differential equations, limits, integrals and roots are the most distinctive ones.

$$\sqrt[3]{64} = 4$$

```
<math xmlns="http://www.w3c.org/mathml" version="2.0">
  <apply> <eq/>
    <apply> <root/>
      <degree> 3 </degree>
      <ci> 64 </ci>
    </apply>
    <cn> 4 </cn>
  </apply>
</math>
```

Contrary to *power*, the *root* element uses a specialized child element to denote the degree. The positive consequence of this is that there cannot be a misunderstanding about what role the child element plays, while in for instance *power* you need to know that the second child element denotes the degree.

<> *sin, cos, tan, cot, scs, sec, . . .* All members of the family of goniometric functions are available as empty element. When needed, their argument is surrounded by braces. They all behave the same.

sin	arcsin	sinh	arcsinh
cos	arccos	cosh	arccosh
tan	arctan	tanh	arctanh
cot	arccot	coth	arccoth
csc	arccsc	csch	arccsch
sec	arcsec	sech	arcsech

These functions are normally typeset in a non italic (often roman) font shape.

$$\sin(x+y) = \sin x \cos y + \cos x \sin y$$

By default the typesetting engine will minimize the number of braces that surrounds the argument of a function.

```
<math xmlns='http://www.w3c.org/mathml' version='2.0'>
  <apply> <eq/>
```

```
    <apply> <sin/>
      <apply> <plus/>
        <ci> x </ci>
        <ci> y </ci>
      </apply>
    </apply>
    <apply> <plus/>
      <apply> <times/>
        <apply> <sin/>
          <ci> x </ci>
        </apply>
        <apply> <cos/>
          <ci> y </ci>
        </apply>
      </apply>
      <apply> <times/>
        <apply> <cos/>
          <ci> x </ci>
        </apply>
        <apply> <sin/>
          <ci> y </ci>
        </apply>
      </apply>
    </apply>
  </apply>
</math>
```

You can specify π as an entity `&pi;` or as empty element *pi*. In many cases it is up to your taste which one you use. There are many symbols that are only available as entity, so in some respect there is no real reason to treat π different.

$$\cos \pi = -1$$

```
<math xmlns="http://www.w3c.org/mathml" version="2.0">
  <apply> <eq/>
    <apply> <cos/>
      <pi/>
    </apply>
    <apply> <minus/>
```

```
      <cn> 1 </cn>
    </apply>
  </apply>
</math>
```

<> ***log, ln, exp*** The *log* and *ln* are typeset similar to the previously discussed goniometric functions. The *exp* element is a special case of *power*. The constant *e* can be specified with *exponentiale*.

$$\ln\left(\mathrm{e}+2\right) \approx 1.55$$

```
<math xmlns="http://www.w3c.org/mathml" version="2.0">
  <apply> <approx/>
    <apply> <ln/>
      <apply> <plus/>
        <exponentiale/>
        <cn> 2 </cn>
      </apply>
    </apply>
    <cn> 1.55 </cn>
  </apply>
</math>
```

$$\mathrm{e}^{2} = 7.3890560989307$$

```
<math xmlns="http://www.w3c.org/mathml" version="2.0">
  <apply> <eq/>
    <apply> <exp/>
      <cn> 2 </cn>
    </apply>
    <cn> 7.3890560989307 </cn>
  </apply>
</math>
```

<> ***quotient, rem*** The result of a division can be a rational number, so $\frac{5}{4}$ is equivalent to 1.25 and 1.25×4 gives 5. An integer division will give 1 with a remainder 2. Many computer languages provide a `div` and `mod` function, and since MathML is also meant for computation, it provides similar concepts, represented by the elements *quotient* and *rem*. The

representation of *quotient* is rather undefined, but the next one is among the recommended alternatives.

$$\lfloor a \,/\, b \rfloor$$

```
<math xmlns="http://www.w3c.org/mathml" version="2.0">
  <apply> <quotient/>
    <ci> a </ci>
    <ci> b </ci>
  </apply>
</math>
```

<> *factorial* Showing the representation of a factorial is rather dull, so we will use a few more elements as well as a processing instruction to illustrate the usage of *factorial*.

$$n! = n \times (n-1) \times (n-2) \times \cdots \times 1$$

```
<math xmlns="http://www.w3c.org/mathml" version="2.0">
  <?context-mathml-directive times symbol yes ?>
  <apply> <eq/>
    <apply> <factorial/>
      <ci> n </ci>
    </apply>
    <apply> <times/>
      <ci> n </ci>
      <apply> <minus/> <ci> n </ci> <cn> 1 </cn> </apply>
      <apply> <minus/> <ci> n </ci> <cn> 2 </cn> </apply>
      <csymbol definitionUrl="cdots"/>
      <cn> 1 </cn>
    </apply>
  </apply>
</math>
```

The processing instruction is responsible for the placement of the × symbols.

<> *min, max, gcd, lcm* These functions can handle more than two arguments. When typeset, these are separated by commas.

$$z = \min \left\{ (x+y)\,, 2x, \frac{1}{y} \right\}$$

```
<math xmlns="http://www.w3c.org/mathml" version="2.0">
  <apply> <eq/>
    <ci> z </ci>
    <apply> <min/>
      <apply> <plus/>   <ci> x </ci> <ci> y </ci> </apply>
      <apply> <times/>  <cn> 2 </cn> <ci> x </ci> </apply>
      <apply> <divide/> <cn> 1 </cn> <ci> y </ci> </apply>
    </apply>
  </apply>
</math>
```

<> ***and, or, xor, not*** Logical expressions can be defined using these elements. The operations are represented by symbols and braces are applied when needed.

$$1001_2 \wedge 0101_2 = 0001_2$$

```
<math xmlns="http://www.w3c.org/mathml" version="2.0">
  <apply> <eq/>
    <apply> <and/>
      <cn type="integer" base="2"> 1001 </cn>
      <cn type="integer" base="2"> 0101 </cn>
    </apply>
    <cn type="integer" base="2"> 0001 </cn>
  </apply>
</math>
```

<> ***set, bvar*** The appearance of a *set* depends on the presence of the child element *bvar*. In its simplest form, a set is represented as a list.

$$\{1,4,8\} \neq$$

```
<math xmlns="http://www.w3c.org/mathml" version="2.0">
  <apply> <neq/>
    <set>
      <cn> 1 </cn>
      <cn> 4 </cn>
      <cn> 8 </cn>
    </set>
    <emptyset/>
  </apply>
</math>
```

A set can be distinguished from a vector by its curly braces. The simplest case is just a comma separated list. The next example demonstrates the declarative case. Without doubt, there will be other alternatives.

$$\{x \mid 2 < x < 8\}$$

```
<math xmlns="http://www.w3c.org/mathml" version="2.0">
  <set>
    <bvar><ci> x </ci></bvar>
    <condition>
      <apply> <lt/>
        <cn> 2 </cn>
        <ci> x </ci>
        <cn> 8 </cn>
      </apply>
    </condition>
  </set>
</math>
```

<> *list* This element is used in different contexts. When used as a top level element, a list is typeset as follows.

$$[1, 1, 3]$$

```
<math xmlns="http://www.w3c.org/mathml" version="2.0">
  <list>
    <cn> 1 </cn>
    <cn> 1 </cn>
    <cn> 3 </cn>
  </list>
</math>
```

When used in a context like *partialdiff*, the list specification becomes a subscript.

$$\mathrm{D}_{1,1,3} f$$

```
<math xmlns="http://www.w3c.org/mathml" version="2.0">
  <apply> <partialdiff/>
```

```
    <list>
      <cn> 1 </cn>
      <cn> 1 </cn>
      <cn> 3 </cn>
    </list>
    <ci type="fn"> f </ci>
  </apply>
</math>
```

The function specification in this formula (which is taken from the specs) can also be specified as `<fn> <ci> f </ci> </fn>` (which is more clear).

<> *union, intersect, . . .* There is a large number of set operators, each represented by a distinctive symbol.

union	$U \cup V$		
intersect	$U \cap V$		
in	$U \in V$	notin	$\not{U}\in V$
subset	$U \subset V$	notsubset	$\not{U}\subset V$
prsubset	$U \subseteq V$	notprsubset	$\not{U}\subseteq V$
setdiff	$U \setminus V$		

These operators are applied as follows:

$$U \cup V$$

```
<math xmlns="http://www.w3c.org/mathml" version="2.0">
  <apply> <union/>
    <ci> U </ci>
    <ci> V </ci>
  </apply>
</math>
```

<> *conjugate, arg, real, imaginary* The visual representation of *conjugate* is a horizontal bar with a width matching the width of the expression.

$$\overline{x + \mathrm{i}y}$$

```
<math xmlns="http://www.w3c.org/mathml" version="2.0">
```

```
  <apply> <conjugate/>
    <apply> <plus/>
      <ci> x </ci>
      <apply> <times/>
        <cn> &ImaginaryI; </cn>
        <ci> y </ci>
      </apply>
    </apply>
  </apply>
</math>
```

The *arg*, *real* and *imaginary* elements trigger the following appearance.

$$\arg\left(x+\mathrm{i}y\right)$$

$$\Re\left(x+\mathrm{i}y\right)$$

$$\mathrm{i}$$

<> **abs, floor, ceiling** There are a couple of functions that turn numbers into positive or rounded ones. In computer languages names are used, but in math we use special boundary characters.

$$|-5| = 5$$

```
<math xmlns="http://www.w3c.org/mathml" version="2.0">
  <apply> <eq/>
    <apply> <abs/> <cn> -5 </cn> </apply>
    <cn> 5 </cn>
  </apply>
</math>
```

$$\lfloor 5.5 \rfloor = 5$$

```
<math xmlns="http://www.w3c.org/mathml" version="2.0">
  <apply> <eq/>
    <apply> <floor/> <cn> 5.5 </cn> </apply>
    <cn> 5 </cn>
  </apply>
</math>
```

$$\lceil 5.5 \rceil = 6$$

```
<math xmlns="http://www.w3c.org/mathml" version="2.0">
  <apply> <eq/>
    <apply> <ceiling/> <cn> 5.5 </cn> </apply>
    <cn> 6 </cn>
  </apply>
</math>
```

<> *interval* An interval is visualized as: $[1,10]$. The *interval* element is a container element and has a begin and endtag. You can specify the closure as attribute:

$$(a,b]$$

```
<math xmlns="http://www.w3c.org/mathml" version="2.0">
  <interval closure="open-closed">
    <ci> a </ci>
    <ci> b </ci>
  </interval>
</math>
```

The following closures are supported:

open	(a,b)
closed	$[a,b]$
open-closed	$(a,b]$
closed-open	$[a,b)$

In strict MathML we use symbols instead of attributes to define the openess:

$$(a,x)$$

```
<math xmlns="http://www.w3c.org/mathml" version="3.0">
    <apply> <csymbol cd="interval1">interval_oo</csymbol>
        <ci>a</ci>
        <ci>x</ci>
    </apply>
</math>
```

$$[a,x]$$

```
<math xmlns="http://www.w3c.org/mathml" version="3.0">
```

```
    <apply> <csymbol cd="interval1">interval_cc</csymbol>
        <ci>a</ci>
        <ci>x</ci>
    </apply>
</math>
```

<> *inverse* This operator is applied to a function. The following example demonstrates that this is one of the few cases (if not the only one) where the first element following an *apply* begintag is an *apply* itself.

$$\sin^{-1} x$$

```
<math xmlns="http://www.w3c.org/mathml" version="2.0">
  <apply>
    <apply> <inverse/> <sin/> </apply>
    <ci> x </ci>
  </apply>
</math>
```

<> *reln* This element is a left-over from the first MathML specification and its usage is no longer advocated. Its current functionality matches the functionality of *apply*.

<> *cartesianproduct, vectorproduct, scalarproduct, outerproduct* The context of the formula will often provide information of what kind of multiplication is meant, but using different symbols to represent the kind of product certainly helps.

$$a \times b$$

```
<math xmlns="http://www.w3c.org/mathml" version="2.0">
  <apply> <cartesianproduct/>
    <ci> a </ci>
    <ci> b </ci>
  </apply>
</math>
```

We have:

cartesian	$a \times b$
vector	$a \times b$

scalar	$a \cdot b$
outer	$a \otimes b$

<> *sum, product, limit, lowlimit, uplimit, bvar* Sums, products and limits have a distinctive look, especially when they have upper and lower limits attached. Unfortunately there is no way to specify the x_i in content MathML. In the next chapter we will see how we can handle that.

$$\sum_{i=1}^{n} \frac{1}{x}$$

```
<math xmlns="http://www.w3c.org/mathml" version="2.0">
  <apply> <sum/>
    <bvar> <ci> i </ci> </bvar>
    <lowlimit> <cn> 1 </cn> </lowlimit>
    <uplimit> <ci> n </ci> </uplimit>
    <apply> <divide/>
      <cn> 1 </cn>
      <ci> x </ci>
    </apply>
  </apply>
</math>
```

When we omit the limits, the *bvar* is still typeset.

$$\prod_{i} \frac{1}{x}$$

```
<math xmlns="http://www.w3c.org/mathml" version="2.0">
  <apply> <product/>
    <bvar>
      <ci> i </ci>
    </bvar>
    <apply> <divide/>
      <cn> 1 </cn>
      <ci> x </ci>
    </apply>
  </apply>
</math>
```

You can specify the condition under which the function is applied.

$$\prod_{x \in \mathbb{R}} f(x)$$

```
<math xmlns="http://www.w3c.org/mathml" version="2.0">
  <apply> <product/>
    <bvar>
      <ci> x </ci>
    </bvar>
    <condition>
      <apply> <in/>
        <ci> x </ci>
        <ci type="set"> R </ci>
      </apply>
    </condition>
    <apply> <ci type="fn"> f </ci>
      <ci> x </ci>
    </apply>
  </apply>
</math>
```

$$\lim_{x \to 0} \sin x$$

```
<math xmlns="http://www.w3c.org/mathml" version="2.0">
  <apply> <limit/>
    <bvar>
      <ci> x </ci>
    </bvar>
    <lowlimit>
      <cn> 0 </cn>
    </lowlimit>
    <apply> <sin/>
      <ci> x </ci>
    </apply>
  </apply>
</math>
```

<> *int, diff, partialdiff, bvar, degree* These elements reach a high level of abstraction. The best way to learn how to use them is to carefully study some examples.

$$\frac{\mathrm{d}\left(\int_{p}^{q} f(x,a)\,\mathrm{d}x\right)}{\mathrm{d}a}$$

```
<math xmlns="http://www.w3c.org/mathml" version="2.0">
  <apply> <diff/>
    <bvar> <ci> a </ci> </bvar>
    <apply> <int/>
      <lowlimit> <ci> p </ci> </lowlimit>
      <uplimit>  <ci> q </ci> </uplimit>
      <bvar> <ci> x </ci> </bvar>
      <apply>
        <fn> <ci> f </ci> </fn>
        <ci> x </ci>
        <ci> a </ci>
      </apply>
    </apply>
  </apply>
</math>
```

The *bvar* element is essential, since it is used to automatically generate some of the components that make up the visual appearance of the formula. If you look at the formal specification of these elements, you will notice that the appearance may depend on your definition. How the formula shows up, depends not only on the *bvar* element, but also on the optional *degree* element within.

$$f'$$

```
<math xmlns="http://www.w3c.org/mathml" version="2.0">
  <apply> <diff/>
    <ci> f </ci>
  </apply>
</math>
```

$$\frac{\mathrm{d}^2 f(x)}{\mathrm{d}x^2}$$

```
<math xmlns="http://www.w3c.org/mathml" version="2.0">
  <apply> <diff/>
    <bvar>
      <ci> x </ci>
      <degree> <cn> 2 </cn> </degree>
    </bvar>
    <apply> <fn> <ci> f </ci> </fn>
      <ci> x </ci>
    </apply>
  </apply>
</math>
```

$$\frac{\mathrm{d}^4 f}{x\,\mathrm{d}f^2}$$

```
<math xmlns="http://www.w3c.org/mathml" version="2.0">
  <apply> <partialdiff/>
    <bvar>
      <degree> <cn> 2 </cn> </degree>
      <ci> x </ci>
    </bvar>
    <bvar> <ci> y </ci> </bvar>
    <bvar> <ci> x </ci> </bvar>
    <degree> <cn> 4 </cn> </degree>
    <ci type="fn"> f </ci>
  </apply>
</math>
```

$$\frac{\mathrm{d}^k f\,(x,y)}{x\,\mathrm{d}f\,(x,y)^m}$$

```
<math xmlns="http://www.w3c.org/mathml" version="2.0">
  <apply> <partialdiff/>
    <bvar>
      <ci> x </ci> <degree> <ci> m </ci> </degree>
    </bvar>
    <bvar>
      <ci> y </ci> <degree> <ci> n </ci> </degree>
    </bvar>
    <degree> <ci> k </ci> </degree>
    <apply> <ci type="fn"> f </ci>
      <ci> x </ci>
      <ci> y </ci>
    </apply>
  </apply>
</math>
```

$$\frac{\mathrm{d}^{m+n} f\left(x,y\right)}{x\,\mathrm{d}f\left(x,y\right)^{m}}$$

```
<math xmlns="http://www.w3c.org/mathml" version="2.0">
  <apply> <partialdiff/>
    <bvar>
      <ci> x </ci> <degree> <ci> m </ci> </degree>
    </bvar>
    <bvar>
      <ci> y </ci> <degree> <ci> n </ci> </degree>
    </bvar>
    <apply> <ci type="fn"> f </ci>
      <ci> x </ci>
      <ci> y </ci>
    </apply>
  </apply>
</math>
```

When a degree is not specified, it is deduced from the context, but since this is not 100% watertight, you can best be complete in your specification.

These examples are taken from the MathML specification. In the example document that comes with this manual you can find a couple more.

<> *fn* There are a lot of predefined functions and operators. If you want to introduce a new one, the *fn* element can be used. In the following example we have turned the ± and ∓ symbols into (coupled) operators.

$$\left(x \pm 1\right)\left(x \mp 1\right) = x^{2} - 1$$

```
<math xmlns="http://www.w3c.org/mathml" version="2.0">
  <apply> <eq/>
    <apply> <times/>
      <apply> <fn> <ci> &plusminus; </ci> </fn>
        <ci> x </ci>
        <cn> 1 </cn>
      </apply>
      <apply> <fn> <ci> &minusplus; </ci> </fn>
        <ci> x </ci>
        <cn> 1 </cn>
      </apply>
    </apply>
    <apply> <minus/>
      <apply> <power/>
        <ci> x </ci>
        <cn> 2 </cn>
      </apply>
      <cn> 1 </cn>
    </apply>
  </apply>
</math>
```

The typeset result depends on the presence of a handler, which in this case happens to be true.

<> *matrix, matrixrow* A matrix is one of the building blocks of linear algebra and therefore both presentational and content MathML have dedicated elements for defining it.

$$\begin{pmatrix} 23 & 87 & c \\ 41 & b & 33 \\ a & 65 & 16 \end{pmatrix}$$

```
<math xmlns="http://www.w3c.org/mathml" version="2.0">
  <matrix>
    <matrixrow> <cn> 23 </cn> <cn> 87 </cn> <ci>  c </ci> </matrixrow>
    <matrixrow> <cn> 41 </cn> <ci>  b </ci> <cn> 33 </cn> </matrixrow>
    <matrixrow> <ci>  a </ci> <cn> 65 </cn> <cn> 16 </cn> </matrixrow>
  </matrix>
</math>
```

<> *vector* We make a difference between a vector specification and a vector variable. A specification is presented as a list:

$$(x, y)$$

```
<math xmlns="http://www.w3c.org/mathml" version="2.0">
```

```
  <vector>
    <ci> x </ci>
    <ci> y </ci>
  </vector>
</math>
```

When the *vector* element has one child element, we use a right arrow to identify the variable as vector.

$$\vec{A} \times \vec{B}$$

```
<math xmlns="http://www.w3c.org/mathml" version="2.0">
  <apply> <vectorproduct/>
    <vector> <ci> A </ci> </vector>
    <vector> <ci> B </ci> </vector>
  </apply>
</math>
```

<> *grad, curl, ident, divergence* These elements expand into named functions, but we can imagine that in the future a more appropriate visualization will be provided as an option.

$$\operatorname{grad} A \neq \operatorname{curl} B \neq \operatorname{identity} C \neq \operatorname{div} D$$

```
<math xmlns="http://www.w3c.org/mathml" version="2.0">
  <apply> <neq/>
    <apply> <grad/>       <ci> A </ci> </apply>
    <apply> <curl/>       <ci> B </ci> </apply>
    <apply> <ident/>      <ci> C </ci> </apply>
    <apply> <divergence/> <ci> D </ci> </apply>
  </apply>
</math>
```

<> *lambda, bvar* The lambda specification of a function needs a *bvar* element. The visualization can be influenced with processing instructions as described in a later chapter.

$$x \mapsto \sin\left(x - \frac{x}{2}\right)$$

```
<math xmlns="http://www.w3c.org/mathml" version="2.0">
```

```
  <lambda>
    <bvar> <ci> x </ci> </bvar>
    <apply> <sin/>
      <apply> <minus/>
        <ci> x </ci>
        <apply> <divide/>
          <ci> x </ci>
          <cn> 2 </cn>
        </apply>
      </apply>
    </apply>
  </lambda>
</math>
```

<> *piecewise, piece, otherwise* There are not so many elements that deal with combinations of formulas or conditions. The *piecewise* is the only real selector available. The following example defines how the state of n depends on the state of x.

$$n = \begin{cases} -1 & x < 0 \\ 1 & x > 0 \\ 0 & \text{otherwise} \end{cases}$$

```
<math xmlns="http://www.w3c.org/mathml" version="2.0">
  <apply> <eq/>
    <ci> n </ci>
    <piecewise>
      <piece>
        <apply> <minus/>
          <cn> 1 </cn>
        </apply>
        <apply> <lt/>
          <ci> x </ci>
          <cn> 0 </cn>
        </apply>
      </piece>
      <piece>
        <cn> 1 </cn>
        <apply> <gt/>
          <ci> x </ci>
          <cn> 0 </cn>
        </apply>
      </piece>
      <otherwise>
        <cn> 0 </cn>
      </otherwise>
    </piecewise>
  </apply>
</math>
```

We could have used a third *piece* instead of (optional) *otherwise*.

<> *forall, exists, condition* Conditions are often used in combination with elements like *forall*. There are several ways to convert and combine them in formulas and environments, so you may expect more alternatives in the future.

$$\forall_x\, x < 9 \mid x < 10$$

```
<math xmlns="http://www.w3c.org/mathml" version="2.0">
  <apply> <forall/>
    <bvar> <ci> x </ci> </bvar>
    <condition>
      <apply> <lt/>
        <ci> x </ci>
        <cn> 9 </cn>
      </apply>
    </condition>
    <apply> <lt/>
      <ci> x </ci>
      <cn> 10 </cn>
    </apply>
  </apply>
</math>
```

The next example is taken from the specifications with a few small changes.

$$\forall_x\, x \in \mathbb{N} \mid \exists_{p,q}\, p \in \mathbb{P} \wedge q \in \mathbb{P} \wedge p + q = 2x$$

```
<math xmlns="http://www.w3c.org/mathml" version="2.0">
  <apply> <forall/>
```

```
    <bvar> <ci> x </ci> </bvar>
    <condition>
      <apply> <in/>
        <ci> x </ci>
        <ci type="set"> N </ci>
      </apply>
    </condition>
    <apply> <exists/>
      <bvar> <ci> p </ci> </bvar>
      <bvar> <ci> q </ci> </bvar>
      <condition>
        <apply> <and/>
          <apply> <in/>
            <ci> p </ci>
            <ci type="set"> P </ci>
          </apply>
          <apply> <in/>
            <ci> q </ci>
            <ci type="set"> P </ci>
          </apply>
          <apply> <eq/>
            <apply> <plus/> <ci> p </ci> <ci> q </ci> </apply>
            <apply> <times/> <cn> 2 </cn> <ci> x </ci> </apply>
          </apply>
        </apply>
      </condition>
    </apply>
  </apply>
</math>
```

<> ***factorof, tendsto*** The *factorof* element is applied to its two child elements and contrary to most functions, the symbol is placed between the elements instead of in front.

$$a \mid b$$

```
<math xmlns="http://www.w3c.org/mathml" version="2.0">
  <apply> <factorof/>
    <ci> a </ci>
    <ci> b </ci>
  </apply>
</math>
```

The same is true for the *tendsto* element.

$$a \to b$$

```
<math xmlns="http://www.w3c.org/mathml" version="2.0">
  <apply> <tendsto/>
    <ci> a </ci>
    <ci> b </ci>
  </apply>
</math>
```

<> *compose* This is a nasty element since it has to take care of braces in special ways and therefore has to analyse its child elements.

$$(f \circ g \circ h)$$

```
<math xmlns="http://www.w3c.org/mathml" version="2.0">
  <apply> <compose/>
    <ci type="fn"> f </ci>
    <ci type="fn"> g </ci>
    <ci type="fn"> h </ci>
  </apply>
</math>
```

$$(f \circ g)\, x$$

```
<math xmlns="http://www.w3c.org/mathml" version="2.0">
  <apply>
    <apply> <compose/>
      <fn> <ci> f </ci> </fn>
      <fn> <ci> g </ci> </fn>
    </apply>
    <ci> x </ci>
  </apply>
</math>
```

<> *laplacian* A laplacian function is typeset using a ∇ (nabla) symbol.

$$\nabla^2 x$$

```
<math xmlns="http://www.w3c.org/mathml" version="2.0">
  <apply> <laplacian/>
    <ci> x </ci>
  </apply>
</math>
```

<> *mean, sdev, variance, median, mode* When statistics shows up in math text books, the *sum* element is likely to show up, probably in combination with the for statistics meaningful symbolic representation of variables. The mean value of a series of observations is defined as:

$$\overline{x} = \frac{\sum x}{n}$$

```
<math xmlns="http://www.w3c.org/mathml" version="2.0">
  <apply> <eq/>
    <apply> <mean/>
      <ci> x </ci>
    </apply>
    <apply> <divide/>
      <apply> <sum/>
        <ci> x </ci>
      </apply>
      <ci> n </ci>
    </apply>
  </apply>
</math>
```

or more beautiful:

$$\overline{x} = \frac{1}{n} \sum x$$

```
<math xmlns="http://www.w3c.org/mathml" version="2.0">
  <apply> <eq/>
    <apply> <mean/>
      <ci> x </ci>
    </apply>
    <apply> <times/>
      <apply> <divide/>
        <cn> 1 </cn>
        <ci> n </ci>
      </apply>
      <apply> <sum/>
        <ci> x </ci>
      </apply>
    </apply>
  </apply>
</math>
```

Of course this definition is not that perfect, but we will present a better alternative in the chapter on combined markup. The definition of the standard deviation is more complicated:

$$\sigma\left(x\right) \approx \sqrt{\frac{\sum\left(x - \overline{x}\right)^2}{n - 1}}$$

```
<math xmlns="http://www.w3c.org/mathml" version="2.0">
  <apply> <approx/>
    <apply> <sdev/>
      <ci> x </ci>
    </apply>
    <apply> <root/>
      <apply> <divide/>
        <apply> <sum/>
          <apply> <power/>
            <apply> <minus/>
              <ci> x </ci>
              <apply> <mean/>
                <ci> x </ci>
              </apply>
            </apply>
            <cn> 2 </cn>
          </apply>
        </apply>
        <apply> <minus/>
          <ci> n </ci>
          <cn> 1 </cn>
        </apply>
      </apply>
    </apply>
  </apply>
</math>
```

The next example demonstrates the usage of the *variance* in its own definition.

$$\sigma\,(x)^2 = \overline{(x - \overline{x})^2} \approx \frac{1}{n-1} \sum (x - \overline{x})^2$$

```
<math xmlns="http://www.w3c.org/mathml" version="2.0">
  <apply> <eq/>
    <apply> <variance/>
      <ci> x </ci>
    </apply>
    <apply> <approx/>
      <apply> <mean/>
        <apply> <power/>
          <apply> <minus/>
            <ci> x </ci>
            <apply> <mean/>
              <ci> x </ci>
            </apply>
          </apply>
          <cn> 2 </cn>
        </apply>
      </apply>
      <apply> <times/>
        <apply> <divide/>
          <cn> 1 </cn>
          <apply> <minus/>
            <ci> n </ci>
            <cn> 1 </cn>
          </apply>
        </apply>
        <apply> <sum/>
          <apply> <power/>
            <apply> <minus/>
```

```
              <ci> x </ci>
              <apply> <mean/>
                <ci> x </ci>
              </apply>
            </apply>
            <cn> 2 </cn>
          </apply>
        </apply>
      </apply>
    </apply>
  </apply>
</math>
```

The *median* and *mode* of a series of observations have no special symbols and are presented as is.

<> ***moment, momentabout, degree*** Because MathML is used for a wide range of applications, there can be information in a definition that does not end up in print but is only used in some cases. This is illustrated in the next example.

$$\left\langle X^3 \right\rangle$$

```
<math xmlns="http://www.w3c.org/mathml" version="2.0">
  <apply> <moment/>
    <degree>
      <cn> 3 </cn>
    </degree>
    <momentabout>
      <ci> p </ci>
    </momentabout>
    <ci> X </ci>
  </apply>
</math>
```

<> ***determinant, transpose*** These two (and the following) are used to manipulate matrices, either or not in a symbolic way. A simple determinant or transpose looks like:

$$|\boldsymbol{A}|$$

```
<math xmlns="http://www.w3c.org/mathml" version="2.0">
  <apply> <determinant/>
    <ci type="matrix"> A </ci>
  </apply>
</math>
```

$$\boldsymbol{A}^{\mathrm{T}}$$

```
<math xmlns="http://www.w3c.org/mathml" version="2.0">
  <apply> <transpose/>
    <ci type="matrix"> A </ci>
  </apply>
</math>
```

When the *determinant* element is applied to a full blown matrix, the braces are omitted and replaced by the vertical bars.

$$|I| = \begin{vmatrix} 1 & 0 \\ 0 & 1 \end{vmatrix} = 1$$

```
<math xmlns='http://www.w3c.org/mathml' version='2.0'>
  <apply> <eq/>
    <apply> <determinant/>
     <ci> I </ci>
    </apply>
    <apply> <determinant/>
      <matrix>
        <matrixrow> <cn> 1 </cn> <cn> 0 </cn> </matrixrow>
        <matrixrow> <cn> 0 </cn> <cn> 1 </cn> </matrixrow>
      </matrix>
    </apply>
    <cn> 1 </cn>
  </apply>
</math>
```

<> **selector** The *selector* element can be used to index a matrix cell or variable. This element honors the braces.

$$\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}_1$$

```
<math xmlns="http://www.w3c.org/mathml" version="2.0">
  <apply> <selector/>
    <matrix>
      <matrixrow> <cn> 1 </cn> <cn> 2 </cn> </matrixrow>
      <matrixrow> <cn> 3 </cn> <cn> 4 </cn> </matrixrow>
    </matrix>
    <cn> 1 </cn>
  </apply>
</math>
```

A more common usage of the selector is the following:

$$x_i$$

```
<math xmlns="http://www.w3c.org/mathml" version="2.0">
  <apply> <selector/>
    <ci> x </ci>
    <ci> i </ci>
  </apply>
</math>
```

It is possible to pass a comma separated list of indices:

$$x_{1,2}$$

```
<math xmlns="http://www.w3c.org/mathml" version="2.0">
  <apply> <selector/>
    <ci> x </ci> <cn> 1,2 </cn>
  </apply>
</math>
```

If you want to have a more verbose index, you can use the *csymbol* element, flagged with text encoding.

$$x_{\mathrm{max}}$$

```
<math xmlns="http://www.w3c.org/mathml" version="2.0">
  <apply> <selector/>
    <ci> x </ci>
    <csymbol encoding="text"> max </csymbol>
  </apply>
</math>
```

<> *card* A cardinality is visualized using vertical bars.

$$|A| = 5$$

```
<math xmlns="http://www.w3c.org/mathml" version="2.0">
  <apply> <eq/>
    <apply> <card/>
      <ci> A </ci>
    </apply>
    <ci> 5 </ci>
  </apply>
</math>
```

<> *domain, codomain, image* The next couple of examples are taken from the MathML specification and demonstrate the usage of the not that spectacular domain related elements.

$$\operatorname{domain} f = \mathbb{R}$$

```
<math xmlns="http://www.w3c.org/mathml" version="2.0">
  <apply> <eq/>
    <apply> <domain/>
      <fn> <ci> f </ci> </fn>
    </apply>
    <reals/>
  </apply>
</math>
```

These are typically situations where the *fn* element may show up.

$$\operatorname{codomain} f = \mathbb{Q}$$

```
<math xmlns="http://www.w3c.org/mathml" version="2.0">
  <apply> <eq/>
    <apply> <codomain/>
      <fn> <ci> f </ci> </fn>
    </apply>
    <rationals/>
  </apply>
</math>
```

This example from the MathML specification demonstrates a typical usage of the *image* element. As with the previous two, it is applied to a function, in this case the predefined *sin*.

$$\mathrm{image}\,(\sin) = [-1, 1]$$

```
<math xmlns="http://www.w3c.org/mathml" version="2.0">
  <apply> <eq/>
    <apply> <image/>
      <sin/>
    </apply>
    <interval>
      <cn> -1 </cn>
      <cn>  1 </cn>
    </interval>
  </apply>
</math>
```

<> *domainofapplication* This is another seldom used element. Actually, this element is a further specification of the outer level applied function.

$$\int_C f$$

```
<math xmlns="http://www.w3c.org/mathml" version="2.0">
  <apply> <int/>
    <domainofapplication>
      <ci> C </ci>
    </domainofapplication>
    <ci> f </ci>
  </apply>
</math>
```

<> *semantics, annotation, annotation-xml* We will never know what Albert Einstein would have thought about MathML. But we do know for sure that coding one of his famous findings in xml takes much more tokens that it takes in TeX.

Within a *semantics* element there can be many *annotation* elements. When using ConTeXt, the elements that can be identified as being encoded in TeX will be treated as such. Currently, the related *annotation-xml* element is ignored.

$$e = mc^2$$

```
<math xmlns="http://www.w3c.org/mathml" version="2.0">
  <semantics>
    <apply> <eq/>
      <ci> e </ci>
      <apply> <times/>
        <ci> m </ci>
        <apply> <power/>
          <ci> c </ci>
          <cn> 2 </cn>
        </apply>
      </apply>
    </apply>
    <annotation encoding="tex">
      e = m c^2
    </annotation>
  </semantics>
</math>
```

Another variant that we support is called ‘calcmath’ which is an efficient way to enter school math. The syntax resembles the one used in advanced calculators.

$$x = \sqrt{\sin(x) + \cos(c)}$$

```
<math xmlns="http://www.w3c.org/mathml" version="2.0">
  <semantics>
    <annotation encoding="calcmath">
      x = sqrt(sin(x) + cos(c))
    </annotation>
  </semantics>
</math>
```

<> *integers, reals, ...* Sets are characterized with special (often blackboard) symbols. These symbols are not always available.

integers	$\mathbb{Z}$
reals	$\mathbb{R}$
rationals	$\mathbb{Q}$

naturalnumbers	ℕ
complexes	ℂ
primes	ℙ

<> *pi, imaginaryi, exponentiale* Being a greek character, π is a distinctive character. In most math documents the imaginary i and exponential e are typeset as any math identifier.

pi	π
imaginaryi	i
exponentiale	e

<> *eulergamma, infinity, emptyset* There are a couple of more special tokens. As with the other ones, they can be changed by reassigning the corresponding entities.

eulergamma	γ
infinity	∞
emptyset	

<> *notanumber* Because MathML is used for more purposes than typesetting, there are a couple of elements that do not make much sense in print. One of these is *notanumber*, which is issued by programs as error code or string.

$$\frac{x}{0} = \mathrm{NaN}$$

```
<math xmlns="http://www.w3c.org/mathml" version="2.0">
  <apply> <eq/>
    <apply> <divide/>
      <ci> x </ci>
      <cn> 0 </cn>
    </apply>
    <notanumber/>
  </apply>
</math>
```

<> *true, false* When assigning to a boolean variable, or in boolean expressions one can use 0 or 1 to identify the states, but if you want to be more verbose, you can use these elements.

$$1_2 \equiv \text{true}$$

```
<math xmlns="http://www.w3c.org/mathml" version="2.0">
  <apply> <equivalent/>
    <cn type="integer" base="2"> 1 </cn>
    <true/>
  </apply>
</math>
```

<> *declare* Reusing definitions would be a nice feature, but for the moment the formal specification of this element currently does not give us the freedom to use it the way we want.

$$\text{declare}\, A \;\text{ as }\; (a, b, c)$$

```
<math xmlns="http://www.w3c.org/mathml" version="2.0">
  <declare>
    <ci> A </ci>
    <vector>
      <ci> a </ci>
      <ci> b </ci>
      <ci> c </ci>
    </vector>
  </declare>
</math>
```

<> *csymbol* This element will be implemented as soon as we have an application for it.

is implemented, examples need to be added here

Mixed markup

The advantage of presentational markup is that you can build complicated formulas using super- and subscripts and other elements. The drawback is that the look and feel is rather fixed and cannot easily be adapted to the purpose that the document serves. Take for instance the difference between

$$\log_2 x$$

and

$${}^2\!\log x$$

Both formulas were defined in content MathML, so no explicit super- and subscripts were used. In the next chapter we will see how to achieve such different appearances.

There are situations where content MathML is not rich enough to achieve the desired output. This omission in content MathML forces us to fall back on presentational markup.

$$P_1 = P_2 = 1.01 \approx 1$$

Here we used presentational elements inside a content *ci* element. We could have omitted the outer *ci* element, but since the content MathML parser may base its decisions on the content elements it finds, it is best to keep the outer element there.

```
<math xmlns="http://www.w3c.org/mathml" version="2.0">
  <apply> <eq/>
    <ci> <msub> <mi> P </mi> <mi> 1 </mi> </msub> </ci>
    <ci> <msub> <mi> P </mi> <mi> 2 </mi> </msub> </ci>
    <apply> <approx/>
      <cn> 1.01 </cn>
      <cn> 1 </cn>
    </apply>
  </apply>
</math>
```

The lack of an index element can be quite prominent. For instance, when in an expose about rendering we want to explore the mapping from coordinates in user space to those in device space, we use the following formula.

$$(D_x, D_y, 1) = (U_x, U_y, 1) \begin{pmatrix} s_x & r_x & 0 \\ r_y & s_y & 0 \\ t_x & t_y & 1 \end{pmatrix}$$

```
<math xmlns="http://www.w3c.org/mathml" version="2.0">
  <apply> <eq/>
    <vector>
      <ci> <msub> <mi> D </mi> <mi> x </mi> </msub> </ci>
      <ci> <msub> <mi> D </mi> <mi> y </mi> </msub> </ci>
      <cn> 1 </cn>
    </vector>
    <apply> <times/>
      <vector>
        <ci> <msub> <mi> U </mi> <mi> x </mi> </msub> </ci>
        <ci> <msub> <mi> U </mi> <mi> y </mi> </msub> </ci>
        <cn> 1 </cn>
      </vector>
      <matrix>
        <matrixrow>
          <ci> <msub> <mi> s </mi> <mi> x </mi> </msub> </ci>
          <ci> <msub> <mi> r </mi> <mi> x </mi> </msub> </ci>
          <cn> 0 </cn>
        </matrixrow>
        <matrixrow>
          <ci> <msub> <mi> r </mi> <mi> y </mi> </msub> </ci>
          <ci> <msub> <mi> s </mi> <mi> y </mi> </msub> </ci>
          <cn> 0 </cn>
        </matrixrow>
        <matrixrow>
          <ci> <msub> <mi> t </mi> <mi> x </mi> </msub> </ci>
          <ci> <msub> <mi> t </mi> <mi> y </mi> </msub> </ci>
          <cn> 1 </cn>
        </matrixrow>
      </matrix>
    </apply>
  </apply>
</math>
```

Again, the *msub* element provides a way out, as in the next examples, which are adapted versions of formulas we used when demonstrating the statistics related elements.

$$\overline{x} = \frac{1}{n} \sum_{i} x$$

```
<math xmlns="http://www.w3c.org/mathml" version="2.0">
  <apply> <eq/>
    <apply> <mean/>
      <ci> x </ci>
    </apply>
    <apply> <times/>
      <apply> <divide/>
        <cn> 1 </cn>
        <ci> n </ci>
      </apply>
      <apply> <sum/>
        <bvar> <ci> i </ci> </bvar>
        <ci> x </ci>
      </apply>
    </apply>
  </apply>
</math>
```

$$\overline{x} = \frac{1}{n}\sum_{i=1}^{n} x$$

```
<math xmlns="http://www.w3c.org/mathml" version="2.0">
  <apply> <eq/>
    <apply> <mean/>
      <ci> x </ci>
    </apply>
    <apply> <times/>
      <apply> <divide/>
        <cn> 1 </cn>
        <ci> n </ci>
      </apply>
      <apply> <sum/>
        <bvar> <ci> i </ci> </bvar>
        <lowlimit> <cn> 1 </cn> </lowlimit>
        <uplimit> <cn> n </cn> </uplimit>
        <ci> x </ci>
      </apply>
    </apply>
  </apply>
```

```
</math>
```

$$\overline{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$$

```
<math xmlns="http://www.w3c.org/mathml" version="2.0">
  <apply> <eq/>
    <apply> <mean/>
      <ci> x </ci>
    </apply>
    <apply> <times/>
      <apply> <divide/>
        <cn> 1 </cn>
        <ci> n </ci>
      </apply>
      <apply> <sum/>
        <bvar> <ci> i </ci> </bvar>
        <lowlimit> <cn> 1 </cn> </lowlimit>
        <uplimit> <cn> n </cn> </uplimit>
        <ci> <msub> <mi> x </mi> <mi> i </mi> </msub> </ci>
      </apply>
    </apply>
  </apply>
</math>
```

You can also use a selector for indexing, so in practice we can avoid the mixed mode:

$$(D_x, D_y, 1) = (U_x, U_y, 1)\begin{pmatrix} s_x & r_x & 0 \\ s_y & r_y & 0 \\ t_x & t_y & 1 \end{pmatrix}$$

```
<math xmlns="http://www.w3c.org/mathml" version="2.0">
  <apply> <eq/>
    <vector>
      <apply> <selector/> <ci> D </ci> <ci> x </ci> </apply>
      <apply> <selector/> <ci> D </ci> <ci> y </ci> </apply>
      <cn> 1 </cn>
    </vector>
    <apply> <times/>
      <vector>
```

```
        <apply> <selector/> <ci> U </ci> <ci> x </ci> </apply>
        <apply> <selector/> <ci> U </ci> <ci> y </ci> </apply>
        <cn> 1 </cn>
      </vector>
      <matrix>
        <matrixrow>
          <apply> <selector/> <ci> s </ci> <ci> x </ci> </apply>
          <apply> <selector/> <ci> r </ci> <ci> x </ci> </apply>
          <cn> 0 </cn>
        </matrixrow>
        <matrixrow>
          <apply> <selector/> <ci> s </ci> <ci> y </ci> </apply>
          <apply> <selector/> <ci> r </ci> <ci> y </ci> </apply>
          <cn> 0 </cn>
        </matrixrow>
        <matrixrow>
          <apply> <selector/> <ci> t </ci> <ci> x </ci> </apply>
          <apply> <selector/> <ci> t </ci> <ci> y </ci> </apply>
          <cn> 1 </cn>
        </matrixrow>
      </matrix>
    </apply>
  </apply>
</math>
```

Mixed markup

Directives

Some elements can be tuned by changing their attributes. Especially when formulas are defined by a team of people or when they are taken from a repository, there is a good chance that inconsistencies will show up.

In ConTeXt, you can influence the appearance by setting the typesetting parameters of (classes of) elements. You can do this either by adding processing instructions, or by using the ConTeXt command `\setupMMLappearance`. Although the first method is more in the spirit of xml, the second method is more efficient and consistent. As a processing instruction, a directive looks like:

```
<?context-mathml-directive element key value ?>
```

This is equivalent to the ConTeXt command:

```
\setupMMLappearance [element] [key=value]
```

Some settings concern a group of elements, in which case a group classification (like `sign`) is used.

<> *scripts* By default, nested super- and subscripts are kind of isolated from each other. If you want a combined script, there is the *msubsup*. You can however force combinations with a directive.

$${x_1}^2$$

```
<math xmlns="http://www.w3c.org/mathml" version="2.0">
  <msup>
    <msub> <mi> x </mi> <mn> 1 </mn> </msub>
    <mn> 2 </mn>
  </msup>
</math>
```

$$x_1^2$$

```
<?context-mathml-directive scripts alternative b ?>

<math xmlns="http://www.w3c.org/mathml" version="2.0">
  <msup>
    <msub> <mi> x </mi> <mn> 1 </mn> </msub>
    <mn> 2 </mn>
  </msup>
</math>
```

<> *sign* The core element of MathML is *apply*. Even simple formulas will often have more than one (nested) *apply*. The most robust way to handle nested formulas is to use braces around each sub formula. No matter how robust this is, when presented in print we want to use as less braces as possible. The next example shows addition as well as subtraction.

$$7 + 5 - 3$$

```
<math xmlns="http://www.w3c.org/mathml" version="2.0">
  <apply> <plus/>
    <cn> 7 </cn>
    <cn> 5 </cn>
    <apply> <minus/>
      <cn> 3 </cn>
    </apply>
  </apply>
</math>
```

In principle subtraction is adding negated numbers, so it would have been natural to have just an addition (*plus*) and negation operator. However, MathML provides both a *plus* and *minus* operator, where the latter can be used as a negation. So in fact we have:

$$7 + 5 + (-3)$$

Now imagine that a teacher wants to stress this negation in the way presented here, using parentheses. Since all the examples shown here are typeset directly from the MathML source, you may expect a solution, so here it is:

```
<math xmlns="http://www.w3c.org/mathml" version="2.0">
  <?context-mathml-directive sign reduction no ?>
  <apply> <plus/>
    <cn> 7 </cn>
    <cn> 5 </cn>
    <apply> <minus/>
      <cn> 3 </cn>
    </apply>
  </apply>
</math>
```

By default signs are reduced, but one can disable that at the document and/or formula level using a processing instruction at the top of the formula. There are of course circumstances where the parentheses cannot be left out.

$$a + (b + c) + d$$

```
<math xmlns="http://www.w3c.org/mathml" version="2.0">
  <apply> <plus/>
    <ci> a </ci>
    <apply> <plus/> <ci> b </ci> <ci> c </ci> </apply>
    <ci> d </ci>
  </apply>
</math>
```

$$a - (b - c) - d$$

```
<math xmlns="http://www.w3c.org/mathml" version="2.0">
  <apply> <minus/>
    <ci> a </ci>
    <apply> <minus/> <ci> b </ci> <ci> c </ci> </apply>
    <ci> d </ci>
  </apply>
</math>
```

$$a + (b - c) + d$$

```
<math xmlns="http://www.w3c.org/mathml" version="2.0">
  <apply> <plus/>
    <ci> a </ci>
    <apply> <minus/> <ci> b </ci> <ci> c </ci> </apply>
    <ci> d </ci>
  </apply>
</math>
```

$$a - (b + c) - d$$

```
<math xmlns="http://www.w3c.org/mathml" version="2.0">
  <apply> <minus/>
    <ci> a </ci>
    <apply> <plus/> <ci> b </ci> <ci> c </ci> </apply>
    <ci> d </ci>
  </apply>
</math>
```

Another place where parentheses are not needed is the following:

```
<math xmlns="http://www.w3c.org/mathml" version="2.0">
  <apply> <minus/>
    <apply> <exp/>
      <cn> 3 </cn>
    </apply>
  </apply>
</math>
```

This means that the interpreter of this kind of MathML has to analyze child elements in order to choose the right way to typeset the formula. The output will look like:

$$-\mathrm{e}^{3}$$

By default, as less braces as possible are used. As demonstrated, a special case is when *plus* and *minus* have one sub element to deal with. If you really want many braces there, you can turn off sign reduction.

`sign reduction`	`yes`	use as less braces as possible
	`no`	always use braces

We will demonstrate these alternatives with an example.

$$a + \sin b + c^{5} + \sin^{2} d + e$$

We need quite some code to encode this formula.

```
<math xmlns="http://www.w3c.org/mathml" version="2.0">
  <apply> <plus/>
    <ci> a </ci>
    <apply> <sin/>
      <ci> b </ci>
    </apply>
    <apply> <power/>
      <ci> c </ci>
      <cn> 5 </cn>
    </apply>
    <apply> <power/>
      <apply> <sin/>
        <ci> d </ci>
      </apply>
      <cn> 2 </cn>
    </apply>
    <ci> e </ci>
  </apply>
</math>
```

With power reduction turned off, we get:

$$a + \sin b + c^5 + (\sin d)^2 + e$$

As directive we used:

```
<?context-mathml-directive power reduction no ?>
```

The following example illustrates that we should be careful in coding such formulas; here the *power* is applied to the argument of *sin*.

$$a + \sin b + c^5 + \sin \left(d^2\right) + e$$

```
<math xmlns="http://www.w3c.org/mathml" version="2.0">
  <apply> <plus/>
    <ci> a </ci>
    <apply> <sin/>
      <ci> b </ci>
    </apply>
    <apply> <power/>
      <ci> c </ci>
      <cn> 5 </cn>
    </apply>
    <apply> <sin/>
      <apply> <power/>
        <ci> d </ci>
        <cn> 2 </cn>
      </apply>
    </apply>
    <ci> e </ci>
  </apply>
</math>
```

<> *divide* Divisions can be very space consuming but there is a way out: using a forward slash symbol. You can set the level at which this will take place. By default, fractions are typeset in the traditional way.

$$\frac{1}{1+\frac{1}{x}}$$

```
<math xmlns="http://www.w3c.org/mathml" version="2.0">
  <apply> <divide/>
    <cn> 1 </cn>
    <apply> <plus/>
      <cn> 1 </cn>
      <apply> <divide/>
        <cn> 1 </cn>
        <ci> x </ci>
      </apply>
    </apply>
  </apply>
</math>
```

$$\frac{1}{1+\frac{1}{1+\frac{1}{x}}}$$

```
<math xmlns="http://www.w3c.org/mathml" version="2.0">
  <apply> <divide/>
    <cn> 1 </cn>
    <apply> <plus/>
      <cn> 1 </cn>
      <apply> <divide/>
        <cn> 1 </cn>
        <apply> <plus/>
          <cn> 1 </cn>
          <apply> <divide/>
            <cn> 1 </cn>
            <ci> x </ci>
          </apply>
        </apply>
      </apply>
    </apply>
  </apply>
</math>
```

$$\frac{1}{1+1\,/\,x}$$

$$\frac{1}{1+1\,/\,(1+1\,/\,x)}$$

```
<?context-mathml-directive divide level 1 ?>
```

$$\frac{1}{1+\frac{1}{x}}$$

$$\frac{1}{1+\frac{1}{1+1/x}}$$

```
<?context-mathml-directive divide level 2 ?>
```

<> *relation* You should keep in mind that (at least level 2) content MathML is not that rich in terms of presenting your ideas in a visually attractive way. On the other hand, because the content is highly structured, some intelligence can be applied when typesetting them. By default, a relation is not vertically aligned but typeset horizontally.

If an application just needs raw formulas, definitions like the following are all right.

```
<math xmlns="http://www.w3c.org/mathml" version="2.0">
  <apply> <eq/>
    <apply> <plus/>
      <ci> a </ci>
      <ci> b </ci>
      <ci> c </ci>
    </apply>
    <apply> <plus/>
      <ci> d </ci>
      <ci> e </ci>
    </apply>
    <apply> <plus/>
```

```
      <ci> f </ci>
      <ci> g </ci>
      <ci> h </ci>
      <ci> i </ci>
    </apply>
    <cn> 123 </cn>
  </apply>
</math>
```

The typeset result will bring no surprises:

$$a+b+c=d+e=f+g+h+i=123$$

But, do we want to show a formula that way? And what happens with much longer formulas? You can influence the appearance with processing instructions.

`relation`	`align`	`no`	don't align relations
		`left`	align all relations left
		`right`	align all relations right
		`first`	place the leftmost relation left
		`last`	place the rightmost relation right

The next couple of formulas demonstrate in what way the previously defined formula is influenced by the processing instructions.

$$\begin{aligned} a+b+c &= \\ d+e &= \\ f+g+h+i &= \\ 123 & \end{aligned}$$

```
<?context-mathml-directive relation align left ?>
```

$$\begin{aligned} & a+b+c \\ &= d+e \\ &= f+g+h+i \\ &= 123 \end{aligned}$$

```
<?context-mathml-directive relation align right ?>
```

$$\begin{aligned} a+b+c &= d+e \\ &= f+g+h+i \\ &= 123 \end{aligned}$$

```
<?context-mathml-directive relation align first ?>
```

$$\begin{aligned} a+b+c &= \\ d+e &= \\ f+g+h+i &= 123 \end{aligned}$$

```
<?context-mathml-directive relation align last ?>
```

<> *base* When in a document several number systems are used, it can make sense to mention the base of the number. There are several ways to identify the base.

`base`	`symbol`	`numbers`	a (decimal) number
		`characters`	one character
		`text`	a mnemonic
		`no`	no symbol

By default, when specified, a base is identified as number.

```
<math xmlns="http://www.w3c.org/mathml" version="2.0">
  <cn type="integer" base="8"> 1427 </cn>
</math>
```

$$1427_{8}$$

```
<?context-mathml-directive base symbol numbers ?>
```

$$1427_{\mathrm{O}}$$

```
<?context-mathml-directive base symbol characters ?>
```

$$1427_{\mathrm{OCT}}$$

```
<?context-mathml-directive base symbol text ?>
```

<> *function* There is a whole bunch of functions available as empty element, like *sin* and *log*. When a function is applied to a function, braces make not much sense and placement is therefore disabled.

function	reduction	yes	chain functions without braces
		no	put braces around nested functions

```
<math xmlns="http://www.w3c.org/mathml" version="2.0">
  <apply> <sin/> <ci> x </ci> </apply>
</math>
```

$$\sin x$$

```
<?context-mathml-directive function reduction yes ?>
```

$$\sin\,(x)$$

```
<?context-mathml-directive function reduction no ?>
```

<> *limits* When limits are placed on top of the limitation symbol, this generally looks better than when they are placed alongside. You can also influence limit placement per element. This feature is available for *int*, *sum*, *product* and *limit*.

limit	location	top	place limits on top of the symbols
		right	attached limits as super/subscripts

```
<math xmlns="http://www.w3c.org/mathml" version="2.0">
  <apply> <int/>
    <bvar> <ci> x </ci> </bvar>
    <lowlimit> <cn> 0 </cn> </lowlimit>
    <uplimit> <cn> 1 </cn> </uplimit>
  </apply>
</math>
```

$$\int\limits_{0}^{1} \mathrm{d}x$$

```
<?context-mathml-directive int location top ?>
```

$$\int_{0}^{1} \mathrm{d}x$$

```
<?context-mathml-directive int location right ?>
```

<> *declare* Currently declarations are not supposed to end up in print. By default we typeset a message, but you can as well completely hide declarations.

declare	state	start	show declarations
		stop	ignore (hide) declarations

<> *lambda* There is more than one way to visualize a lambda function. As with some other settings, changing the appearance can best take place at the document level.

lambda	alternative	b	show lambda as arrow
		a	show lambda as set

```
<math xmlns="http://www.w3c.org/mathml" version="2.0">
  <lambda>
    <bvar> <ci> x </ci> </bvar>
    <apply> <log/>
      <ci> x </ci>
    </apply>
  </lambda>
</math>
```

$$\lambda\,(x, \log x)$$

```
<?context-mathml-directive lambda alternative a ?>
```

$$x \mapsto \log x$$

```
<?context-mathml-directive lambda alternative b ?>
```

<> *power* Taking the power of a function looks clumsy when braces are put around the function. Therefore, by default, the power is applied to the function symbol instead of the whole function.

power	reduction	yes	attach symbol to function symbol
		no	attach symbol to function argument

```
<math xmlns="http://www.w3c.org/mathml" version="2.0">
  <apply> <power/>
```

```
    <apply> <ln/>
      <ci> x </ci>
    </apply>
    <cn> 3 </cn>
  </apply>
</math>
```

$$\ln^3 x$$

```
<?context-mathml-directive power reduction yes ?>
```

$$(\ln x)^3$$

```
<?context-mathml-directive power reduction no ?>
```

<> *diff* Covering all kind of differential formulas is not trivial. Currently we support two locations for the operand (function). By default the operand is placed above the division line.

`diff`	`location`	`top`	put the operand in the fraction
		`right`	put the operand after the fraction

```
<math xmlns="http://www.w3c.org/mathml" version="2.0">
  <apply> <diff/>
    <bvar>
      <ci> x </ci>
      <degree> <cn> 2 </cn> </degree>
    </bvar>
    <apply> <fn> <ci> f </ci> </fn>
      <apply> <plus/>
        <apply> <times/>
          <cn> 2 </cn>
          <ci> x </ci>
        </apply>
        <cn> 1 </cn>
      </apply>
    </apply>
  </apply>
</math>
```

$$\frac{\mathrm{d}^2 f(2x+1)}{\mathrm{d}x^2}$$

```
<?context-mathml-directive diff location top ?>
```

$$\frac{\mathrm{d}^2}{\mathrm{d}x^2}(f(2x+1))$$

```
<?context-mathml-directive diff location right ?>
```

<> *vector* Depending on the complication of a vector or on the available space, you may wish to typeset a vector horizontally or vertically. By default a vector is typeset horizontally.

vector	direction	horizontal	put vector elements alongside
		vertical	stack vector elements

```
<math xmlns="http://www.w3c.org/mathml" version="2.0">
  <apply> <eq/>
    <vector>
      <ci> x </ci>
      <ci> y </ci>
      <ci> z </ci>
    </vector>
    <vector>
      <cn> 1 </cn>
      <cn> 0 </cn>
      <cn> 1 </cn>
    </vector>
  </apply>
</math>
```

$$(x, y, z) = (1, 0, 1)$$

```
<?context-mathml-directive vector direction horizontal ?>
```

$$(\,x,y,z\,) = (\,1,0,1\,)$$

```
<?context-mathml-directive vector direction vertical ?>
```

<> *times* Depending on the audience, a multiplication sign is implicit (absent) or represented by a regular times symbol or a dot.

`times`	`symbol`	`no`	don’t add a symbol
		`yes`	separate operands by a times (×)
		`dot`	separate operands by a dot (·)
`auto`	`symbol`	`no`	don’t check for succesive numbers
		`yes`	separate succesive numbers by a times (×)
		`dot`	separate succesive numbers by a dot (·)

```
<math xmlns="http://www.w3c.org/mathml" version="2.0">
  <apply> <plus/>
    <ci> x </ci>
    <apply> <times/>
      <cn> 2 </cn>
      <ci> x </ci>
    </apply>
  </apply>
</math>
```

$$x + 2x$$

```
<?context-mathml-directive times symbol no ?>
```

$$x + 2 \times x$$

```
<?context-mathml-directive times symbol yes ?>
```

$$x + 2 \cdot x$$

```
<?context-mathml-directive times symbol dot ?>
```

<> *log* The location of a logbase depends on tradition and/or preference, which is why we offer a few alternatives: as pre superscript (in the right top corner before the symbol) or as post subscript (in the lower left corner after the symbol).

`log`	`location`	`right`	place logbase at the right top
		`left`	place logbase at the lower left

```
<math xmlns="http://www.w3c.org/mathml" version="2.0">
  <apply> <log/>
    <logbase>
```

```
      <ci> 3 </ci>
    </logbase>
    <apply> <plus/>
      <ci> x </ci>
      <cn> 1 </cn>
    </apply>
  </apply>
</math>
```

$$\log_3 (x + 1)$$

```
<?context-mathml-directive log location right ?>
```

$$^3\log (x + 1)$$

```
<?context-mathml-directive log location left ?>
```

<> *polar* For polar notation we provide several renderings:

polar	alternative	a	explicit polar notation
		b	exponential power notation
		c	exponential function notation

```
<math xmlns="http://www.w3c.org/mathml" version="2.0">
  <cn type="polar"> 2 <sep/> <pi/> </cn>
</math>
```

$$\mathrm{Polar}\,(2, \pi)$$

```
<?context-mathml-directive polar alternative a ?>
```

$$\mathrm{e}^{2+\mathrm{i}}$$

```
<?context-mathml-directive polar alternative b ?>
```

$$\exp\,(2 + \mathrm{i})$$

```
<?context-mathml-directive polar alternative c ?>
```

<> *e-notation* Depending on the context, you may want to typeset the number `1.23e4` not as this sequence, but using a multiplier construct. As with the *times*, we support both multiplication symbols.

enotation	symbol	no	no interpretation
		yes	split exponent, using ×
		dot	split exponent, using ·

```
<math xmlns="http://www.w3c.org/mathml" version="2.0">
  <cn type="e-notation">10<sep/>23</cn>
</math>
```

$$10\,\mathrm{e}\,23$$

```
<?context-mathml-directive enotation symbol no ?>
```

$$10 \times 10^{23}$$

```
<?context-mathml-directive enotation symbol yes ?>
```

$$10 \cdot 10^{23}$$

```
<?context-mathml-directive enotation symbol dot ?>
```

Typesetting modes

Math can be typeset inline or display. In order not to widen up the text of a paragraph too much, inline math is typeset more cramped. Since MathML does provide just a general purpose *math* element we have to provide the information needed using other elements. Consider the following text.

> To what extent is math supposed to reflect the truth and nothing but the truth? Consider the simple expression $10 = 3 + 7$. Many readers will consider this the truth, but then, can we assume that the decimal notation is used?
>
> $$10 = 3 + x$$
>
> In many elementary math books, you can find expressions like the previous. Because in our daily life we use the decimal numbering system, we can safely assume that $x = 7$. But, without explicitly mentioning this boundary condition, more solutions are correct.
>
> $$10 = 3 + 5 \tag{1.a}$$
>
> In formula 1.a we see an at first sight wrong formula. But, if we tell you that octal numbers are used, your opinion may change instantly. A rather clean way out of this confusion is to extend the notation of numbers by explicitly mentioning the base.
>
> $$10_8 = 3_8 + 5_8 \tag{2.b}$$
>
> Of course, when a whole document is in octal notation, a proper introduction is better than annotated numbers as used in formula 2.a.

In terms of xml this can look like:

```
<document>
To what extent is math supposed to reflect the truth and nothing but
the truth? Consider the simple expression
    <math xmlns="http://www.w3c.org/mathml" version="2.0">
        <apply> <eq/>
            <cn> 10 </cn>
            <apply> <plus/>
                <cn> 3 </cn>
                <cn> 7 </cn>
            </apply>
        </apply>
    </math>. Many readers will consider this the truth, but then,
```

can we assume that the decimal notation is used?

```
<formula>
  <math xmlns="http://www.w3c.org/mathml" version="2.0">
    <apply> <eq/>
      <cn> 10 </cn>
      <apply> <plus/>
        <cn> 3 </cn>
        <ci> x </ci>
      </apply>
    </apply>
  </math>
</formula>
```

In many elementary math books, you can find expressions like the previous. Because in our daily life we use the decimal numbering system, we can safely assume that

```
    <math xmlns="http://www.w3c.org/mathml" version="2.0">
        <apply> <eq/>
            <ci> x </ci>
            <cn> 7 </cn>
        </apply>
    </math>. But, without explicitly mentioning this boundary condition,
```

more solutions are correct.

```
<formula label="octal" sublabel="a">
  <math xmlns="http://www.w3c.org/mathml" version="2.0">
    <apply> <eq/>
      <cn> 10 </cn>
      <apply> <plus/>
        <cn> 3 </cn>
        <cn> 5 </cn>
      </apply>
    </apply>
  </math>
</formula>
```

In <textref label="octal">formula</textref> we see an at first sight

```
wrong formula. But, if we tell you that octal numbers are used, your
opinion may change instantly. A rather clean way out of this confusion
is to extend the notation of numbers by explicitly mentioning the base.

<subformula label="octal base" sublabel="b">
  <math xmlns="http://www.w3c.org/mathml" version="2.0">
    <apply> <eq/>
      <cn type="integer" base="8"> 10 </cn>
      <apply> <plus/>
        <cn type="integer" base="8"> 3 </cn>
        <cn type="integer" base="8"> 5 </cn>
      </apply>
    </apply>
  </math>
</subformula>

Of course, when a whole document is in octal notation, a proper
introduction is better than annotated numbers as used in <textref
label="octal base">formula</textref>.
</document>
```

Math that is part of the text flow is automatically handled as in line math. If needed you can encapsulate the code in an *imath* environment. Display math is recognized as such when it is a separate paragraph, but since this is more a TeX feature than an xml one, you should encapsulate display math either in a *dmath* element or in a *formula* or *subformula* element.

For a while you can use attribute `mode` with values `display` or `inline`. Recent MathML specifications provide the `display` attribute with values `block` or `inline`. We support both.

Getting started

A comfortable way to get accustomed to MathML is to make small documents of the following form:

```
\usemodule[mathml]

\starttext

\startbuffer
<math xmlns="http://www.w3c.org/mathml" version="2.0">
  <apply> <cos/>
    <ci> x </ci>
  </apply>
</math>
\stopbuffer

\processxmlbuffer

\stoptext
```

As you see, we can mix MathML with normal TEX code. A document like this is processed in the normal way using the `context` command. If you also want to see the original code, you can say:

```
\usemodule[mathml]

\starttext

\startbuffer
<math xmlns="http://www.w3c.org/mathml" version="2.0">
  <apply> <cos/>
    <ci> x </ci>
  </apply>
</math>
\stopbuffer

\processxmlbuffer

\typebuffer
```

```
\stoptext
```

Like TEX and METAPOST code, buffers can contain MathML code. The advantage of this method is that we only have to key in the data once. It also permits you to experiment with processing instructions.

```
\startbuffer[mml]
<math xmlns="http://www.w3c.org/mathml" version="2.0">
  <apply> <log/>
    <logbase> <cn> 3.5 </cn> </logbase>
    <ci> x </ci>
  </apply>
</math>
\stopbuffer

\startbuffer[pi]
 <?context-mathml-directive log location right ?>
\stopbuffer

\processxmlbuffer[pi,mml]

\startbuffer[pi]
 <?context-mathml-directive log location left ?>
\stopbuffer

\processxmlbuffer[pi,mml]
```

If you like coding your documents in TEX but want to experiment with MathML, combining both languages in the way demonstrated here may be an option. When you provide enough structure in your TEX code, converting a document to xml is then not that hard to do. Where coding directly in xml is kind of annoying, coding MathML is less cumbersome, because you can structure your formulas pretty well, especially since the fragments are small so that proper indentation is possible.

OpenMath

Because OpenMath is now a subset of MathML we can to some extend also support this coding. We do a straightforward remapping to content MathML so any rendering that is supported there is also supported in the equivalent OpenMath code.

$$y = f(x) - f(x - 1)$$

```
<OMOBJ xmlns="http://www.openmath.org/OpenMath" version="2.0">
  <OMA> <OMS cd="relation1" name="eq"/>
    <OMV name="y"/>
    <OMA> <OMS cd="arith1" name="minus"/>
      <OMA> <OMV name="f"/>
        <OMV name="x"/>
      </OMA>
      <OMA> <OMV name="f"/>
        <OMA> <OMS cd="arith1" name="minus"/>
          <OMV name="x"/>
          <OMI>1</OMI>
        </OMA>
      </OMA>
    </OMA>
  </OMA>
</OMOBJ>
```

Because in practice we may use a mixture of math encodings this can come in handy because it saves conversion of the xml source.

CalcMath

We support two types of annotation markup: TEX (tex) and what we call ‘calculator math’ (calcmath). The second type is also available directly. Inline calcmath is coded using the *icm* element.

This is an inline formula $\sin(x^2 + \frac{1}{x})$ just to demonstrate the idea of calculator math.

```
<document>
  This is an inline formula <icm>sin(x^2+1/x)</icm> just to demonstrate
  the idea of calculator math.
</document>
```

If one edits the xml file directly this can type quite some coding. For more complex formulas one can revert to content MathML, or when interactivity is needed to OpenMath.

The argument that one should use a dedicated editor for math instead is not that convincing for authors who have to key on lots of small snippets of math. And one can always transform this code in its more bloated variant. The calcmath converter is dedicated to Frits Spijkers, author of Dutch math schoolbooks and fluent in all those math encodings methods we force upon him. The code resembles that used in calculators at schools and we used it in projects with computer aided feedback where students had to key in math.

A few examples

<> Derivatives

derivate 1

$$\frac{\mathrm{d}a}{\mathrm{d}x} = 0$$

```
<math xmlns='http://www.w3c.org/mathml' version='2.0'>
  <apply> <eq/>
    <apply> <diff/>
      <bvar> <ci> x </ci> </bvar>
      <ci> a </ci>
    </apply>
    <ci> 0 </ci>
  </apply>
</math>
```

derivate 2

$$\frac{\mathrm{d}x}{\mathrm{d}x} = 1$$

```
<math xmlns='http://www.w3c.org/mathml' version='2.0'>
  <apply> <eq/>
    <apply> <diff/>
      <bvar> <ci> x </ci> </bvar>
      <ci> x </ci>
    </apply>
    <cn> 1 </cn>
  </apply>
</math>
```

derivate 3

$$\frac{\mathrm{d}\,(au)}{\mathrm{d}x} = a\frac{\mathrm{d}u}{\mathrm{d}x}$$

```
<math xmlns='http://www.w3c.org/mathml' version='2.0'>
  <apply> <eq/>
    <apply> <diff/>
      <bvar> <ci> x </ci> </bvar>
      <apply> <times/>
        <ci> a </ci>
        <ci> u </ci>
      </apply>
    </apply>
    <apply> <times/>
      <ci> a </ci>
      <apply> <diff/>
        <bvar> <ci> x </ci> </bvar>
        <ci> u </ci>
      </apply>
    </apply>
  </apply>
</math>
```

derivate 4

$$\frac{\mathrm{d}\,(u+v+w)}{\mathrm{d}x} = \frac{\mathrm{d}u}{\mathrm{d}x} + \frac{\mathrm{d}v}{\mathrm{d}x} + \frac{\mathrm{d}w}{\mathrm{d}x}$$

```
<math xmlns='http://www.w3c.org/mathml' version='2.0'>
  <apply> <eq/>
    <apply> <diff/>
      <bvar> <ci> x </ci> </bvar>
      <apply> <plus/>
        <ci> u </ci>
        <ci> v </ci>
        <ci> w </ci>
      </apply>
    </apply>
    <apply> <plus/>
      <apply> <diff/>
        <bvar> <ci> x </ci> </bvar>
        <ci> u </ci>
      </apply>
      <apply> <diff/>
        <bvar> <ci> x </ci> </bvar>
        <ci> v </ci>
      </apply>
      <apply> <diff/>
        <bvar> <ci> x </ci> </bvar>
        <ci> w </ci>
      </apply>
    </apply>
  </apply>
```

```
</math>
```

derivate 5

$$\frac{\mathrm{d}\,(uv)}{\mathrm{d}x} = u\frac{\mathrm{d}u}{\mathrm{d}x} + v\frac{\mathrm{d}v}{\mathrm{d}x}$$

```
<math xmlns='http://www.w3c.org/mathml' version='2.0'>
  <apply> <eq/>
    <apply> <diff/>
      <bvar> <ci> x </ci> </bvar>
      <apply> <times/>
        <ci> u </ci>
        <ci> v </ci>
      </apply>
    </apply>
    <apply> <plus/>
      <apply> <times/>
        <ci> u </ci>
        <apply> <diff/>
          <bvar> <ci> x </ci> </bvar>
          <ci> u </ci>
        </apply>
      </apply>
      <apply> <times/>
        <ci> v </ci>
        <apply> <diff/>
          <bvar> <ci> x </ci> </bvar>
          <ci> v </ci>
        </apply>
      </apply>
    </apply>
  </apply>
</math>
```

derivate 6

$$\frac{\mathrm{d}\,(uvw)}{\mathrm{d}x} = vw\frac{\mathrm{d}u}{\mathrm{d}x} + uw\frac{\mathrm{d}v}{\mathrm{d}x} + uv\frac{\mathrm{d}w}{\mathrm{d}x}$$

```
<math xmlns='http://www.w3c.org/mathml' version='2.0'>
  <apply> <eq/>
    <apply> <diff/>
      <bvar> <ci> x </ci> </bvar>
      <apply> <times/>
        <ci> u </ci>
        <ci> v </ci>
        <ci> w </ci>
      </apply>
    </apply>
    <apply> <plus/>
      <apply> <times/>
        <ci> v </ci>
        <ci> w </ci>
        <apply> <diff/>
          <bvar> <ci> x </ci> </bvar>
          <ci> u </ci>
        </apply>
      </apply>
      <apply> <times/>
        <ci> u </ci>
        <ci> w </ci>
        <apply> <diff/>
          <bvar> <ci> x </ci> </bvar>
          <ci> v </ci>
        </apply>
      </apply>
      <apply> <times/>
        <ci> u </ci>
         <ci> v </ci>
        <apply> <diff/>
          <bvar> <ci> x </ci> </bvar>
          <ci> w </ci>
        </apply>
      </apply>
    </apply>
  </apply>
</math>
```

derivate 7

$$\frac{\mathrm{d}\left(\frac{u}{v}\right)}{\mathrm{d}x} = \frac{v\frac{\mathrm{d}u}{\mathrm{d}x} - u\frac{\mathrm{d}v}{\mathrm{d}x}}{v^2} = \frac{1}{v}\frac{\mathrm{d}u}{\mathrm{d}x} - \frac{u}{v^2}\frac{\mathrm{d}v}{\mathrm{d}x}$$

```
<math xmlns='http://www.w3c.org/mathml' version='2.0'>
  <apply> <eq/>
    <apply> <diff/>
      <bvar> <ci> x </ci> </bvar>
      <apply> <divide/>
        <ci> u </ci>
        <ci> v </ci>
      </apply>
    </apply>
    <apply> <divide/>
      <apply> <minus/>
        <apply> <times/>
          <ci> v </ci>
          <apply> <diff/>
            <bvar> <ci> x </ci> </bvar>
            <ci> u </ci>
          </apply>
        </apply>
        <apply> <times/>
          <ci> u </ci>
          <apply> <diff/>
            <bvar> <ci> x </ci> </bvar>
            <ci> v </ci>
          </apply>
        </apply>
      </apply>
      <apply> <power/>
        <ci> v </ci>
        <cn> 2 </cn>
      </apply>
    </apply>
    <apply> <minus/>
      <apply> <times/>
        <apply> <divide/>
          <cn> 1 </cn>
          <ci> v </ci>
        </apply>
        <apply> <diff/>
          <bvar> <ci> x </ci> </bvar>
          <ci> u </ci>
        </apply>
      </apply>
      <apply> <times/>
        <apply> <divide/>
          <cn> u </cn>
          <apply> <power/>
            <ci> v </ci>
            <cn> 2 </cn>
          </apply>
        </apply>
        <apply> <diff/>
          <bvar> <ci> x </ci> </bvar>
          <ci> v </ci>
        </apply>
      </apply>
    </apply>
  </apply>
</math>
```

derivate 8

$$\frac{\mathrm{d}\left(u^{n}\right)}{\mathrm{d}x} = n\left(u\right)^{n-1}\frac{\mathrm{d}u}{\mathrm{d}x}$$

```
<math xmlns='http://www.w3c.org/mathml' version='2.0'>
  <apply> <eq/>
    <apply> <diff/>
      <bvar> <ci> x </ci> </bvar>
      <apply> <power/>
        <ci> u </ci>
        <ci> n </ci>
      </apply>
    </apply>
    <apply> <times/>
      <ci> n </ci>
      <apply> <power/>
```

```
        <ci> u </ci>
        <apply> <minus/>
          <ci> n </ci>
          <cn> 1 </cn>
        </apply>
      </apply>
      <apply> <diff/>
        <bvar> <ci> x </ci> </bvar>
        <ci> u </ci>
      </apply>
    </apply>
  </apply>
</math>
```

derivate 9

$$\frac{\mathrm{d}\sqrt{u}}{\mathrm{d}x} = \frac{1}{2\sqrt{u}}\frac{\mathrm{d}u}{\mathrm{d}x}$$

```
<math xmlns='http://www.w3c.org/mathml' version='2.0'>
  <apply> <eq/>
    <apply> <diff/>
      <bvar> <ci> x </ci> </bvar>
      <apply> <root/>
        <ci> u </ci>
      </apply>
    </apply>
    <apply> <times/>
      <apply> <divide/>
        <cn> 1 </cn>
        <apply> <times/>
          <cn> 2 </cn>
          <apply> <root/>
            <ci> u </ci>
          </apply>
        </apply>
      </apply>
      <apply> <diff/>
        <bvar> <ci> x </ci> </bvar>
        <ci> u </ci>
      </apply>
    </apply>
  </apply>
</math>
```

derivate 10

$$\frac{\mathrm{d}\left(\frac{1}{u}\right)}{\mathrm{d}x} = -\frac{1}{u^2}\frac{\mathrm{d}u}{\mathrm{d}x}$$

```
<math xmlns='http://www.w3c.org/mathml' version='2.0'>
  <apply> <eq/>
    <apply> <diff/>
      <bvar> <ci> x </ci> </bvar>
      <apply> <divide/>
        <cn> 1 </cn>
        <ci> u </ci>
      </apply>
    </apply>
    <apply> <times/>
      <apply> <minus/>
        <apply> <divide/>
          <cn> 1 </cn>
          <apply> <power/>
            <ci> u </ci>
            <cn> 2 </cn>
          </apply>
        </apply>
      </apply>
      <apply> <diff/>
        <bvar> <ci> x </ci> </bvar>
        <ci> u </ci>
      </apply>
    </apply>
  </apply>
</math>
```

derivate 11

$$\frac{\mathrm{d}\left(\frac{1}{u^n}\right)}{\mathrm{d}x} = -\frac{n}{(u)^{n+1}}\frac{\mathrm{d}u}{\mathrm{d}x}$$

```
<math xmlns='http://www.w3c.org/mathml' version='2.0'>
  <apply> <eq/>
```

```
    <apply> <diff/>
      <bvar> <ci> x </ci> </bvar>
      <apply> <divide/>
        <cn> 1 </cn>
        <apply> <power/>
          <ci> u </ci>
          <cn> n </cn>
        </apply>
      </apply>
    </apply>
    <apply> <times/>
      <apply> <minus/>
        <apply> <divide/>
          <ci> n </ci>
          <apply> <power/>
            <ci> u </ci>
            <apply> <plus/>
              <ci> n </ci>
              <cn> 1 </cn>
            </apply>
          </apply>
        </apply>
      </apply>
      <apply> <diff/>
        <bvar> <ci> x </ci> </bvar>
        <ci> u </ci>
      </apply>
    </apply>
  </apply>
</math>
```

derivate 43

$$\frac{\mathrm{d}}{\mathrm{d}x} = \frac{\mathrm{d}\log\left(u+\sqrt{u^2+1}\right)}{\mathrm{d}x} = \frac{1}{\sqrt{u^2+1}}\frac{\mathrm{d}u}{\mathrm{d}x}$$

```
<math xmlns='http://www.w3c.org/mathml' version='2.0'>
  <apply> <eq/>
    <apply> <diff/>
      <bvar> <ci> x </ci> </bvar>
      <apply> <inverse/>
        <apply> <sinh/>
          <ci> u </ci>
        </apply>
      </apply>
    </apply>
    <apply> <diff/>
      <bvar> <ci> x </ci> </bvar>
      <apply> <log/>
        <apply> <plus/>
          <ci> u </ci>
          <apply> <root/>
            <apply> <plus/>
              <apply> <power/>
                <ci> u </ci>
                <cn> 2 </cn>
              </apply>
              <cn> 1 </cn>
            </apply>
          </apply>
        </apply>
      </apply>
    </apply>
    <apply> <times/>
      <apply> <divide/>
        <cn> 1 </cn>
        <apply> <root/>
          <apply> <plus/>
            <apply> <power/>
              <ci> u </ci>
              <cn> 2 </cn>
            </apply>
            <cn> 1 </cn>
          </apply>
        </apply>
      </apply>
      <apply> <diff/>
        <bvar> <ci> x </ci> </bvar>
```

```
        <ci> u </ci>
      </apply>
    </apply>
  </apply>
</math>
```

<> *Integrals*

integral 22

$$\int \left(\frac{1}{x\sqrt{a^2 \pm x^2}}\right) \mathrm{d}x = -\frac{1}{a}\log\frac{a+\sqrt{a^2 \pm x^2}}{x}$$

```
<math xmlns='http://www.w3c.org/mathml' version='2.0'>
  <apply> <eq/>
    <apply> <int/>
      <bvar> <ci> x </ci> </bvar>
      <apply> <divide/>
        <cn> 1 </cn>
        <apply> <times/>
          <ci> x </ci>
          <apply> <root/>
            <apply> <fn> <ci> &plusminus; </ci> </fn>
              <apply> <power/>
                <ci> a </ci>
                <cn> 2 </cn>
              </apply>
              <apply> <power/>
                <ci> x </ci>
                <cn> 2 </cn>
              </apply>
            </apply>
          </apply>
        </apply>
      </apply>
    </apply>
    <apply> <minus/>
      <apply> <times/>
        <apply> <divide/>
          <cn> 1 </cn> <ci> a </ci>
        </apply>
        <apply> <log/>
          <apply> <divide/>
            <apply> <plus/>
              <ci> a </ci>
              <apply> <root/>
                <apply> <fn> <ci> &plusminus; </ci> </fn>
                  <apply> <power/>
                    <ci> a </ci>
                    <cn> 2 </cn>
                  </apply>
                  <apply> <power/>
                    <ci> x </ci>
                    <cn> 2 </cn>
                  </apply>
                </apply>
              </apply>
            </apply>
            <ci> x </ci>
          </apply>
        </apply>
      </apply>
    </apply>
  </apply>
</math>
```

integral 61 $$\int \left(\frac{1}{a+bx^2}\right) \mathrm{d}x = \frac{1}{2\sqrt{-ab}} \log \frac{a+x\sqrt{-ab}}{a-x\sqrt{-ab}} \vee \frac{1}{\sqrt{-ab}} \tanh^{-1} \frac{x\sqrt{-ab}}{a}$$

```
<math xmlns='http://www.w3c.org/mathml' version='2.0'>
  <apply> <eq/>
    <apply> <int/>
      <bvar> <ci> x </ci> </bvar>
      <apply> <divide/>
        <cn> 1 </cn>
        <apply> <plus/>
          <ci> a </ci>
          <apply> <times/>
            <ci> b </ci>
            <apply> <power/>
              <ci> x </ci>
              <cn> 2 </cn>
            </apply>
          </apply>
        </apply>
      </apply>
    </apply>
    <apply> <or/>
      <apply> <times/>
        <apply> <divide/>
          <cn> 1 </cn>
          <apply> <times/>
            <cn> 2 </cn>
            <apply> <root/>
              <apply> <minus/>
                <apply> <times/>
                  <ci> a </ci>
                  <ci> b </ci>
                </apply>
              </apply>
            </apply>
          </apply>
        </apply>
        <apply> <log/>
          <apply> <divide/>
            <apply> <plus/>
              <ci> a </ci>
              <apply> <times/>
                <ci> x </ci>
                <apply> <root/>
                  <apply> <minus/>
                    <apply> <times/>
                      <ci> a </ci>
                      <ci> b </ci>
                    </apply>
                  </apply>
                </apply>
              </apply>
            </apply>
            <apply> <minus/>
              <ci> a </ci>
              <apply> <times/>
                <ci> x </ci>
                <apply> <root/>
                  <apply> <minus/>
                    <apply> <times/>
                      <ci> a </ci>
                      <ci> b </ci>
                    </apply>
                  </apply>
                </apply>
              </apply>
            </apply>
          </apply>
        </apply>
      </apply>
      <apply> <times/>
        <apply> <divide/>
          <cn> 1 </cn>
          <apply> <root/>
            <apply> <minus/>
              <apply> <times/>
                <ci> a </ci>
                <ci> b </ci>
              </apply>
            </apply>
          </apply>
        </apply>
        <apply> <power/>
          <apply> <tanh/>
            <apply> <divide/>
              <apply> <times/>
                <ci> x </ci>
                <apply> <root/>
                  <apply> <minus/>
```

```
                    <apply> <times/>
                      <ci> a </ci>
                      <ci> b </ci>
                    </apply>
                  </apply>
                </apply>
              </apply>
              <ci> a </ci>
            </apply>
          </apply>
          <apply> <minus/>
            <cn> 1 </cn>
          </apply>
        </apply>
      </apply>
    </apply>
  </apply>
</math>
```

integral 380

$$\int \left(\frac{1}{\cos\left(ax\right)\left(1\pm\sin\left(ax\right)\right)}\right) \mathrm{d}x = \left(\frac{1}{2a\left(1\pm\sin\left(ax\right)\right)}\right) + \frac{1}{2a}\log\tan\left(\frac{\pi}{4}+\frac{ax}{2}\right)$$

```
<math xmlns='http://www.w3c.org/mathml' version='2.0'>
  <apply> <eq/>
    <apply> <int/>
      <bvar> <ci> x </ci> </bvar>
      <apply> <divide/>
        <cn> 1 </cn>
        <apply> <times/>
          <apply> <cos/>
            <apply> <times/>
              <ci> a </ci>
              <ci> x </ci>
            </apply>
          </apply>
          <apply> <fn> <ci> &plusminus; </ci> </fn>
            <cn> 1 </cn>
            <apply> <sin/>
              <apply> <times/>
                <ci> a </ci>
                <ci> x </ci>
              </apply>
            </apply>
          </apply>
        </apply>
      </apply>
    </apply>
    <apply> <plus/>
      <apply> <fn> <ci> &minusplus; </ci> </fn>
        <apply> <divide/>
          <cn> 1 </cn>
          <apply> <times/>
            <cn> 2 </cn>
            <ci> a </ci>
            <apply> <fn> <ci> &plusminus; </ci> </fn>
              <cn> 1 </cn>
              <apply> <sin/>
                <apply> <times/>
                  <ci> a </ci>
                  <ci> x </ci>
                </apply>
              </apply>
            </apply>
          </apply>
        </apply>
      </apply>
      <apply> <times/>
        <apply> <divide/>
          <cn> 1 </cn>
          <apply> <times/>
            <cn> 2 </cn>
            <ci> a </ci>
          </apply>
        </apply>
        <apply> <log/>
          <apply> <tan/>
```

```
            <apply> <plus/>
              <apply> <divide/>
                <ci> &pi; </ci>
                <cn> 4 </cn>
              </apply>
              <apply> <divide/>
                <apply> <times/>
                  <ci> a </ci>
                  <ci> x </ci>
                </apply>
                <cn> 2 </cn>
              </apply>
            </apply>
          </apply>
        </apply>
      </apply>
    </apply>
  </apply>
</math>
```

<> *Series*

serie 1

$$1-\frac{1}{3}+\frac{1}{5}-\frac{1}{7}+\ldots=\frac{\pi}{4}$$

```
<math xmlns='http://www.w3c.org/mathml' version='2.0'>
  <apply> <eq/>
    <apply> <plus/>
      <cn> 1 </cn>
      <apply> <minus/>
        <apply> <divide/>
          <cn> 1 </cn>
          <cn> 3 </cn>
        </apply>
      </apply>
      <apply> <divide/>
        <cn> 1 </cn>
        <cn> 5 </cn>
      </apply>
      <apply> <minus/>
        <apply> <divide/>
          <cn> 1 </cn>
          <cn> 7 </cn>
        </apply>
      </apply>
      <ci> &cdots; </ci>
    </apply>
    <apply> <divide/>
      <ci> &pi; </ci>
      <cn> 4 </cn>
    </apply>
 </apply>
</math>
```

serie 2

$$1+\frac{1}{2^2}+\frac{1}{3^2}+\frac{1}{4^2}+\dots=\frac{\pi^2}{6}$$

```
<math xmlns='http://www.w3c.org/mathml' version='2.0'>
  <apply> <eq/>
    <apply> <plus/>
      <cn> 1 </cn>
      <apply> <divide/>
        <cn> 1 </cn>
        <apply> <power/>
          <cn> 2 </cn>
          <cn> 2 </cn>
        </apply>
      </apply>
      <apply> <divide/>
        <cn> 1 </cn>
        <apply> <power/>
          <cn> 3 </cn>
          <cn> 2 </cn>
        </apply>
      </apply>
      <apply> <divide/>
        <cn> 1 </cn>
        <apply> <power/>
          <cn> 4 </cn>
          <cn> 2 </cn>
        </apply>
      </apply>
      <ci> &cdots; </ci>
    </apply>
    <apply> <divide/>
      <apply> <power/>
        <ci> &pi; </ci>
        <cn> 2 </cn>
      </apply>
      <cn> 6 </cn>
    </apply>
  </apply>
</math>
```

serie 3

$$1 - \frac{1}{2^2} + \frac{1}{3^2} - \frac{1}{4^2} + \ldots = \frac{\pi^2}{12}$$

```
<math xmlns='http://www.w3c.org/mathml' version='2.0'>
  <apply> <eq/>
    <apply> <plus/>
      <cn> 1 </cn>
      <apply> <minus/>
        <apply> <divide/>
          <cn> 1 </cn>
          <apply> <power/>
            <cn> 2 </cn>
            <cn> 2 </cn>
          </apply>
        </apply>
      </apply>
      <apply> <divide/>
        <cn> 1 </cn>
        <apply> <power/>
          <cn> 3 </cn>
          <cn> 2 </cn>
        </apply>
      </apply>
      <apply> <minus/>
        <apply> <divide/>
          <cn> 1 </cn>
          <apply> <power/>
            <cn> 4 </cn>
            <cn> 2 </cn>
          </apply>
        </apply>
      </apply>
      <ci> &cdots; </ci>
    </apply>
    <apply> <divide/>
      <apply> <power/>
        <ci> &pi; </ci>
        <cn> 2 </cn>
      </apply>
      <cn> 12 </cn>
    </apply>
  </apply>
</math>
```

serie 1 $\forall\, x \in \mathbb{R} \,\Big|\, \mathrm{e}^{x} = 1 + x + \frac{x^{2}}{2!} + \frac{x^{3}}{3!} + \ldots + \frac{x^{n}}{n!} + \ldots$

```
<math xmlns='http://www.w3c.org/mathml' version='2.0'>
  <apply> <forall/>
    <condition>
      <apply> <in/>
        <ci> x </ci>
        <ci> &reals; </ci>
      </apply>
    </condition>
    <apply> <eq/>
      <apply> <power/>
        <ci> &exponentiale; </ci>
        <ci> x </ci>
      </apply>
      <apply> <plus/>
```

```
        <cn> 1 </cn>
        <ci> x </ci>
        <apply> <divide/>
          <apply> <power/>
            <ci> x </ci>
            <cn> 2 </cn>
          </apply>
          <apply> <factorial/>
            <cn> 2 </cn>
          </apply>
        </apply>
        <apply> <divide/>
          <apply> <power/>
            <ci> x </ci>
            <cn> 3 </cn>
          </apply>
          <apply> <factorial/>
            <cn> 3 </cn>
          </apply>
        </apply>
        <ci> &cdots; </ci>
        <apply> <divide/>
          <apply> <power/>
            <ci> x </ci>
            <ci> n </ci>
          </apply>
          <apply> <factorial/>
            <ci> n </ci>
          </apply>
        </apply>
        <ci> &cdots; </ci>
      </apply>
    </apply>
  </apply>
</math>
```

serie 2 $$\forall\, x \in \mathbb{R} \;\Big|\; (\mathrm{e})^{-x} = 1 - x + \frac{x^2}{2!} - \frac{x^3}{3!} + \ldots + (-1)^n \frac{x^n}{n!} \ldots$$

```
<math xmlns='http://www.w3c.org/mathml' version='2.0'>
  <apply> <forall/>
    <condition>
      <apply> <in/>
        <ci> x </ci>
        <ci> &reals; </ci>
      </apply>
    </condition>
    <apply> <eq/>
      <apply> <power/>
        <ci> &exponentiale; </ci>
        <apply> <minus/>
          <ci> x </ci>
        </apply>
      </apply>
      <apply> <plus/>
        <cn> 1 </cn>
        <apply> <minus/>
          <ci> x </ci>
        </apply>
        <apply> <divide/>
          <apply> <power/>
            <ci> x </ci>
            <cn> 2 </cn>
          </apply>
          <apply> <factorial/>
            <cn> 2 </cn>
          </apply>
        </apply>
        <apply> <minus/>
          <apply> <divide/>
            <apply> <power/>
              <ci> x </ci>
              <cn> 3 </cn>
            </apply>
            <apply> <factorial/>
              <cn> 3 </cn>
            </apply>
          </apply>
        </apply>
        <ci> &cdots; </ci>
        <apply> <times/>
          <apply> <power/>
            <apply> <minus/>
              <cn> 1 </cn>
            </apply>
            <ci> n </ci>
          </apply>
          <apply> <divide/>
            <apply> <power/>
              <ci> x </ci>
              <ci> n </ci>
            </apply>
            <apply> <factorial/>
              <ci> n </ci>
            </apply>
          </apply>
          <ci> &cdots; </ci>
        </apply>
      </apply>
    </apply>
  </apply>
</math>
```

<> Logs

log 1 $\forall\, a > 0 \land b > 0 \,\Big|\, \log_g ab = \log_g a + \log_g b$

```
<math xmlns='http://www.w3c.org/mathml' version='2.0'>
  <apply> <forall/>
    <condition>
      <apply> <and/>
        <apply> <gt/>
```

```
          <ci> a </ci>
          <cn> 0 </cn>
        </apply>
        <apply> <gt/>
          <ci> b </ci>
          <cn> 0 </cn>
        </apply>
      </apply>
    </condition>
    <apply> <eq/>
      <apply> <log/>
        <logbase> <ci> g </ci> </logbase>
        <apply> <times/>
          <ci> a </ci>
          <ci> b </ci>
        </apply>
      </apply>
      <apply> <plus/>
        <apply> <log/>
          <logbase> <ci> g </ci> </logbase>
          <ci> a </ci>
        </apply>
        <apply> <log/>
          <logbase> <ci> g </ci> </logbase>
          <ci> b </ci>
        </apply>
      </apply>
    </apply>
  </apply>
</math>
```

log 2 $\forall\, a > 0 \wedge b > 0 \,\Big|\, \log_g \frac{a}{b} = \log_g a - \log_g b$

```
<math xmlns='http://www.w3c.org/mathml' version='2.0'>
  <apply> <forall/>
    <condition>
      <apply> <and/>
        <apply> <gt/>
```

```
          <ci> a </ci>
          <cn> 0 </cn>
        </apply>
        <apply> <gt/>
          <ci> b </ci>
          <cn> 0 </cn>
        </apply>
      </apply>
    </condition>
    <apply> <eq/>
      <apply> <log/>
        <logbase> <ci> g </ci> </logbase>
        <apply> <divide/>
          <ci> a </ci>
          <ci> b </ci>
        </apply>
      </apply>
      <apply> <minus/>
        <apply> <log/>
          <logbase> <ci> g </ci> </logbase>
          <ci> a </ci>
        </apply>
        <apply> <log/>
          <logbase> <ci> g </ci> </logbase>
          <ci> b </ci>
        </apply>
      </apply>
    </apply>
  </apply>
</math>
```

log 3

$$\forall b \in \mathbb{R} \wedge a > 0 \,\Big|\, \log_g a^b = b \log_g a$$

```
<math xmlns='http://www.w3c.org/mathml' version='2.0'>
  <apply> <forall/>
    <condition>
      <apply> <and/>
        <apply> <in/>
```

```
          <ci> b </ci>
          <ci> &reals; </ci>
        </apply>
        <apply> <gt/>
          <ci> a </ci>
          <cn> 0 </cn>
        </apply>
      </apply>
    </condition>
    <apply> <eq/>
      <apply> <log/>
        <logbase> <ci> g </ci> </logbase>
        <apply> <power/>
          <ci> a </ci>
          <ci> b </ci>
        </apply>
      </apply>
      <apply> <times/>
        <ci> b </ci>
        <apply> <log/>
          <logbase> <ci> g </ci> </logbase>
          <ci> a </ci>
        </apply>
      </apply>
    </apply>
  </apply>
</math>
```

log 4

$$\forall\, a > 0 \;\Big|\; \log_g a = \frac{\log_p a}{\log_p g}$$

```
<math xmlns='http://www.w3c.org/mathml' version='2.0'>
  <apply> <forall/>
    <condition>
      <apply> <and/>
        <apply> <gt/>
          <ci> a </ci>
          <cn> 0 </cn>
        </apply>
      </apply>
    </condition>
    <apply> <eq/>
      <apply> <log/>
        <logbase> <ci> g </ci> </logbase>
        <ci> a </ci>
      </apply>
      <apply> <divide/>
        <apply> <log/>
          <logbase> <ci> p </ci> </logbase>
          <ci> a </ci>
        </apply>
        <apply> <log/>
          <logbase> <ci> p </ci> </logbase>
          <ci> g </ci>
        </apply>
      </apply>
    </apply>
  </apply>
</math>
```

<> Goniometrics

gonio 1

$$\sin(x + y) = \sin x \cos y + \cos x \sin y$$

```
<math xmlns='http://www.w3c.org/mathml' version='2.0'>
  <apply> <eq/>
    <apply> <sin/>
      <apply> <plus/>
        <ci> x </ci>
        <ci> y </ci>
      </apply>
    </apply>
    <apply> <plus/>
      <apply> <times/>
        <apply> <sin/>
          <ci> x </ci>
        </apply>
    <apply> <cos/>
      <ci> y </ci>
    </apply>
   </apply>
   <apply> <times/>
    <apply> <cos/>
     <ci> x </ci>
    </apply>
    <apply> <sin/>
     <ci> y </ci>
    </apply>
   </apply>
  </apply>
 </apply>
</math>
```

gonio 2 $$\sin(x - y) = \sin x \cos y - \cos x \sin y$$

```
<math xmlns='http://www.w3c.org/mathml' version='2.0'>
 <apply> <eq/>
  <apply> <sin/>
   <apply> <minus/>
    <ci> x </ci>
    <ci> y </ci>
   </apply>
  </apply>
  <apply> <minus/>
   <apply> <times/>
    <apply> <sin/>
     <ci> x </ci>
    </apply>
    <apply> <cos/>
     <ci> y </ci>
    </apply>
   </apply>
   <apply> <times/>
    <apply> <cos/>
     <ci> x </ci>
```

```
        </apply>
        <apply> <sin/>
          <ci> y </ci>
        </apply>
      </apply>
    </apply>
  </apply>
</math>
```

gonio 3 $\cos\left(x+y\right)=\cos x\cos y-\sin x\sin y$

```
<math xmlns='http://www.w3c.org/mathml' version='2.0'>
  <apply> <eq/>
    <apply> <cos/>
      <apply> <plus/>
        <ci> x </ci>
        <ci> y </ci>
      </apply>
    </apply>
    <apply> <minus/>
      <apply> <times/>
        <apply> <cos/>
          <ci> x </ci>
        </apply>
        <apply> <cos/>
          <ci> y </ci>
        </apply>
      </apply>
      <apply> <times/>
        <apply> <sin/>
          <ci> x </ci>
        </apply>
        <apply> <sin/>
          <ci> y </ci>
        </apply>
      </apply>
    </apply>
  </apply>
```

```
</math>
```

gonio 4

$$\cos\left(x - y\right) = \cos x \cos y + \sin x \sin y$$

```
<math xmlns='http://www.w3c.org/mathml' version='2.0'>
  <apply> <eq/>
    <apply> <cos/>
      <apply> <minus/>
        <ci> x </ci>
        <ci> y </ci>
      </apply>
    </apply>
    <apply> <plus/>
      <apply> <times/>
        <apply> <cos/>
          <ci> x </ci>
        </apply>
        <apply> <cos/>
          <ci> y </ci>
        </apply>
      </apply>
      <apply> <times/>
        <apply> <sin/>
          <ci> x </ci>
        </apply>
        <apply> <sin/>
          <ci> y </ci>
        </apply>
      </apply>
    </apply>
  </apply>
</math>
```

gonio 5

$$\tan\left(x + y\right) = \frac{\tan x + \tan y}{1 - \tan x \tan y}$$

```
<math xmlns='http://www.w3c.org/mathml' version='2.0'>
  <apply> <eq/>
    <apply> <tan/>
      <apply> <plus/>
```

```
        <ci> x </ci>
        <ci> y </ci>
      </apply>
    </apply>
    <apply> <divide/>
      <apply> <plus/>
        <apply> <tan/>
          <ci> x </ci>
        </apply>
        <apply> <tan/>
          <ci> y </ci>
        </apply>
      </apply>
      <apply> <minus/>
        <cn> 1 </cn>
        <apply> <times/>
          <apply> <tan/>
            <ci> x </ci>
          </apply>
          <apply> <tan/>
            <ci> y </ci>
          </apply>
        </apply>
      </apply>
    </apply>
  </apply>
</math>
```

gonio 6

$$\tan\left(x - y\right) = \frac{\tan x - \tan y}{1 + \tan x \tan y}$$

```
<math xmlns='http://www.w3c.org/mathml' version='2.0'>
  <apply> <eq/>
    <apply> <tan/>
      <apply> <minus/>
        <ci> x </ci>
        <ci> y </ci>
      </apply>
    </apply>
    <apply> <divide/>
      <apply> <minus/>
        <apply> <tan/>
          <ci> x </ci>
        </apply>
        <apply> <tan/>
          <ci> y </ci>
        </apply>
      </apply>
      <apply> <plus/>
        <cn> 1 </cn>
        <apply> <times/>
          <apply> <tan/>
            <ci> x </ci>
          </apply>
          <apply> <tan/>
            <ci> y </ci>
          </apply>
        </apply>
      </apply>
    </apply>
  </apply>
</math>
```

gonio 7

$$\sin p + \sin q = 2 \sin \frac{p+q}{2} \cos \frac{p-q}{2}$$

```
<math xmlns='http://www.w3c.org/mathml' version='2.0'>
  <apply> <eq/>
    <apply> <plus/>
      <apply> <sin/>
        <ci> p </ci>
      </apply>
      <apply> <sin/>
        <ci> q </ci>
      </apply>
    </apply>
    <apply> <times/>
      <cn> 2 </cn>
      <apply> <sin/>
        <apply> <divide/>
          <apply> <plus/>
            <ci> p </ci>
            <ci> q </ci>
          </apply>
          <cn> 2 </cn>
        </apply>
      </apply>
      <apply> <cos/>
        <apply> <divide/>
          <apply> <minus/>
            <ci> p </ci>
            <ci> q </ci>
          </apply>
          <cn> 2 </cn>
        </apply>
      </apply>
    </apply>
  </apply>
</math>
```

gonio 8

$$\sin p - \sin q = 2\cos\frac{p+q}{2}\sin\frac{p-q}{2}$$

```
<math xmlns='http://www.w3c.org/mathml' version='2.0'>
  <apply> <eq/>
    <apply> <minus/>
      <apply> <sin/>
        <ci> p </ci>
      </apply>
      <apply> <sin/>
        <ci> q </ci>
      </apply>
    </apply>
    <apply> <times/>
      <cn> 2 </cn>
      <apply> <cos/>
        <apply> <divide/>
```

```
          <apply> <plus/>
            <ci> p </ci>
            <ci> q </ci>
          </apply>
          <cn> 2 </cn>
        </apply>
      </apply>
      <apply> <sin/>
        <apply> <divide/>
          <apply> <minus/>
            <ci> p </ci>
            <ci> q </ci>
          </apply>
          <cn> 2 </cn>
        </apply>
      </apply>
    </apply>
  </apply>
</math>
```

gonio 9

$$\cos p + \cos q = 2\cos\frac{p+q}{2}\cos\frac{p-q}{2}$$

```
<math xmlns='http://www.w3c.org/mathml' version='2.0'>
  <apply> <eq/>
    <apply> <plus/>
      <apply> <cos/>
        <ci> p </ci>
      </apply>
      <apply> <cos/>
        <ci> q </ci>
      </apply>
    </apply>
    <apply> <times/>
      <cn> 2 </cn>
      <apply> <cos/>
        <apply> <divide/>
          <apply> <plus/>
            <ci> p </ci>
            <ci> q </ci>
          </apply>
          <cn> 2 </cn>
        </apply>
      </apply>
      <apply> <cos/>
        <apply> <divide/>
          <apply> <minus/>
            <ci> p </ci>
            <ci> q </ci>
          </apply>
          <cn> 2 </cn>
        </apply>
      </apply>
    </apply>
  </apply>
</math>
```

gonio 10

$$\cos p - \cos q = -2 \sin \frac{p+q}{2} \sin \frac{p-q}{2}$$

```
<math xmlns='http://www.w3c.org/mathml' version='2.0'>
  <apply> <eq/>
    <apply> <minus/>
      <apply> <cos/>
        <ci> p </ci>
      </apply>
      <apply> <cos/>
        <ci> q </ci>
      </apply>
    </apply>
    <apply> <minus/>
      <apply> <times/>
        <cn> 2 </cn>
        <apply> <sin/>
          <apply> <divide/>
             <apply> <plus/>
              <ci> p </ci>
              <ci> q </ci>
```

```
            </apply>
            <cn> 2 </cn>
          </apply>
        </apply>
        <apply> <sin/>
          <apply> <divide/>
            <apply> <minus/>
              <ci> p </ci>
              <ci> q </ci>
            </apply>
            <cn> 2 </cn>
          </apply>
        </apply>
      </apply>
    </apply>
  </apply>
</math>
```

gonio 11 $$2\sin\alpha\cos\beta = \sin(\alpha+\beta) + \sin(\alpha-\beta)$$

```
<math xmlns='http://www.w3c.org/mathml' version='2.0'>
  <apply> <eq/>
    <apply> <times/>
      <cn> 2 </cn>
      <apply> <sin/>
        <ci> &alpha; </ci>
      </apply>
      <apply> <cos/>
        <ci> &beta; </ci>
      </apply>
    </apply>
    <apply> <plus/>
      <apply> <sin/>
        <apply> <plus/>
          <ci> &alpha; </ci>
          <ci> &beta; </ci>
        </apply>
      </apply>
      <apply> <sin/>
        <apply> <minus/>
          <ci> &alpha; </ci>
          <ci> &beta; </ci>
        </apply>
      </apply>
    </apply>
  </apply>
</math>
```

gonio 12 $$2\cos\alpha\sin\beta = \sin(\alpha+\beta) - \sin(\alpha-\beta)$$

```
<math xmlns='http://www.w3c.org/mathml' version='2.0'>
  <apply> <eq/>
    <apply> <times/>
      <cn> 2 </cn>
      <apply> <cos/>
        <ci> &alpha; </ci>
      </apply>
      <apply> <sin/>
        <ci> &beta; </ci>
      </apply>
    </apply>
    <apply> <minus/>
      <apply> <sin/>
        <apply> <plus/>
          <ci> &alpha; </ci>
          <ci> &beta; </ci>
        </apply>
      </apply>
      <apply> <sin/>
        <apply> <minus/>
          <ci> &alpha; </ci>
          <ci> &beta; </ci>
        </apply>
      </apply>
    </apply>
  </apply>
```

```
</math>
```

gonio 13 $2\cos\alpha\cos\beta = \cos(\alpha+\beta)+\cos(\alpha-\beta)$

```
<math xmlns='http://www.w3c.org/mathml' version='2.0'>
  <apply> <eq/>
    <apply> <times/>
      <cn> 2 </cn>
      <apply> <cos/>
        <ci> &alpha; </ci>
      </apply>
      <apply> <cos/>
        <ci> &beta; </ci>
      </apply>
    </apply>
    <apply> <plus/>
      <apply> <cos/>
        <apply> <plus/>
          <ci> &alpha; </ci>
          <ci> &beta; </ci>
        </apply>
      </apply>
      <apply> <cos/>
        <apply> <minus/>
          <ci> &alpha; </ci>
          <ci> &beta; </ci>
        </apply>
      </apply>
    </apply>
  </apply>
</math>
```

gonio 14 $-2\sin\alpha\cos\beta = \sin(\alpha+\beta)-\sin(\alpha-\beta)$

```
<math xmlns='http://www.w3c.org/mathml' version='2.0'>
  <apply> <eq/>
    <apply> <minus/>
      <apply> <times/>
        <cn> 2 </cn>
        <apply> <sin/>
          <ci> &alpha; </ci>
        </apply>
        <apply> <cos/>
          <ci> &beta; </ci>
        </apply>
      </apply>
    </apply>
    <apply> <minus/>
      <apply> <sin/>
        <apply> <plus/>
          <ci> &alpha; </ci>
          <ci> &beta; </ci>
        </apply>
      </apply>
      <apply> <sin/>
        <apply> <minus/>
          <ci> &alpha; </ci>
          <ci> &beta; </ci>
        </apply>
      </apply>
    </apply>
  </apply>
</math>
```

gonio 15

$$\forall \,\triangle ABC \left|\, \frac{a}{\sin\alpha} + \frac{b}{\sin\beta} + \frac{c}{\sin\gamma} \right.$$

```
<math xmlns='http://www.w3c.org/mathml' version='2.0'>
  <apply> <forall/>
    <condition>
      <mrow>
        <mi> &bigtriangleup; </mi>
        <mi> A </mi>
        <mi> B </mi>
        <mi> C </mi>
      </mrow>
    </condition>
    <apply> <plus/>
```

```
      <apply> <divide/>
        <ci> a </ci>
        <apply> <sin/>
          <ci> &alpha; </ci>
        </apply>
      </apply>
      <apply> <divide/>
        <ci> b </ci>
        <apply> <sin/>
          <ci> &beta; </ci>
        </apply>
      </apply>
      <apply> <divide/>
        <ci> c </ci>
        <apply> <sin/>
          <ci> &gamma; </ci>
        </apply>
      </apply>
    </apply>
  </apply>
</math>
```

gonio 16

$$\forall\,\triangle ABC\;\left|\begin{array}{l} a^2 = b^2 + c^2 - 2bc\cos\alpha \\ b^2 = a^2 + c^2 - 2ac\cos\beta \\ c^2 = a^2 + b^2 - 2ab\cos\gamma \end{array}\right.$$

```
<math xmlns='http://www.w3c.org/mathml' version='2.0'>
  <apply> <forall/>
    <condition>
      <mrow>
        <mi> &bigtriangleup; </mi>
        <mi> A </mi>
        <mi> B </mi>
        <mi> C </mi>
      </mrow>
    </condition>
    <apply> <eq/>
      <apply> <power/>
        <ci> a </ci>
```

```
   <cn> 2 </cn>
  </apply>
  <apply> <plus/>
   <apply> <power/>
    <ci> b </ci>
    <cn> 2 </cn>
   </apply>
   <apply> <power/>
    <ci> c </ci>
    <cn> 2 </cn>
   </apply>
   <apply> <minus/>
    <apply> <times/>
     <cn> 2 </cn>
     <ci> b </ci>
     <ci> c </ci>
     <apply> <cos/>
      <ci> &alpha; </ci>
     </apply>
    </apply>
   </apply>
  </apply>
 </apply>
<apply> <eq/>
  <apply> <power/>
   <ci> b </ci>
   <cn> 2 </cn>
  </apply>
  <apply> <plus/>
   <apply> <power/>
    <ci> a </ci>
    <cn> 2 </cn>
   </apply>
   <apply> <power/>
    <ci> c </ci>
    <cn> 2 </cn>
   </apply>
   <apply> <minus/>
```

```
        <apply> <times/>
          <cn> 2 </cn>
          <ci> a </ci>
          <ci> c </ci>
          <apply> <cos/>
            <ci> &beta; </ci>
          </apply>
        </apply>
      </apply>
    </apply>
  </apply>
 <apply> <eq/>
    <apply> <power/>
      <ci> c </ci>
      <cn> 2 </cn>
    </apply>
    <apply> <plus/>
      <apply> <power/>
        <ci> a </ci>
        <cn> 2 </cn>
      </apply>
      <apply> <power/>
        <ci> b </ci>
        <cn> 2 </cn>
      </apply>
      <apply> <minus/>
        <apply> <times/>
          <cn> 2 </cn>
          <ci> a </ci>
          <ci> b </ci>
          <apply> <cos/>
            <ci> &gamma; </ci>
          </apply>
        </apply>
      </apply>
    </apply>
  </apply>
</apply>
```

```
</math>
```

<> Statistics

statistic 1

$$\overline{x} = \frac{1}{n} \sum x_i$$

```
<math xmlns='http://www.w3c.org/mathml' version='2.0'>
  <apply> <eq/>
    <apply> <mean/>
      <ci> x </ci>
    </apply>
    <apply> <times/>
      <apply> <divide/>
        <cn> 1 </cn>
        <ci> n </ci>
      </apply>
      <apply> <sum/>
        <ci> <msub> <mi> x </mi> <mi> i </mi> </msub> </ci>
      </apply>
    </apply>
  </apply>
</math>
```

statistic 2

$$\sigma\left(x\right) \approx \sqrt{\frac{\sum\left(x_i - \overline{x}\right)^2}{n-1}}$$

```
<math xmlns='http://www.w3c.org/mathml' version='2.0'>
  <apply> <approx/>
    <apply> <sdev/>
      <ci> x </ci>
    </apply>
    <apply> <root/>
      <apply> <divide/>
        <apply> <sum/>
          <apply> <power/>
            <apply> <minus/>
              <ci> <msub> <mi> x </mi> <mi> i </mi> </msub> </ci>
              <apply> <mean/>
```

```
                <ci> x </ci>
              </apply>
            </apply>
            <cn> 2 </cn>
          </apply>
        </apply>
        <apply> <minus/>
          <ci> n </ci>
          <cn> 1 </cn>
        </apply>
      </apply>
    </apply>
  </apply>
</math>
```

statistic 3

$$\sigma\,(x)^2 \approx \overline{(x_i-\overline{x})^2} = \frac{1}{n-1}\sum(x_i-\overline{x})^2$$

```
<math xmlns='http://www.w3c.org/mathml' version='2.0'>
  <apply> <approx/>
    <apply> <variance/>
      <ci> x </ci>
    </apply>
    <apply> <eq/>
      <apply> <mean/>
        <apply> <power/>
          <apply> <minus/>
            <ci> <msub> <mi> x </mi> <mi> i </mi> </msub> </ci>
            <apply> <mean/>
              <ci> x </ci>
            </apply>
          </apply>
          <cn> 2 </cn>
        </apply>
      </apply>
      <apply> <times/>
        <apply> <divide/>
          <cn> 1 </cn>
          <apply> <minus/>
            <ci> n </ci>
            <cn> 1 </cn>
          </apply>
        </apply>
        <apply> <sum/>
          <apply> <power/>
            <apply> <minus/>
              <ci> <msub> <mi> x </mi> <mi> i </mi> </msub> </ci>
              <apply> <mean/>
                <ci> x </ci>
              </apply>
            </apply>
            <cn> 2 </cn>
          </apply>
        </apply>
      </apply>
    </apply>
  </apply>
</math>
```

<> *Matrices*

matrix 1

$$\begin{vmatrix} \sin\alpha & \cos\alpha \\ \sin\beta & \cos\beta \end{vmatrix} = \sin(\alpha - \beta)$$

```
<math xmlns='http://www.w3c.org/mathml' version='2.0'>
  <apply> <eq/>
    <apply> <determinant/>
      <matrix>
        <matrixrow>
          <apply> <sin/> <ci> &alpha; </ci> </apply>
          <apply> <cos/> <ci> &alpha; </ci> </apply>
        </matrixrow>
        <matrixrow>
          <apply> <sin/> <ci> &beta; </ci> </apply>
          <apply> <cos/> <ci> &beta; </ci> </apply>
        </matrixrow>
      </matrix>
    </apply>
    <apply> <sin/>
      <apply> <minus/>
        <ci> &alpha; </ci>
        <ci> &beta; </ci>
      </apply>
    </apply>
  </apply>
</math>
```

matrix 2

$$|I| = \begin{vmatrix} 1 & 0 \\ 0 & 1 \end{vmatrix} = 1$$

```
<math xmlns='http://www.w3c.org/mathml' version='2.0'>
  <apply> <eq/>
    <apply> <determinant/>
     <ci> I </ci>
    </apply>
    <apply> <determinant/>
      <matrix>
        <matrixrow> <cn> 1 </cn> <cn> 0 </cn> </matrixrow>
        <matrixrow> <cn> 0 </cn> <cn> 1 </cn> </matrixrow>
      </matrix>
    </apply>
    <cn> 1 </cn>
  </apply>
</math>
```

Unicode Math

Support for MathML showed up in ConTEXt by the end of second millenium. The first more or less complete version of this manual dates from the end of 1999. At that time Unicode math was no fact yet and entities were the way to get special symbols done. Mapping the names of symbols onto something that could be rendered was up to the xml processors and typesetting engine.

Nowadays we can use Unicode directly although it has the drawback that not all editing applications show the corresponding shapes. It is for this reason that entities will have their use for a while. In the next table we see the official ones. The table is actually larger, but we only show the shapes that have a math property in the ConTEXt character database. The full list is supported and can be found in the following documents:

```
http://www.w3.org/2003/entities/2007/w3centities-f.ent
http://www.w3.org/2003/entities/2007/htmlmathml-f.ent
```

Α	00391	Agr
Α	00391	Alpha
≔	02254	Assign
∖	02216	Backslash
∵	02235	Because
Β	00392	Beta
Β	00392	Bgr
˘	002D8	Breve
≎	0224E	Bumpeq
⋒	022D2	Cap
	02145	CapitalDifferentialD
∰	02230	Cconint
·	000B7	CenterDot
Χ	003A7	Chi
⊙	02299	CircleDot
⊖	02296	CircleMinus
⊕	02295	CirclePlus
⊗	02297	CircleTimes
∲	02232	ClockwiseContourIntegral
∷	02237	Colon
≡	02261	Congruent
∯	0222F	Conint
∮	0222E	ContourIntegral
ℂ	02102	Copf
∐	02210	Coproduct
∳	02233	CounterClockwiseContourIntegral
⋓	022D3	Cup
≍	0224D	CupCap
	02145	DD
	02911	DDotrahd
‡	02021	Dagger
↡	021A1	Darr
Δ	00394	Delta
Δ	00394	Dgr
´	000B4	DiacriticalAcute
˙	002D9	DiacriticalDot
`	00060	DiacriticalGrave
˜	002DC	DiacriticalTilde
⋄	022C4	Diamond
	02146	DifferentialD
¨	000A8	Dot
∯	0222F	DoubleContourIntegral
¨	000A8	DoubleDot
⇓	021D3	DoubleDownArrow
⇐	021D0	DoubleLeftArrow
⇔	021D4	DoubleLeftRightArrow

	027F8	DoubleLongLeftArrow	^	0005E	Hat
	027FA	DoubleLongLeftRightArrow	≎	0224E	HumpDownHump
	027F9	DoubleLongRightArrow	ℑ	02111	Ifr
⇒	021D2	DoubleRightArrow	Ι	00399	Igr
⊨	022A8	DoubleRightTee	ℑ	02111	Im
⇑	021D1	DoubleUpArrow		02148	ImaginaryI
⇕	021D5	DoubleUpDownArrow	⇒	021D2	Implies
∥	02225	DoubleVerticalBar	∬	0222C	Int
↓	02193	DownArrow	∫	0222B	Integral
⇵	021F5	DownArrowUpArrow	⋂	022C2	Intersection
↽	021BD	DownLeftVector	Ι	00399	Iota
⇁	021C1	DownRightVector	Χ	003A7	KHgr
⊤	022A4	DownTee	Κ	0039A	Kappa
↧	021A7	DownTeeArrow	Κ	0039A	Kgr
⇓	021D3	Downarrow	<	0003C	LT
Η	00397	EEgr	Λ	0039B	Lambda
Ε	00395	Egr		027EA	Lang
∈	02208	Element	↞	0219E	Larr
Ε	00395	Epsilon	⟨	027E8	LeftAngleBracket
≂	02242	EqualTilde	←	02190	LeftArrow
⇌	021CC	Equilibrium	⇤	021E4	LeftArrowBar
Η	00397	Eta	⇆	021C6	LeftArrowRightArrow
∃	02203	Exists	⌈	02308	LeftCeiling
	02147	ExponentialE	⟦	027E6	LeftDoubleBracket
∀	02200	ForAll	⇃	021C3	LeftDownVector
>	0003E	GT	⌊	0230A	LeftFloor
Γ	00393	Gamma	↔	02194	LeftRightArrow
Ϝ	003DC	Gammad	⊣	022A3	LeftTee
⋙	022D9	Gg	↤	021A4	LeftTeeArrow
Γ	00393	Ggr	⊲	022B2	LeftTriangle
≥	02265	GreaterEqual	↿	021BF	LeftUpVector
⋛	022DB	GreaterEqualLess	↼	021BC	LeftVector
≧	02267	GreaterFullEqual	⇐	021D0	Leftarrow
≷	02277	GreaterLess	⇔	021D4	Leftrightarrow
⩾	02A7E	GreaterSlantEqual	⋚	022DA	LessEqualGreater
≳	02273	GreaterTilde	≦	02266	LessFullEqual
≫	0226B	Gt	≶	02276	LessGreater
ˇ	002C7	Hacek	⩽	02A7D	LessSlantEqual

≲	02272	LessTilde
Λ	0039B	Lgr
⋘	022D8	Ll
⇚	021DA	Lleftarrow
	027F5	LongLeftArrow
	027F7	LongLeftRightArrow
	027F6	LongRightArrow
	027F8	Longleftarrow
	027FA	Longleftrightarrow
	027F9	Longrightarrow
↙	02199	LowerLeftArrow
↘	02198	LowerRightArrow
↰	021B0	Lsh
≪	0226A	Lt
Μ	0039C	Mgr
∓	02213	MinusPlus
Μ	0039C	Mu
≫	0226B	NestedGreaterGreater
≪	0226A	NestedLessLess
Ν	0039D	Ngr
ℕ	02115	Nopf
≢	02262	NotCongruent
≭	0226D	NotCupCap
∦	02226	NotDoubleVerticalBar
∉	02209	NotElement
≠	02260	NotEqual
≂	02242	NotEqualTilde
∄	02204	NotExists
≯	0226F	NotGreater
≱	02271	NotGreaterEqual
≧	02267	NotGreaterFullEqual
≫	0226B	NotGreaterGreater
≹	02279	NotGreaterLess
⩾	02A7E	NotGreaterSlantEqual
≵	02275	NotGreaterTilde
≎	0224E	NotHumpDownHump
⋪	022EA	NotLeftTriangle
⋬	022EC	NotLeftTriangleEqual
≮	0226E	NotLess
≰	02270	NotLessEqual
≸	02278	NotLessGreater
≪	0226A	NotLessLess
⩽	02A7D	NotLessSlantEqual
≴	02274	NotLessTilde
⊀	02280	NotPrecedes
⪯	02AAF	NotPrecedesEqual
⋠	022E0	NotPrecedesSlantEqual
∌	0220C	NotReverseElement
⋫	022EB	NotRightTriangle
⋭	022ED	NotRightTriangleEqual
⊏	0228F	NotSquareSubset
⋢	022E2	NotSquareSubsetEqual
⊐	02290	NotSquareSuperset
⋣	022E3	NotSquareSupersetEqual
⊂	02282	NotSubset
⊈	02288	NotSubsetEqual
⊁	02281	NotSucceeds
⪰	02AB0	NotSucceedsEqual
⋡	022E1	NotSucceedsSlantEqual
≿	0227F	NotSucceedsTilde
⊃	02283	NotSuperset
⊉	02289	NotSupersetEqual
≁	02241	NotTilde
≄	02244	NotTildeEqual
≇	02247	NotTildeFullEqual
≉	02249	NotTildeTilde
∤	02224	NotVerticalBar
Ν	0039D	Nu
Ω	003A9	OHgr
Ο	0039F	Ogr
Ω	003A9	Omega
Ο	0039F	Omicron
⏞	023DE	OverBrace
⏜	023DC	OverParenthesis
Φ	003A6	PHgr
Ψ	003A8	PSgr

∂	02202	PartialD
Π	003A0	Pgr
Φ	003A6	Phi
Π	003A0	Pi
±	000B1	PlusMinus
ℙ	02119	Popf
≺	0227A	Precedes
⪯	02AAF	PrecedesEqual
≼	0227C	PrecedesSlantEqual
≾	0227E	PrecedesTilde
∏	0220F	Product
∷	02237	Proportion
∝	0221D	Proportional
Ψ	003A8	Psi
"	00022	QUOT
ℚ	0211A	Qopf
	027EB	Rang
↠	021A0	Rarr
	02916	Rarrtl
ℜ	0211C	Re
∋	0220B	ReverseElement
⇋	021CB	ReverseEquilibrium
ℜ	0211C	Rfr
Ρ	003A1	Rgr
Ρ	003A1	Rho
⟩	027E9	RightAngleBracket
→	02192	RightArrow
⇥	021E5	RightArrowBar
⇄	021C4	RightArrowLeftArrow
⌉	02309	RightCeiling
⟧	027E7	RightDoubleBracket
⇂	021C2	RightDownVector
⌋	0230B	RightFloor
⊢	022A2	RightTee
↦	021A6	RightTeeArrow
⊳	022B3	RightTriangle
↾	021BE	RightUpVector
⇀	021C0	RightVector
⇒	021D2	Rightarrow
ℝ	0211D	Ropf
⇛	021DB	Rrightarrow
↱	021B1	Rsh
Σ	003A3	Sgr
↓	02193	ShortDownArrow
←	02190	ShortLeftArrow
→	02192	ShortRightArrow
↑	02191	ShortUpArrow
Σ	003A3	Sigma
∘	02218	SmallCircle
√	0221A	Sqrt
□	025A1	Square
⊓	02293	SquareIntersection
⊏	0228F	SquareSubset
⊑	02291	SquareSubsetEqual
⊐	02290	SquareSuperset
⊒	02292	SquareSupersetEqual
⊔	02294	SquareUnion
⋆	022C6	Star
⋐	022D0	Sub
⋐	022D0	Subset
⊆	02286	SubsetEqual
≻	0227B	Succeeds
⪰	02AB0	SucceedsEqual
≽	0227D	SucceedsSlantEqual
≿	0227F	SucceedsTilde
∋	0220B	SuchThat
∑	02211	Sum
⋑	022D1	Sup
⊃	02283	Superset
⊇	02287	SupersetEqual
⋑	022D1	Supset
Θ	00398	THgr
Τ	003A4	Tau
Τ	003A4	Tgr
∴	02234	Therefore
Θ	00398	Theta

∼	0223C	Tilde
≃	02243	TildeEqual
≅	02245	TildeFullEqual
≈	02248	TildeTilde
⃛	020DB	TripleDot
↟	0219F	Uarr
Υ	003A5	Ugr
⏟	023DF	UnderBrace
⏝	023DD	UnderParenthesis
⋃	022C3	Union
⊎	0228E	UnionPlus
↑	02191	UpArrow
⇅	021C5	UpArrowDownArrow
↕	02195	UpDownArrow
⊥	022A5	UpTee
↥	021A5	UpTeeArrow
⇑	021D1	Uparrow
⇕	021D5	Updownarrow
↖	02196	UpperLeftArrow
↗	02197	UpperRightArrow
Υ	003A5	Upsilon
⊫	022AB	VDash
⊩	022A9	Vdash
⋁	022C1	Vee
‖	02016	Verbar
‖	02016	Vert
∣	02223	VerticalBar
|	0007C	VerticalLine
≀	02240	VerticalTilde
⊪	022AA	Vvdash
⋀	022C0	Wedge
Ξ	0039E	Xgr
Ξ	0039E	Xi
Ζ	00396	Zeta
Ζ	00396	Zgr
ℤ	02124	Zopf
´	000B4	acute
α	003B1	agr
ℵ	02135	alefsym
ℵ	02135	aleph
α	003B1	alpha
⨿	02A3F	amalg
∧	02227	and
∠	02220	ang
∠	02220	angle
∡	02221	angmsd
∟	0221F	angrt
∢	02222	angsph
≈	02248	ap
≊	0224A	ape
'	00027	apos
≈	02248	approx
≊	0224A	approxeq
*	0002A	ast
≈	02248	asymp
≍	0224D	asympeq
∳	02233	awconint
≌	0224C	backcong
	003F6	backepsilon
∽	0223D	backsim
≌	0224C	bcong
∵	02235	becaus
∵	02235	because
	003F6	bepsi
β	003B2	beta
ℶ	02136	beth
≬	0226C	between
β	003B2	bgr
⋂	022C2	bigcap
	025EF	bigcirc
⋃	022C3	bigcup
⨁	02A01	bigoplus
⨂	02A02	bigotimes
⨆	02A06	bigsqcup
	02605	bigstar
▽	025BD	bigtriangledown

△	025B3	bigtriangleup	♣	02663	clubs
⨄	02A04	biguplus	♣	02663	clubsuit
⋁	022C1	bigvee	:	0003A	colon
⋀	022C0	bigwedge	≔	02254	colone
	0290D	bkarow	≔	02254	coloneq
	029EB	blacklozenge	,	0002C	comma
▸	025B8	blacktriangleright	∁	02201	comp
=	0003D	bne	∘	02218	compfn
≡	02261	bnequiv	∁	02201	complement
⊥	022A5	bot	ℂ	02102	complexes
⊥	022A5	bottom	≅	02245	cong
⋈	022C8	bowtie	∮	0222E	conint
⊟	0229F	boxminus	∐	02210	coprod
⊞	0229E	boxplus	↵	021B5	crarr
⊠	022A0	boxtimes	⋯	022EF	ctdot
˘	002D8	breve	⋞	022DE	cuepr
∽	0223D	bsim	⋟	022DF	cuesc
\	0005C	bsol	↶	021B6	cularr
•	02022	bull	∪	0222A	cup
•	02022	bullet	∪	0222A	cups
≎	0224E	bump	↷	021B7	curarr
∩	02229	cap	⋞	022DE	curlyeqprec
∩	02229	caps	⋟	022DF	curlyeqsucc
ˇ	002C7	caron	⋎	022CE	curlyvee
…	02026	cdots	⋏	022CF	curlywedge
·	000B7	centerdot	↶	021B6	curvearrowleft
✓	02713	check	↷	021B7	curvearrowright
✓	02713	checkmark	⋎	022CE	cuvee
χ	003C7	chi	⋏	022CF	cuwed
ˆ	002C6	circ	∲	02232	cwconint
≗	02257	circeq	∱	02231	cwint
↺	021BA	circlearrowleft	⇓	021D3	dArr
↻	021BB	circlearrowright	†	02020	dagger
Ⓢ	024C8	circledS	ℸ	02138	daleth
⊛	0229B	circledast	↓	02193	darr
⊚	0229A	circledcirc	⊣	022A3	dashv
⊝	0229D	circleddash		02146	dd
≗	02257	cire	‡	02021	ddagger

⇊	021CA	ddarr
δ	003B4	delta
δ	003B4	dgr
⇃	021C3	dharl
⇂	021C2	dharr
⋄	022C4	diam
⋄	022C4	diamond
¨	000A8	die
÷	000F7	div
÷	000F7	divide
⋇	022C7	divideontimes
⋇	022C7	divonx
	0231E	dlcorn
˙	002D9	dot
≑	02251	doteqdot
∸	02238	dotminus
∔	02214	dotplus
⊡	022A1	dotsquare
↓	02193	downarrow
⇊	021CA	downdownarrows
⇃	021C3	downharpoonleft
⇂	021C2	downharpoonright
	0231F	drcorn
⋱	022F1	dtdot
⇵	021F5	duarr
	027FF	dzigrarr
≑	02251	eDot
≖	02256	ecir
≕	02255	ecolon
	02147	ee
η	003B7	eegr
≒	02252	efDot
ε	003B5	egr
⪖	02A96	egs
ℓ	02113	ell
⪕	02A95	els
∅	02205	empty
∅	02205	emptyset
∅	02205	emptyv
ε	003B5	epsi
ε	003B5	epsilon
ϵ	003F5	epsiv
≖	02256	eqcirc
≕	02255	eqcolon
≂	02242	eqsim
⪖	02A96	eqslantgtr
⪕	02A95	eqslantless
=	0003D	equals
≟	0225F	equest
≡	02261	equiv
≓	02253	erDot
≂	02242	esim
η	003B7	eta
	000F0	eth
!	00021	excl
∃	02203	exist
	02147	exponentiale
≒	02252	fallingdotseq
f	00066	fjlig
♭	0266D	flat
∀	02200	forall
⋔	022D4	fork
⁄	02044	frasl
⌢	02322	frown
≧	02267	gE
⪌	02A8C	gEl
γ	003B3	gamma
⪆	02A86	gap
≥	02265	ge
⋛	022DB	gel
≥	02265	geq
≧	02267	geqq
⩾	02A7E	geqslant
⩾	02A7E	ges
⋛	022DB	gesl
≫	0226B	gg

⋙	022D9	ggg
γ	003B3	ggr
ℷ	02137	gimel
≷	02277	gl
≩	02269	gnE
⪊	02A8A	gnap
⪊	02A8A	gnapprox
⪈	02A88	gne
⪈	02A88	gneq
≩	02269	gneqq
⋧	022E7	gnsim
`	00060	grave
≳	02273	gsim
>	0003E	gt
⋗	022D7	gtdot
⪆	02A86	gtrapprox
⋗	022D7	gtrdot
⋛	022DB	gtreqless
⪌	02A8C	gtreqqless
≷	02277	gtrless
≳	02273	gtrsim
≩	02269	gvertneqq
≩	02269	gvnE
⇔	021D4	hArr
↔	02194	harr
↭	021AD	harrw
ℏ	0210F	hbar
…	02026	hellip
	02925	hksearow
	02926	hkswarow
⇿	021FF	hoarr
↩	021A9	hookleftarrow
↪	021AA	hookrightarrow
ℏ	0210F	hslash
	00127	hstrok
⇔	021D4	iff
ι	003B9	igr
	02148	ii
∭	0222D	iiint
	02129	iiota
ℑ	02111	image
ℑ	02111	imagpart
∈	02208	in
∞	0221E	infin
∫	0222B	int
⊺	022BA	intcal
ℤ	02124	integers
⊺	022BA	intercal
ι	003B9	iota
∈	02208	isin
∈	02208	isinv
κ	003BA	kappa
ϰ	003F0	kappav
κ	003BA	kgr
χ	003C7	khgr
⇚	021DA	lAarr
⇐	021D0	lArr
≦	02266	lE
⪋	02A8B	lEg
λ	003BB	lambda
⟨	027E8	lang
⟨	027E8	langle
⪅	02A85	lap
←	02190	larr
⇤	021E4	larrb
↩	021A9	larrhk
↫	021AB	larrlp
↢	021A2	larrtl
	0290C	lbarr
{	0007B	lbrace
[	0005B	lbrack
⌈	02308	lceil
{	0007B	lcub
↲	021B2	ldsh
≤	02264	le
←	02190	leftarrow

↢	021A2	leftarrowtail
↽	021BD	leftharpoondown
↼	021BC	leftharpoonup
⇇	021C7	leftleftarrows
↔	02194	leftrightarrow
⇆	021C6	leftrightarrows
⇋	021CB	leftrightharpoons
↭	021AD	leftrightsquigarrow
⋋	022CB	leftthreetimes
⋚	022DA	leg
≤	02264	leq
≦	02266	leqq
⩽	02A7D	leqslant
⩽	02A7D	les
⋚	022DA	lesg
⪅	02A85	lessapprox
⋖	022D6	lessdot
⋚	022DA	lesseqgtr
⪋	02A8B	lesseqqgtr
≶	02276	lessgtr
≲	02272	lesssim
⌊	0230A	lfloor
≶	02276	lg
λ	003BB	lgr
↽	021BD	lhard
↼	021BC	lharu
≪	0226A	ll
⇇	021C7	llarr
	0231E	llcorner
	023B0	lmoust
	023B0	lmoustache
≨	02268	lnE
⪉	02A89	lnap
⪉	02A89	lnapprox
⪇	02A87	lne
⪇	02A87	lneq
≨	02268	lneqq
⋦	022E6	lnsim
⇽	021FD	loarr
⟦	027E6	lobrk
	027F5	longleftarrow
	027F7	longleftrightarrow
	027FC	longmapsto
	027F6	longrightarrow
↫	021AB	looparrowleft
↬	021AC	looparrowright
∗	02217	lowast
◊	025CA	loz
◊	025CA	lozenge
	029EB	lozf
(	00028	lpar
⇆	021C6	lrarr
	0231F	lrcorner
⇋	021CB	lrhar
↰	021B0	lsh
≲	02272	lsim
[	0005B	lsqb
<	0003C	lt
⋖	022D6	ltdot
⋋	022CB	lthree
⋉	022C9	ltimes
≨	02268	lvertneqq
≨	02268	lvnE
¯	000AF	macr
✠	02720	malt
✠	02720	maltese
↦	021A6	map
↦	021A6	mapsto
↧	021A7	mapstodown
↤	021A4	mapstoleft
↥	021A5	mapstoup
∡	02221	measuredangle
μ	003BC	mgr
℧	02127	mho
∣	02223	mid
*	0002A	midast

·	000B7	middot
−	02212	minus
⊟	0229F	minusb
∸	02238	minusd
∓	02213	minusplus
…	02026	mldr
∓	02213	mnplus
⊧	022A7	models
∓	02213	mp
μ	003BC	mu
⊸	022B8	multimap
⊸	022B8	mumap
⋙	022D9	nGg
≫	0226B	nGt
≫	0226B	nGtv
⇍	021CD	nLeftarrow
⇎	021CE	nLeftrightarrow
⋘	022D8	nLl
≪	0226A	nLt
≪	0226A	nLtv
⇏	021CF	nRightarrow
⊯	022AF	nVDash
⊮	022AE	nVdash
∠	02220	nang
≉	02249	nap
≉	02249	napprox
♮	0266E	natur
♮	0266E	natural
ℕ	02115	naturals
≎	0224E	nbump
≇	02247	ncong
≠	02260	ne
⇗	021D7	neArr
	02924	nearhk
↗	02197	nearr
↗	02197	nearrow
≢	02262	nequiv
≂	02242	nesim
∄	02204	nexist
∄	02204	nexists
≧	02267	ngE
≱	02271	nge
≱	02271	ngeq
≧	02267	ngeqq
⩾	02A7E	ngeqslant
⩾	02A7E	nges
ν	003BD	ngr
≵	02275	ngsim
≯	0226F	ngt
≯	0226F	ngtr
⇎	021CE	nhArr
↮	021AE	nharr
∋	0220B	ni
∋	0220B	niv
⇍	021CD	nlArr
≦	02266	nlE
↚	0219A	nlarr
≰	02270	nle
↚	0219A	nleftarrow
↮	021AE	nleftrightarrow
≰	02270	nleq
≦	02266	nleqq
⩽	02A7D	nleqslant
⩽	02A7D	nles
≮	0226E	nless
≴	02274	nlsim
≮	0226E	nlt
⋪	022EA	nltri
⋬	022EC	nltrie
∤	02224	nmid
¬	000AC	not
∉	02209	notin
∉	02209	notinva
∌	0220C	notni
∌	0220C	notniva
∦	02226	npar

∦	02226	nparallel
∂	02202	npart
⊀	02280	npr
⋠	022E0	nprcue
⪯	02AAF	npre
⊀	02280	nprec
⪯	02AAF	npreceq
⇏	021CF	nrArr
↛	0219B	nrarr
↝	0219D	nrarrw
↛	0219B	nrightarrow
⋫	022EB	nrtri
⋭	022ED	nrtrie
⊁	02281	nsc
⋡	022E1	nsccue
⪰	02AB0	nsce
∤	02224	nshortmid
∦	02226	nshortparallel
≁	02241	nsim
≄	02244	nsime
≄	02244	nsimeq
∤	02224	nsmid
∦	02226	nspar
⋢	022E2	nsqsube
⋣	022E3	nsqsupe
⊄	02284	nsub
⫅	02AC5	nsubE
⊈	02288	nsube
⊂	02282	nsubset
⊈	02288	nsubseteq
⫅	02AC5	nsubseteqq
⊁	02281	nsucc
⪰	02AB0	nsucceq
⊅	02285	nsup
⫆	02AC6	nsupE
⊉	02289	nsupe
⊃	02283	nsupset
⊉	02289	nsupseteq
⫆	02AC6	nsupseteqq
≹	02279	ntgl
≸	02278	ntlg
⋪	022EA	ntriangleleft
⋬	022EC	ntrianglelefteq
⋫	022EB	ntriangleright
⋭	022ED	ntrianglerighteq
ν	003BD	nu
⊭	022AD	nvDash
≍	0224D	nvap
⊬	022AC	nvdash
≥	02265	nvge
>	0003E	nvgt
≤	02264	nvle
∼	0223C	nvsim
⇖	021D6	nwArr
	02923	nwarhk
↖	02196	nwarr
↖	02196	nwarrow
Ⓢ	024C8	oS
⊛	0229B	oast
⊚	0229A	ocir
⊝	0229D	odash
⊙	02299	odot
ο	003BF	ogr
ω	003C9	ohgr
Ω	003A9	ohm
∮	0222E	oint
↺	021BA	olarr
ω	003C9	omega
ο	003BF	omicron
⊖	02296	ominus
⊕	02295	oplus
∨	02228	or
↻	021BB	orarr
⊘	02298	osol
⊗	02297	otimes
∥	02225	par

¶	000B6	para
∥	02225	parallel
∂	02202	part
.	0002E	period
⊥	022A5	perp
π	003C0	pgr
φ	003C6	phgr
φ	003C6	phi
ϕ	003D5	phiv
π	003C0	pi
⋔	022D4	pitchfork
ϖ	003D6	piv
ℏ	0210F	planck
ℎ	0210E	planckh
ℏ	0210F	plankv
+	0002B	plus
⊞	0229E	plusb
∔	02214	plusdo
±	000B1	plusminus
±	000B1	plusmn
±	000B1	pm
≺	0227A	pr
⪳	02AB3	prE
⪷	02AB7	prap
≼	0227C	prcue
⪯	02AAF	pre
≺	0227A	prec
⪷	02AB7	precapprox
≼	0227C	preccurlyeq
⪯	02AAF	preceq
⪹	02AB9	precnapprox
⪵	02AB5	precneqq
⋨	022E8	precnsim
≾	0227E	precsim
′	02032	prime
ℙ	02119	primes
⪵	02AB5	prnE
⪹	02AB9	prnap
⋨	022E8	prnsim
∏	0220F	prod
∝	0221D	prop
∝	0221D	propto
≾	0227E	prsim
ψ	003C8	psgr
ψ	003C8	psi
?	0003F	quest
≟	0225F	questeq
"	00022	quot
⇛	021DB	rAarr
⇒	021D2	rArr
∽	0223D	race
√	0221A	radic
⟩	027E9	rang
⟩	027E9	rangle
→	02192	rarr
⇥	021E5	rarrb
↪	021AA	rarrhk
↬	021AC	rarrlp
↣	021A3	rarrtl
↝	0219D	rarrw
∶	02236	ratio
ℚ	0211A	rationals
	0290D	rbarr
}	0007D	rbrace
]	0005D	rbrack
⌉	02309	rceil
}	0007D	rcub
↳	021B3	rdsh
ℜ	0211C	real
ℜ	0211C	realpart
ℝ	0211D	reals
⌋	0230B	rfloor
ρ	003C1	rgr
⇁	021C1	rhard
⇀	021C0	rharu
ρ	003C1	rho

→	02192	rightarrow
↣	021A3	rightarrowtail
⇁	021C1	rightharpoondown
⇀	021C0	rightharpoonup
⇄	021C4	rightleftarrows
⇌	021CC	rightleftharpoons
⇉	021C9	rightrightarrows
↝	0219D	rightsquigarrow
⋌	022CC	rightthreetimes
˚	002DA	ring
≓	02253	risingdotseq
⇄	021C4	rlarr
⇌	021CC	rlhar
	023B1	rmoust
	023B1	rmoustache
⇾	021FE	roarr
⟧	027E7	robrk
)	00029	rpar
⇉	021C9	rrarr
↱	021B1	rsh
]	0005D	rsqb
⋌	022CC	rthree
⋊	022CA	rtimes
▸	025B8	rtrif
≻	0227B	sc
⪴	02AB4	scE
⪸	02AB8	scap
≽	0227D	sccue
⪰	02AB0	sce
⪶	02AB6	scnE
⪺	02ABA	scnap
⋩	022E9	scnsim
≿	0227F	scsim
⋅	022C5	sdot
⊡	022A1	sdotb
⇘	021D8	seArr
	02925	searhk
↘	02198	searr
↘	02198	searrow
§	000A7	sect
;	0003B	semi
∖	02216	setminus
∖	02216	setmn
ς	003C2	sfgr
⌢	02322	sfrown
σ	003C3	sgr
♯	0266F	sharp
∣	02223	shortmid
∥	02225	shortparallel
σ	003C3	sigma
ς	003C2	sigmaf
ς	003C2	sigmav
∼	0223C	sim
≃	02243	sime
≃	02243	simeq
≆	02246	simne
←	02190	slarr
∖	02216	smallsetminus
∣	02223	smid
⌣	02323	smile
/	0002F	sol
♠	02660	spades
♠	02660	spadesuit
∥	02225	spar
⊓	02293	sqcap
⊓	02293	sqcaps
⊔	02294	sqcup
⊔	02294	sqcups
⊏	0228F	sqsub
⊑	02291	sqsube
⊏	0228F	sqsubset
⊑	02291	sqsubseteq
⊐	02290	sqsup
⊒	02292	sqsupe
⊐	02290	sqsupset
⊒	02292	sqsupseteq

□	025A1	squ
□	025A1	square
→	02192	srarr
∖	02216	ssetmn
⌣	02323	ssmile
⋆	022C6	sstarf
	02605	starf
ϵ	003F5	straightepsilon
ϕ	003D5	straightphi
¯	000AF	strns
⊂	02282	sub
⫅	02AC5	subE
⊆	02286	sube
⫋	02ACB	subnE
⊊	0228A	subne
⊂	02282	subset
⊆	02286	subseteq
⫅	02AC5	subseteqq
⊊	0228A	subsetneq
⫋	02ACB	subsetneqq
≻	0227B	succ
⪸	02AB8	succapprox
≽	0227D	succcurlyeq
⪰	02AB0	succeq
⪺	02ABA	succnapprox
⪶	02AB6	succneqq
⋩	022E9	succnsim
≿	0227F	succsim
∑	02211	sum
⊃	02283	sup
⫆	02AC6	supE
⊇	02287	supe
⫌	02ACC	supnE
⊋	0228B	supne
⊃	02283	supset
⊇	02287	supseteq
⫆	02AC6	supseteqq
⊋	0228B	supsetneq
⫌	02ACC	supsetneqq
⇙	021D9	swArr
	02926	swarhk
↙	02199	swarr
↙	02199	swarrow
τ	003C4	tau
⃛	020DB	tdot
τ	003C4	tgr
∴	02234	there4
∴	02234	therefore
θ	003B8	theta
ϑ	003D1	thetasym
ϑ	003D1	thetav
θ	003B8	thgr
≈	02248	thickapprox
∼	0223C	thicksim
≈	02248	thkap
∼	0223C	thksim
˜	002DC	tilde
×	000D7	times
⊠	022A0	timesb
∭	0222D	tint
⊤	022A4	top
≜	0225C	triangleq
≜	0225C	trie
≬	0226C	twixt
↞	0219E	twoheadleftarrow
↠	021A0	twoheadrightarrow
⇑	021D1	uArr
↑	02191	uarr
⇅	021C5	udarr
υ	003C5	ugr
↿	021BF	uharl
↾	021BE	uharr
	0231C	ulcorn
	0231C	ulcorner
¨	000A8	uml
↑	02191	uparrow

↕	02195	updownarrow
↿	021BF	upharpoonleft
↾	021BE	upharpoonright
⊎	0228E	uplus
υ	003C5	upsi
υ	003C5	upsilon
⇈	021C8	upuparrows
	0231D	urcorn
	0231D	urcorner
⋰	022F0	utdot
⇈	021C8	uuarr
⇕	021D5	vArr
⊨	022A8	vDash
ϵ	003F5	varepsilon
ϰ	003F0	varkappa
∅	02205	varnothing
ϕ	003D5	varphi
ϖ	003D6	varpi
∝	0221D	varpropto
↕	02195	varr
ς	003C2	varsigma
⊊	0228A	varsubsetneq
⫋	02ACB	varsubsetneqq
⊋	0228B	varsupsetneq
⫌	02ACC	varsupsetneqq
ϑ	003D1	vartheta
⊲	022B2	vartriangleleft
⊳	022B3	vartriangleright
⊢	022A2	vdash
∨	02228	vee
⊻	022BB	veebar
≚	0225A	veeeq
⋮	022EE	vellip
|	0007C	verbar
|	0007C	vert
⊲	022B2	vltri
⊂	02282	vnsub
⊃	02283	vnsup
∝	0221D	vprop
⊳	022B3	vrtri
⫋	02ACB	vsubnE
⊊	0228A	vsubne
⫌	02ACC	vsupnE
⊋	0228B	vsupne
∧	02227	wedge
≙	02259	wedgeq
℘	02118	weierp
℘	02118	wp
≀	02240	wr
≀	02240	wreath
⋂	022C2	xcap
	025EF	xcirc
⋃	022C3	xcup
▽	025BD	xdtri
ξ	003BE	xgr
	027FA	xhArr
	027F7	xharr
ξ	003BE	xi
	027F8	xlArr
	027F5	xlarr
	027FC	xmap
⨁	02A01	xoplus
⨂	02A02	xotime
	027F9	xrArr
	027F6	xrarr
⨆	02A06	xsqcup
⨄	02A04	xuplus
△	025B3	xutri
⋁	022C1	xvee
⋀	022C0	xwedge
¥	000A5	yen
ζ	003B6	zeta
ζ	003B6	zgr
⇝	021DD	zigrarr

A different way to look at this is Unicode itself. Here's the list of characters that have a math related property in ConTEXt.

00021	!	close
00022	"	default
00027	'	default
00028	(	open
00029	)	close
0002A	*	binary
0002B	+	binary
0002C	,	ord punctuation
0002E	.	ord punctuation
0002F	/	binary
00030	0	number
00031	1	number
00032	2	number
00033	3	number
00034	4	number
00035	5	number
00036	6	number
00037	7	number
00038	8	number
00039	9	number
0003A	:	relation
0003B	;	punctuation
0003C	<	relation
0003D	=	relation
0003E	>	relation
0003F	?	close
00041	A	variable
00042	B	variable
00043	C	variable
00044	D	variable
00045	E	variable
00046	F	variable
00047	G	variable
00048	H	variable
00049	I	variable
0004A	J	variable
0004B	K	variable
0004C	L	variable
0004D	M	variable
0004E	N	variable
0004F	O	variable
00050	P	variable
00051	Q	variable
00052	R	variable
00053	S	variable
00054	T	variable
00055	U	variable
00056	V	variable
00057	W	variable
00058	X	variable
00059	Y	variable
0005A	Z	variable
0005B	[	open
0005C	\	nothing
0005D	]	close
0005E	^	accent
00060	`	accent
00061	a	variable
00062	b	variable
00063	c	variable
00064	d	variable
00065	e	variable
00066	f	variable
00067	g	variable
00068	h	variable
00069	i	variable
0006A	j	variable
0006B	k	variable
0006C	l	variable
0006D	m	variable

0006E	n	variable
0006F	o	variable
00070	p	variable
00071	q	variable
00072	r	variable
00073	s	variable
00074	t	variable
00075	u	variable
00076	v	variable
00077	w	variable
00078	x	variable
00079	y	variable
0007A	z	variable
0007B	{	open
0007C	|	close delimiter
		nothing open relation
0007D	}	close
000A5	¥	nothing
000A7	§	box
000A8	¨	accent
000AC	¬	ord
000AF	¯	accent
000B1	±	binary
000B4	´	accent
000B6	¶	box
000B7	·	binary
000D7	×	binary
000F0		ord
000F7	÷	binary
00127		ord
002C6	ˆ	accent
002C7	ˇ	accent
002D8	˘	accent
002D9	˙	accent
002DA	˚	accent
002DC	˜	accent
00338	/	relation
00391	Α	variable
00392	Β	variable
00393	Γ	variable
00394	Δ	variable
00395	Ε	variable
00396	Ζ	variable
00397	Η	variable
00398	Θ	variable
00399	Ι	variable
0039A	Κ	variable
0039B	Λ	variable
0039C	Μ	variable
0039D	Ν	variable
0039E	Ξ	variable
0039F	Ο	variable
003A0	Π	variable
003A1	Ρ	variable
003A3	Σ	variable
003A4	Τ	variable
003A5	Υ	variable
003A6	Φ	variable
003A7	Χ	variable
003A8	Ψ	variable
003A9	Ω	variable
003B1	α	variable
003B2	β	variable
003B3	γ	variable
003B4	δ	variable
003B5	ε	variable
003B6	ζ	variable
003B7	η	variable
003B8	θ	variable
003B9	ι	variable
003BA	κ	variable
003BB	λ	variable
003BC	μ	variable
003BD	ν	variable
003BE	ξ	variable
003BF	ο	variable

003C0	π	variable
003C1	ρ	variable
003C2	ς	variable
003C3	σ	variable
003C4	τ	variable
003C5	υ	variable
003C6	φ	variable
003C7	χ	variable
003C8	ψ	variable
003C9	ω	variable
003D1	ϑ	variable
003D5	ϕ	variable
003D6	ϖ	variable
003DC	Ϝ	variable
003F0	ϰ	ord
003F5	ϵ	variable
003F6		variable
02016	‖	close delimiter nothing open
02020	†	binary box
02021	‡	binary box
02022	•	binary
02026	…	inner
02032	′	nothing
02044	⁄	binary close
020D7	⃗	accent
020DB	⃛	accent
020DD		binary default
020DE		default
020DF		default
020E9		accent
02102	ℂ	variable
02107	ℇ	variable
0210E	ℎ	variable
0210F	ℏ	variable
02111	ℑ	default
02113	ℓ	default
02115	ℕ	variable
02118	℘	default
02119	ℙ	variable
0211A	ℚ	variable
0211C	ℜ	default
0211D	ℝ	variable
02124	ℤ	variable
02126		variable
02127	℧	variable
02129		variable
0212B		variable
02132	Ⅎ	ord
02135	ℵ	default
02136	ℶ	default
02137	ℷ	default
02138	ℸ	default
02141		ord
02142		ord
02143		ord
02144		ord
02145		nothing
02146		nothing
02147		nothing
02148		nothing
02149		nothing
0214A		ord
0214B		bin
02190	←	over relation under
02191	↑	relation
02192	→	over relation under
02193	↓	relation
02194	↔	relation
02195	↕	relation
02196	↖	relation
02197	↗	relation
02198	↘	relation
02199	↙	relation
0219A	↚	relation
0219B	↛	relation

0219C	↜	relation
0219D	↝	relation
0219E	↞	relation
0219F	↟	relation
021A0	↠	relation
021A1	↡	relation
021A2	↢	relation
021A3	↣	relation
021A4	↤	relation
021A5	↥	relation
021A6	↦	relation
021A7	↧	relation
021A8	↨	ord
021A9	↩	relation
021AA	↪	relation
021AB	↫	relation
021AC	↬	relation
021AD	↭	relation
021AE	↮	relation
021AF	↯	relation
021B0	↰	relation
021B1	↱	relation
021B2	↲	relation
021B3	↳	relation
021B4	↴	ord
021B5	↵	ord
021B6	↶	relation
021B7	↷	relation
021B8	↸	relation
021B9	↹	relation
021BA	↺	relation
021BB	↻	relation
021BC	↼	relation
021BD	↽	relation
021BE	↾	relation
021BF	↿	relation
021C0	⇀	relation
021C1	⇁	relation
021C2	⇂	relation
021C3	⇃	relation
021C4	⇄	relation
021C5	⇅	relation
021C6	⇆	relation
021C7	⇇	relation
021C8	⇈	relation
021C9	⇉	relation
021CA	⇊	relation
021CB	⇋	relation
021CC	⇌	relation
021CD	⇍	relation
021CE	⇎	relation
021CF	⇏	relation
021D0	⇐	relation
021D1	⇑	relation
021D2	⇒	relation
021D3	⇓	relation
021D4	⇔	relation
021D5	⇕	relation
021D6	⇖	relation
021D7	⇗	relation
021D8	⇘	relation
021D9	⇙	relation
021DA	⇚	relation
021DB	⇛	relation
021DC	⇜	relation
021DD	⇝	relation
021DE	⇞	relation
021DF	⇟	relation
021E0	⇠	relation
021E1	⇡	relation
021E2	⇢	relation
021E3	⇣	relation
021E4	⇤	relation
021E5	⇥	relation
021E6	⇦	ord
021E7	⇧	ord

021E8	⇨	ord
021E9	⇩	ord
021EB	⇫	ord
021F4	⇴	relation
021F5	⇵	relation
021F6	⇶	relation
021F7	⇷	relation
021F8	⇸	relation
021F9	⇹	relation
021FA	⇺	relation
021FB	⇻	relation
021FC	⇼	relation
021FD	⇽	relation
021FE	⇾	relation
021FF	⇿	relation
02200	∀	ord
02201	∁	ord
02202	∂	default
02203	∃	ord
02204	∄	ord
02205	∅	default
02208	∈	relation
02209	∉	relation
0220B	∋	relation
0220C	∌	relation
0220F	∏	limop
02210	∐	limop
02211	∑	limop
02212	−	binary
02213	∓	binary
02214	∔	binary
02216	∖	binary
02217	∗	binary
02218	∘	binary
02219	∙	binary
0221A	√	radical
0221D	∝	relation
0221E	∞	default
0221F	∟	ord
02220	∠	ord
02221	∡	ord
02222	∢	ord
02223	∣	binary
02224	∤	binary relation
02225	∥	relation
02226	∦	relation
02227	∧	binary
02228	∨	bin
02229	∩	binary
0222A	∪	binary
0222B	∫	limop nothing
0222C	∬	limop nothing
0222D	∭	limop nothing
0222E	∮	limop
0222F	∯	limop
02230	∰	limop
02231	∱	limop
02232	∲	limop
02233	∳	limop
02234	∴	relation
02235	∵	relation
02236	∶	punctuation
02237	∷	relation
02238	∸	binary
02239	∹	relation
0223C	∼	relation
0223D	∽	relation
02240	≀	binary
02241	≁	relation
02242	≂	relation
02243	≃	relation
02244	≄	relation
02245	≅	relation
02246	≆	relation
02247	≇	relation
02248	≈	relation

02249	≉	relation
0224A	≊	relation
0224C	≌	relation
0224D	≍	relation
0224E	≎	relation
02251	≑	relation
02252	≒	relation
02253	≓	relation
02254	≔	relation
02255	≕	relation
02256	≖	relation
02257	≗	relation
02259	≙	relation
0225A	≚	relation
0225B	≛	relation
0225C	≜	relation
0225D	≝	relation
0225E	≞	relation
0225F	≟	relation
02260	≠	relation
02261	≡	relation
02262	≢	relation
02264	≤	relation
02265	≥	relation
02266	≦	relation
02267	≧	relation
02268	≨	relation
02269	≩	relation
0226A	≪	relation
0226B	≫	relation
0226C	≬	relation
0226D	≭	relation
0226E	≮	relation
0226F	≯	relation
02270	≰	relation
02271	≱	relation
02272	≲	relation
02273	≳	relation
02274	≴	relation
02275	≵	relation
02276	≶	relation
02277	≷	relation
02278	≸	relation
02279	≹	relation
0227A	≺	relation
0227B	≻	relation
0227C	≼	relation
0227D	≽	relation
0227E	≾	relation
0227F	≿	relation
02280	⊀	relation
02281	⊁	relation
02282	⊂	relation
02283	⊃	relation
02284	⊄	relation
02285	⊅	relation
02286	⊆	relation
02287	⊇	relation
02288	⊈	relation
02289	⊉	relation
0228A	⊊	relation
0228B	⊋	relation
0228E	⊎	binary
0228F	⊏	relation
02290	⊐	relation
02291	⊑	binary
02292	⊒	binary
02293	⊓	binary
02294	⊔	binary
02295	⊕	binary
02296	⊖	binary
02297	⊗	binary
02298	⊘	binary
02299	⊙	binary
0229A	⊚	binary
0229B	⊛	binary

0229C	⊜	binary
0229D	⊝	binary
0229E	⊞	binary
0229F	⊟	binary
022A0	⊠	binary
022A1	⊡	binary
022A2	⊢	relation
022A3	⊣	relation
022A4	⊤	default
022A5	⊥	default relation
022A7	⊧	relation
022A8	⊨	relation
022A9	⊩	relation
022AA	⊪	relation
022AB	⊫	relation
022AC	⊬	relation
022AD	⊭	relation
022AE	⊮	relation
022AF	⊯	relation
022B2	⊲	bin
022B3	⊳	bin
022B8	⊸	relation
022BA	⊺	binary
022BB	⊻	binary
022BC	⊼	binary
022C0	⋀	limop
022C1	⋁	limop
022C2	⋂	limop
022C3	⋃	limop
022C4	⋄	binary
022C5	⋅	binary punctuation
022C6	⋆	binary
022C7	⋇	binary
022C8	⋈	relation
022C9	⋉	binary
022CA	⋊	binary
022CB	⋋	binary
022CC	⋌	binary
022CE	⋎	binary
022CF	⋏	binary
022D0	⋐	relation
022D1	⋑	relation
022D2	⋒	binary
022D3	⋓	binary
022D4	⋔	relation
022D6	⋖	binary
022D7	⋗	binary
022D8	⋘	relation
022D9	⋙	relation
022DA	⋚	relation
022DB	⋛	relation
022DC	⋜	relation
022DD	⋝	relation
022DE	⋞	relation
022DF	⋟	relation
022E0	⋠	relation
022E1	⋡	relation
022E2	⋢	relation
022E3	⋣	relation
022E4	⋤	relation
022E5	⋥	relation
022E6	⋦	relation
022E7	⋧	relation
022E8	⋨	relation
022E9	⋩	relation
022EA	⋪	relation
022EB	⋫	relation
022EC	⋬	relation
022ED	⋭	relation
022EE	⋮	inner
022EF	⋯	inner
022F0	⋰	inner
022F1	⋱	inner
02300		ord
02308	⌈	open
02309	⌉	close

0230A	⌊	open	027E8	⟨	open
0230B	⌋	close	027E9	⟩	close
0231C		open	027EA		open
0231D		close	027EB		close
0231E		open	027EE		open
0231F		close	027EF		close
02322	⌢	relation	027F5		relation
02323	⌣	relation	027F6		relation
023B0		open	027F7		relation
023B1		close	027F8		relation
023DC	⏜	topaccent	027F9		relation
023DD	⏝	botaccent	027FA		relation
023DE	⏞	topaccent	027FB		relation
023DF	⏟	botaccent	027FC		relation
023E0		accent	027FD		relation
023E1		accent	027FE		relation
024C7		ord	027FF		relation
024C8	Ⓢ	ord	02906		relation
025A1	□	ord	02907		relation
025A2		ord	0290A		relation
025B3	△	binary ord	0290B		relation
025B6	▶	bin	0290C		relation
025B8	▸	bin	0290D		relation
025BD	▽	binary	02911		relation
025CA	◊	ord	02916		relation
025EF		binary	02917		relation
02605		ord	02921		relation
02660	♠	default	02922		relation
02661		default	02923		relation
02662		default	02924		relation
02663	♣	default	02925		relation
0266D	♭	default	02926		relation
0266E	♮	default	029EB		ord
0266F	♯	default	02A01	⨁	limop
02713	✓	nothing	02A02	⨂	limop
02720	✠	nothing	02A03	⨃	limop
027E6	⟦	open	02A04	⨄	limop
027E7	⟧	close	02A05	⨅	limop

02A06	⨆	limop	02AB6	⪶	relation
02A09	⨉	limop	02AB7	⪷	relation
02A3F	⨿	binary	02AB8	⪸	relation
02A7D	⩽	relation	02AB9	⪹	relation
02A7E	⩾	relation	02ABA	⪺	relation
02A85	⪅	relation	02AC5	⫅	relation
02A86	⪆	relation	02AC6	⫆	relation
02A87	⪇	relation	02ACB	⫋	relation
02A88	⪈	relation	02ACC	⫌	relation
02A89	⪉	relation	12035		ord
02A8A	⪊	relation	1D6A4	𝚤	default
02A8B	⪋	relation	1D6A5	𝚥	default
02A8C	⪌	relation	1D6FB		default
02A95	⪕	relation	1D717	𝜗	default
02A96	⪖	relation	1D718	𝜘	default
02AAF	⪯	relation	1D71A	𝜚	variable
02AB0	⪰	relation	FE302		accent
02AB1	⪱	relation	FE303		accent
02AB2	⪲	relation	FE321		relation
02AB3	⪳	relation	FE322		relation
02AB4	⪴	relation	FE323		relation
02AB5	⪵	relation	FE324		relation

Traditionally (in TeX) one enters ascii characters to represent identifiers and use a font switch to get for instance a bold rendering. In Unicode it is more natural to use code points that represent the meaning. So, instead if enterinf

So instead of keying in byte U+0058 for a bold x one will use an utf sequence representing U+1D431. Because there are not than many editors that show all those Unicode characters it still makes sense to use regular latin and greek alphabets combined with directives that tell what real alphabet is used. For ConTeXt it does not matter what approach is chosen: both work ok and internally characters are mapped onto the right slot. When a font does not provide a shape a fallback is chosen. Technically one can construct a complete math font by combining all kind of fonts, but this is normally not needed.

Here we show the combinations of styles and alternatives. Not all combinations are present in Unicode. Actually, as Unicode math is rather agnostic of cultural determined math rendering, at some point ConTeXt could provide more.[1] Also, modern OpenType fonts

[1] An example is the German handwriting style Suetterlin that is still used for vectors.

can have alternatives, for instance variants of script, blackboard or fraktur. This is not related to Unicode and it makes no sense to encode that in MathML, but a setup of the rendering.

regular normal	ABCDEFGHIJKLMNOPQRSTUVWXYZ	00036 - 00039
	abcdefghijklmnopqrstuvwxyz	00039 - 00031
	ΑΒΓΔΕΖΗΘΙΚΛΜΝΞΟΠΡΣΤΥΦΧΨΩ	00039 - 00039
	αβγδεζηθικλμνξοπρςστυφχψωϑϕϖϰϱϵ	00039 - 00031
	0123456789	00034 - 00035
regular bold	**ABCDEFGHIJKLMNOPQRSTUVWXYZ**	1D400 - 1D419
	abcdefghijklmnopqrstuvwxyz	1D41A - 1D433
	ΑΒΓΔΕΖΗΘΙΚΛΜΝΞΟΠΡΣΤΥΦΧΨΩ	1D6A8 - 1D6C0
	αβγδεζηθικλμνξοπρςστυφχψωϑϕϖϰϱϵ	1D6C2 - 1D6DC
	0123456789	1D7CE - 1D7D7
regular italic	*ABCDEFGHIJKLMNOPQRSTUVWXYZ*	1D434 - 1D44D
	abcdefghijklmnopqrstuvwxyz	1D44E - 1D467
	ΑΒΓΔΕΖΗΘΙΚΛΜΝΞΟΠΡΣΤΥΦΧΨΩ	1D6E2 - 1D6FA
	αβγδεζηθικλμνξοπρςστυφχψωϑϕϖϰϱϵ	1D6FC - 1D716
	0123456789	00034 - 00035
regular bolditalic	***ABCDEFGHIJKLMNOPQRSTUVWXYZ***	1D468 - 1D481
	abcdefghijklmnopqrstuvwxyz	1D482 - 1D49B
	ΑΒΓΔΕΖΗΘΙΚΛΜΝΞΟΠΡΣΤΥΦΧΨΩ	1D71C - 1D734
	αβγδεζηθικλμνξοπρςστυφχψωϑϕϖϰϱϵ	1D736 - 1D750
	0123456789	1D7CE - 1D7D7
sansserif normal	ABCDEFGHIJKLMNOPQRSTUVWXYZ	1D5A0 - 1D5B9
	abcdefghijklmnopqrstuvwxyz	1D5BA - 1D5D3
	ΑΒΓΔΕΖΗΘΙΚΛΜΝΞΟΠΡΣΤΥΦΧΨΩ	00039 - 00039
	αβγδεζηθικλμνξοπρςστυφχψωϑϕϖϰϱϵ	00039 - 00031
	0123456789	00034 - 00035
sansserif bold	**ABCDEFGHIJKLMNOPQRSTUVWXYZ**	1D5D4 - 1D5ED
	abcdefghijklmnopqrstuvwxyz	1D5EE - 1D607
	ΑΒΓΔΕΖΗΘΙΚΛΜΝΞΟΠΡΣΤΥΦΧΨΩ	1D756 - 1D76E
	αβγδεζηθικλμνξοπρςστυφχψωϑϕϖϰϱϵ	1D770 - 1D78A
	0123456789	1D7EC - 1D7F5
sansserif italic	*ABCDEFGHIJKLMNOPQRSTUVWXYZ*	1D608 - 1D621
	abcdefghijklmnopqrstuvwxyz	1D622 - 1D63B
	ΑΒΓΔΕΖΗΘΙΚΛΜΝΞΟΠΡΣΤΥΦΧΨΩ	00039 - 00039
	αβγδεζηθικλμνξοπρςστυφχψωϑϕϖϰϱϵ	00039 - 00031

	0123456789	00034 - 00035
sansserif bolditalic	***ABCDEFGHIJKLMNOPQRSTUVWXYZ***	1D63C - 1D655
	abcdefghijklmnopqrstuvwxyz	1D656 - 1D66F
	ΑΒΓΔΕΖΗΘΙΚΛΜΝΞΟΠΡΣΤΥΦΧΨΩ	1D790 - 1D7A8
	αβγδεζηθικλμνξοπρςστυφχψωϑϕϖϰϱϵ	1D7AA - 1D7C4
	0123456789	1D7CE - 1D7D7
monospaced normal	ABCDEFGHIJKLMNOPQRSTUVWXYZ	1D670 - 1D689
	abcdefghijklmnopqrstuvwxyz	1D68A - 1D6A3
	ΑΒΓΔΕΖΗΘΙΚΛΜΝΞΟΠΡΣΤΥΦΧΨΩ	00039 - 00039
	αβγδεζηθικλμνξοπρςστυφχψωϑϕϖϰϱϵ	00039 - 00031
	0123456789	1D7F6 - 1D7FF
monospaced bold	ABCDEFGHIJKLMNOPQRSTUVWXYZ	1D5A0 - 1D5B9
	abcdefghijklmnopqrstuvwxyz	1D5BA - 1D5D3
	ΑΒΓΔΕΖΗΘΙΚΛΜΝΞΟΠΡΣΤΥΦΧΨΩ	00039 - 00039
	αβγδεζηθικλμνξοπρςστυφχψωϑϕϖϰϱϵ	00039 - 00031
	0123456789	00034 - 00035
monospaced italic	ABCDEFGHIJKLMNOPQRSTUVWXYZ	1D5A0 - 1D5B9
	abcdefghijklmnopqrstuvwxyz	1D5BA - 1D5D3
	ΑΒΓΔΕΖΗΘΙΚΛΜΝΞΟΠΡΣΤΥΦΧΨΩ	00039 - 00039
	αβγδεζηθικλμνξοπρςστυφχψωϑϕϖϰϱϵ	00039 - 00031
	0123456789	00034 - 00035
monospaced bolditalic	**ABCDEFGHIJKLMNOPQRSTUVWXYZ**	1D5D4 - 1D5ED
	abcdefghijklmnopqrstuvwxyz	1D5EE - 1D607
	ΑΒΓΔΕΖΗΘΙΚΛΜΝΞΟΠΡΣΤΥΦΧΨΩ	1D756 - 1D76E
	αβγδεζηθικλμνξοπρςστυφχψωϑϕϖϰϱϵ	1D770 - 1D78A
	0123456789	1D7EC - 1D7F5
fraktur normal	𝔄𝔅ℭ𝔇𝔈𝔉𝔊ℌℑ𝔍𝔎𝔏𝔐𝔑𝔒𝔓𝔔ℜ𝔖𝔗𝔘𝔙𝔚𝔛𝔜ℨ	1D504 - 02128
	𝔞𝔟𝔠𝔡𝔢𝔣𝔤𝔥𝔦𝔧𝔨𝔩𝔪𝔫𝔬𝔭𝔮𝔯𝔰𝔱𝔲𝔳𝔴𝔵𝔶𝔷	1D51E - 1D537
	ΑΒΓΔΕΖΗΘΙΚΛΜΝΞΟΠΡΣΤΥΦΧΨΩ	00039 - 00039
	αβγδεζηθικλμνξοπρςστυφχψωϑϕϖϰϱϵ	00039 - 00031
	0123456789	00034 - 00035
fraktur bold	𝔄𝔅ℭ𝔇𝔈𝔉𝔊ℌℑ𝔍𝔎𝔏𝔐𝔑𝔒𝔓𝔔ℜ𝔖𝔗𝔘𝔙𝔚𝔛𝔜ℨ	1D504 - 02128
	𝔞𝔟𝔠𝔡𝔢𝔣𝔤𝔥𝔦𝔧𝔨𝔩𝔪𝔫𝔬𝔭𝔮𝔯𝔰𝔱𝔲𝔳𝔴𝔵𝔶𝔷	1D51E - 1D537
	ΑΒΓΔΕΖΗΘΙΚΛΜΝΞΟΠΡΣΤΥΦΧΨΩ	00039 - 00039
	αβγδεζηθικλμνξοπρςστυφχψωϑϕϖϰϱϵ	00039 - 00031
	0123456789	00034 - 00035
fraktur italic	𝔄𝔅ℭ𝔇𝔈𝔉𝔊ℌℑ𝔍𝔎𝔏𝔐𝔑𝔒𝔓𝔔ℜ𝔖𝔗𝔘𝔙𝔚𝔛𝔜ℨ	1D504 - 02128
	𝔞𝔟𝔠𝔡𝔢𝔣𝔤𝔥𝔦𝔧𝔨𝔩𝔪𝔫𝔬𝔭𝔮𝔯𝔰𝔱𝔲𝔳𝔴𝔵𝔶𝔷	1D51E - 1D537

	ΑΒΓΔΕΖΗΘΙΚΛΜΝΞΟΠΡΣΤΥΦΧΨΩ	00039 - 00039
	αβγδεζηθικλμνξοπρςστυφχψωϑϕϖϰϱϵ	00039 - 00031
	0123456789	00034 - 00035
fraktur bolditalic	𝔄𝔅ℭ𝔇𝔈𝔉𝔊ℌℑ𝔍𝔎𝔏𝔐𝔑𝔒𝔓𝔔ℜ𝔖𝔗𝔘𝔙𝔚𝔛𝔜ℨ	1D504 - 02128
	𝔞𝔟𝔠𝔡𝔢𝔣𝔤𝔥𝔦𝔧𝔨𝔩𝔪𝔫𝔬𝔭𝔮𝔯𝔰𝔱𝔲𝔳𝔴𝔵𝔶𝔷	1D51E - 1D537
	ΑΒΓΔΕΖΗΘΙΚΛΜΝΞΟΠΡΣΤΥΦΧΨΩ	00039 - 00039
	αβγδεζηθικλμνξοπρςστυφχψωϑϕϖϰϱϵ	00039 - 00031
	0123456789	00034 - 00035
script normal	𝒜ℬ𝒞𝒟ℰℱ𝒢ℋℐ𝒥𝒦ℒℳ𝒩𝒪𝒫𝒬ℛ𝒮𝒯𝒰𝒱𝒲𝒳𝒴𝒵	1D49C - 1D4B5
	𝒶𝒷𝒸𝒹ℯ𝒻ℊ𝒽𝒾𝒿𝓀𝓁𝓂𝓃ℴ𝓅𝓆𝓇𝓈𝓉𝓊𝓋𝓌𝓍𝓎𝓏	1D4B6 - 1D4CF
	ΑΒΓΔΕΖΗΘΙΚΛΜΝΞΟΠΡΣΤΥΦΧΨΩ	00039 - 00039
	αβγδεζηθικλμνξοπρςστυφχψωϑϕϖϰϱϵ	00039 - 00031
	0123456789	00034 - 00035
script bold	𝒜ℬ𝒞𝒟ℰℱ𝒢ℋℐ𝒥𝒦ℒℳ𝒩𝒪𝒫𝒬ℛ𝒮𝒯𝒰𝒱𝒲𝒳𝒴𝒵	1D49C - 1D4B5
	𝒶𝒷𝒸𝒹ℯ𝒻ℊ𝒽𝒾𝒿𝓀𝓁𝓂𝓃ℴ𝓅𝓆𝓇𝓈𝓉𝓊𝓋𝓌𝓍𝓎𝓏	1D4B6 - 1D4CF
	ΑΒΓΔΕΖΗΘΙΚΛΜΝΞΟΠΡΣΤΥΦΧΨΩ	00039 - 00039
	αβγδεζηθικλμνξοπρςστυφχψωϑϕϖϰϱϵ	00039 - 00031
	0123456789	00034 - 00035
script italic	𝒜ℬ𝒞𝒟ℰℱ𝒢ℋℐ𝒥𝒦ℒℳ𝒩𝒪𝒫𝒬ℛ𝒮𝒯𝒰𝒱𝒲𝒳𝒴𝒵	1D49C - 1D4B5
	𝒶𝒷𝒸𝒹ℯ𝒻ℊ𝒽𝒾𝒿𝓀𝓁𝓂𝓃ℴ𝓅𝓆𝓇𝓈𝓉𝓊𝓋𝓌𝓍𝓎𝓏	1D4B6 - 1D4CF
	ΑΒΓΔΕΖΗΘΙΚΛΜΝΞΟΠΡΣΤΥΦΧΨΩ	00039 - 00039
	αβγδεζηθικλμνξοπρςστυφχψωϑϕϖϰϱϵ	00039 - 00031
	0123456789	00034 - 00035
script bolditalic	𝒜ℬ𝒞𝒟ℰℱ𝒢ℋℐ𝒥𝒦ℒℳ𝒩𝒪𝒫𝒬ℛ𝒮𝒯𝒰𝒱𝒲𝒳𝒴𝒵	1D49C - 1D4B5
	𝒶𝒷𝒸𝒹ℯ𝒻ℊ𝒽𝒾𝒿𝓀𝓁𝓂𝓃ℴ𝓅𝓆𝓇𝓈𝓉𝓊𝓋𝓌𝓍𝓎𝓏	1D4B6 - 1D4CF
	ΑΒΓΔΕΖΗΘΙΚΛΜΝΞΟΠΡΣΤΥΦΧΨΩ	00039 - 00039
	αβγδεζηθικλμνξοπρςστυφχψωϑϕϖϰϱϵ	00039 - 00031
	0123456789	00034 - 00035
blackboard normal	𝔸𝔹ℂ𝔻𝔼𝔽𝔾ℍ𝕀𝕁𝕂𝕃𝕄ℕ𝕆ℙℚℝ𝕊𝕋𝕌𝕍𝕎𝕏𝕐ℤ	1D538 - 02124
	abcdefghij𝕜lmnopqrstuvwxyz	1D552 - 1D56B
	ΑΒΓΔΕΖΗΘΙΚΛΜΝΞΟΠΡΣΤΥΦΧΨΩ	00039 - 00039
	αβπδεζηθικλμνξογρςστυφχψωϑϕϖϰϱϵ	00039 - 00031
	0123456789	1D7D8 - 1D7E1
blackboard bold	𝔸𝔹ℂ𝔻𝔼𝔽𝔾ℍ𝕀𝕁𝕂𝕃𝕄ℕ𝕆ℙℚℝ𝕊𝕋𝕌𝕍𝕎𝕏𝕐ℤ	1D538 - 02124
	abcdefghij𝕜lmnopqrstuvwxyz	1D552 - 1D56B
	ΑΒΓΔΕΖΗΘΙΚΛΜΝΞΟΠΡΣΤΥΦΧΨΩ	00039 - 00039
	αβπδεζηθικλμνξογρςστυφχψωϑϕϖϰϱϵ	00039 - 00031
	0123456789	1D7D8 - 1D7E1

blackboard italic	𝔸𝔹ℂ𝔻𝔼𝔽𝔾ℍ𝕀𝕁𝕂𝕃𝕄ℕ𝕆ℙℚℝ𝕊𝕋𝕌𝕍𝕎𝕏𝕐ℤ	1D538 - 02124
	abcdefghij𝕜lmnopqrstuvwxyz	1D552 - 1D56B
	ΑΒΓΔΕΖΗΘΙΚΛΜΝΞΟΠΡΣΤΥΦΧΨΩ	00039 - 00039
	αβπδεζηθικλμνξογρςστυφχψωϑϕϖϰϱϵ	00039 - 00031
	0123456789	1D7D8 - 1D7E1
blackboard bolditalic	𝔸𝔹ℂ𝔻𝔼𝔽𝔾ℍ𝕀𝕁𝕂𝕃𝕄ℕ𝕆ℙℚℝ𝕊𝕋𝕌𝕍𝕎𝕏𝕐ℤ	1D538 - 02124
	abcdefghij𝕜lmnopqrstuvwxyz	1D552 - 1D56B
	ΑΒΓΔΕΖΗΘΙΚΛΜΝΞΟΠΡΣΤΥΦΧΨΩ	00039 - 00039
	αβπδεζηθικλμνξογρςστυφχψωϑϕϖϰϱϵ	00039 - 00031
	0123456789	1D7D8 - 1D7E1