Content & Form in Mathematics
Presenting and Capturing
Mathematics for the Web in *MathML*

Michael Kohlhase

Professur für Wissensrepräsentation und -verarbeitung
Informatik, FAU Erlangen-Nürnberg
http://kwarc.info

March 14, 2018

Math on the Web $\widehat{=}$ *MathML*

- *MathML* extends HTML with mathematical formulae
- *MathML* is the first XML application by the W3C (1999)

Math on the Web $\widehat{=}$ *MathML*

- *MathML* extends HTML with mathematical formulae
- *MathML* is the first XML application by the W3C (1999)
- *MathML* has two sub-languages
  - Presentation MathML (layout) and
  - Content MathML (structure/semantics)

Math on the Web $\widehat{=}$ *MathML*

- *MathML* extends HTML with mathematical formulae
- *MathML* is the first XML application by the W3C (1999)
- *MathML* has two sub-languages
  - Presentation MathML (layout) and
  - Content MathML (structure/semantics)
- MathML3 reconciles *OpenMath* and Content MathML. (2010)
- *MathML* is part of HTML5 (well-integrated; current standard)

Math on the Web $\widehat{=}$ *MathML*

- *MathML* extends HTML with mathematical formulae
- *MathML* is the first XML application by the W3C (1999)
- *MathML* has two sub-languages
  - Presentation MathML (layout) and
  - Content MathML (structure/semantics)
- MathML3 reconciles *OpenMath* and Content MathML. (2010)
- *MathML* is part of HTML5 (well-integrated; current standard)
- two major browsers support *MathML* natively: `FireFox` and `Safari` ($\sim 15\%$)
- all browsers support *MathML* via MathJax (JavsScript+CSS+Fonts)

Math on the Web $\widehat{=}$ *MathML*

- *MathML* extends HTML with mathematical formulae
- *MathML* is the first XML application by the W3C (1999)
- *MathML* has two sub-languages
  - Presentation MathML (layout) and
  - Content MathML (structure/semantics)
- MathML3 reconciles *OpenMath* and Content MathML. (2010)
- *MathML* is part of HTML5 (well-integrated; current standard)
- two major browsers support *MathML* natively: `FireFox` and `Safari` ($\sim 15\%$)
- all browsers support *MathML* via MathJax (JavsScript+CSS+Fonts)
- *MathML* support in `Chrome` under way ($\sim 78\%$)
  see `https://mathml.igalia.com/`

Math on the Web $\widehat{=}$ *MathML*

- *MathML* extends HTML with mathematical formulae
- *MathML* is the first XML application by the W3C (1999)
- *MathML* has two sub-languages
  - Presentation MathML (layout) and
  - Content MathML (structure/semantics)
- MathML3 reconciles *OpenMath* and Content MathML. (2010)
- *MathML* is part of HTML5 (well-integrated; current standard)
- two major browsers support *MathML* natively: `FireFox` and `Safari` ($\sim 15\%$)
- all browsers support *MathML* via MathJax (JavsScript+CSS+Fonts)
- *MathML* support in `Chrome` under way ($\sim 78\%$)
  see `https://mathml.igalia.com/`
- Join the MathML Association (`http://mathml-association.org/`)

MathML: Mathematical Markup Language

> *MathML* is an XML application for describing mathematical notation and capturing both its structure and content. The goal of *MathML* is to enable mathematics to be served, received, and processed on the World Wide Web, just as HTML has enabled this functionality for text.
>
> *from the MathML2 Recommendation*

Representation of Formulae as Expression Trees

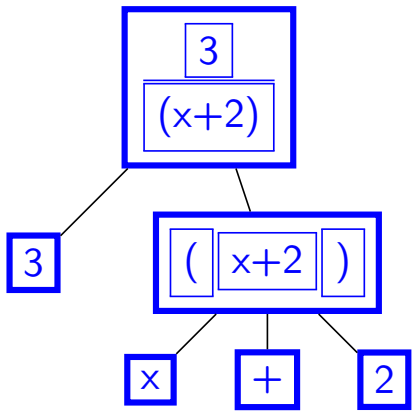
- Mathematical Expressions are build up as expression trees
  - of layout schemata in Presentation-*MathML*
  - of functional subexpressions in Content-*MathML*
- Example: $\frac{3}{x+2}$

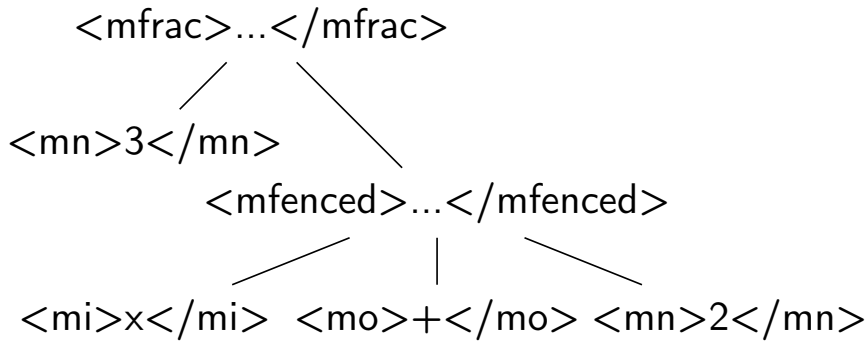
```
<mfrac>
  <mn>3</mn>
  <mfenced>
     <mi>x</mi>
     <mo>+</mo>
     <mn>2</mn>
   </mfenced>
</mfrac>
```

```
<apply>
  <divide/>
  <cn>3</cn>
  <apply>
    <plus/>
    <ci>x</ci>
    <cn>2</cn>
  </apply>
</apply>
```

Layout Schemata and the *MathML* Box model

- Presentation MathML represents the visual appearance of a formula in a tree of layout primitives
- **Example 0.1 (Presentation MathML for $3/(x+2)$).**





P-*MathML* Token Elements

- Tokens Elements directly contain character data (the only way to include it)
  Attributes: fontweight, fontfamily and fontstyle, color...
- Identifiers: <**mi**>... </**mi**> (∼ variables, italicized)
- Numbers: <**mn**>... </**mn**> (numbers)
- Operators: <**mo**>... </**mo**> (constants, functions, upright)
- Operator display is often ideosyncratic (Operator Dictionaries for defaults)
  - Examples: spacing, *-scripts in sums and limits, stretchy integrals,...
  - Attributes: lspace, rspace, stretchy, and movablelimits.
  - Operators include delimiter characters like
    - parentheses (which stretch),
    - punctuation (which has uneven spacing around it) and
    - accents (which also stretch).

MathML Symbols in UniCode

- Problem: Mathematical formula use lots of non-ASCII symbols (not on your keyboard)
- Math Symbols: α, β, ... Θ, $\int$, $\uplus$, $\pm$, ∞, $\mathbb{N}$, $\mathbb{R}$, ... (+ ca. 5000 more)
- Recap: The UniCode standard collects all characters of all languages in the world. (100 000 so far)
- Idea: Math is a language, use UniCode for its characters.
- Recap: Each UniCode character is identified by an unambiguous name and an integer number called its code point (a number $\leq 1\,100\,000$)
- **Example 0.2 (Some Math Symbols).**
  - The integral symbol $\int$ has the number U+8747 and the name `INTEGRAL`
  - The universal quantifier $\forall$ has the number U+8704 and the name `FOR ALL`
  - The letter θ has number U+952 and the name `GREEK SMALL LETTER THETA`

For *MathML*: UniCode letters can be used in HTML directly (and in *MathML*). Encode them via their code point as θ (decimal) or θ (hex).

General Layout Schemata

- horizontal row: <**mrow**>child1 ... </**mrow**> (alignment and grouping)
- fraction: <**mfrac**>numerator denominator </**mfrac**>
  Attribute: linethickness (set to 0 for binomial coefficients)
- Radicals: <**msqrt**>child1 ... </**msqrt**> and

  <**mroot**>base index</**mroot**>
- grouping with parenthesis: <**mfenced**>child ... </**mfenced**>
  Attributes: open="(" and close="]" to specify parentheses
- grouping and style: <**mstyle**>child ... </**mstyle**> (pre-set attributes)

First Practical Markup Challenge (aka. Practice Example)

- We will jointly practice with concrete examples, here $x^2 + 4x + 4 = 0$
- General Workflow: write, test, repeat until done.
  - bring out your favorite text editor. (it really does not matter which one)
  - prepare a HTML5 file test.html

```
<html>
  <body>
    testing a polynomial:
    <math displaystyle="true"> ...</math>
  </body>
</html>
```

  - have a look at it in FireFox
  - replace the <math> element by your markup for $x^2 + 4x + 4 = 0$
  - have a look at it in FireFox again (does it look right)

Example: $x^2 + 4x + 4 = 0$

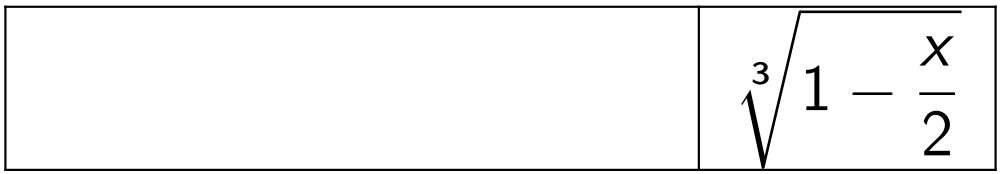
just presentation	some structure
<mrow> <msup> <mi>x</mi> <mn>2</mn> </msup> <mo>+</mo> <mn>4</mn> <mi>x</mi> <mo>+</mo> <mn>4</mn> <mo>=</mo> <mn>0</mn> </mrow>	<mrow> <mrow> <msup> <mi>x</mi> <mn>2</mn> </msup> <mo>+</mo> <mrow> <mn>4</mn> <mi>x</mi> </mrow> <mo>+</mo> <mn>4</mn> </mrow> <mo>=</mo> <mn>0</mn> </mrow>

Example: Grouping Arguments by `mfenced`

$f(x+y)$	$f(x+y)$

`<mrow>`
`  <mi>f</mi>`
`  <mfenced>`
`  <mrow>`
`  <mi>x</mi>`
`  <mo>+</mo>`
`  <mi>y</mi>`
`  </mrow>`
`  </mfenced>`
`</mrow>`
	`<mrow>`
	`  <mi>f</mi>`
	`  <mfenced>`
	`  <mstyle color='#ff0000'>`
	`  <mrow>`
	`  <mi>x</mi>`
	`  <mo>+</mo>`
	`  <mi>y</mi>`
	`  </mrow>`
	`  </mstyle>`
	`  </mfenced>`
	`</mrow>`

Example: <mfrac> and <mroot>



Example: <mfrac> and <mroot>

```
<mroot>
  <mrow>
    <mn>1</mn>
    <mo>−</mo>
    <mfrac>
      <mi>x</mi>
      <mn>2</mn>
    </mfrac>
  </mrow>
  <mn>3</mn>
</mroot>
```

$$\sqrt[3]{1-\frac{x}{2}}$$

Example: The quadratic formula $x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$

Example: The quadratic formula $x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$

```
<mrow>
  <mi>x</mi>
  <mo>=</mo>
  <mfrac>
    <mrow>
      <mrow><mo>−</mo><mi>b</mi></mrow>
      <mo>&plusmn;</mo>
      <msqrt>
        <mrow>
          <msup><mi>b</mi><mn>2</mn></msup>
          <mo>−</mo>
          <mrow><mn>4</mn><mi>a</mi><mi>c</mi></mrow>
        </mrow>
      </msqrt>
    </mrow>
    <mrow><mn>2</mn><mo>&InvisibleTimes;</mo><mi>a</mi></mrow>
  </mfrac>
</mrow>
```

Script Schemata

- Indices: G^1, H_5, R^i_j...
  - Super: <**msup**>base script </**msup**>
  - Subs: <**msub**>base script </**msub**>
  - Both: <**msubsup**>base superscript subscript</**msub**> (vertical alignment!)
- Bars and Arrows: $\overline{X}$, $\underbrace{Y}$,lue $\overline{Z}$,...
  - Under: <**munder**>base script</**munder**>
  - Over: <**mover**>base script</**mover**>
  - Both: <**munderover**>base underscript overscript </**munderover**>
- Tensor-like: use <none/> for missing scripts

```
<mmultiscripts>
   base (sub sup)* [<mprescripts/> (psub psup)*]
</mmultiscripts>
```

msub + msup vs. msubsup

msub + msup	msubsup
<msup> <msub> <mi>x</mi> <mn>1</mn> </msub> <mi>α</mi> </msup>	<msubsup> <mi>x</mi> <mn>1</mn> <mi>α</mi> </msubsup>
x_1^{α}	x_1^{α}

Example: Movable Limits on Sums

- **Example 0.3.** $\sum_{i=1}^{\infty} x^i + \sum_{i=1}^{\infty} x^i$

```
<mrow>
  <mstyle displaystyle='true'>
    <munderover>
      <mo>&sum;</mo>
      <mrow><mi>i</mi><mo>=</mo><mn>1</mn></mrow>
      <mi>&infty;</mi>
    </munderover>
    <msup><mi>x</mi><mi>i</mi></msup>
  </mstyle>
  <mo>+</mo>
  <mstyle displaystyle='false'>
    <munderover>
      <mo>&sum;</mo>
      <mrow><mi>i</mi><mo>=</mo><mn>1</mn></mrow>
      <mi>&infty;</mi>
    </munderover>
    <msup><mi>x</mi><mi>i</mi></msup>
  </mstyle>
</mrow>
```

Content Mathml: Expression Trees in Prefix Notation I

- Prefix Notation saves parentheses (so does postfix, BTW)

$(x-y)/2$	$x-(y/2)$
<apply> <divide/> <apply> <minus/> <ci>x</ci> <ci>y</ci> </apply> <cn>2</cn> </apply>	<apply> <minus/> <ci>x</ci> <apply> <divide/> <ci>y</ci> <cn>2</cn> </apply> </apply>

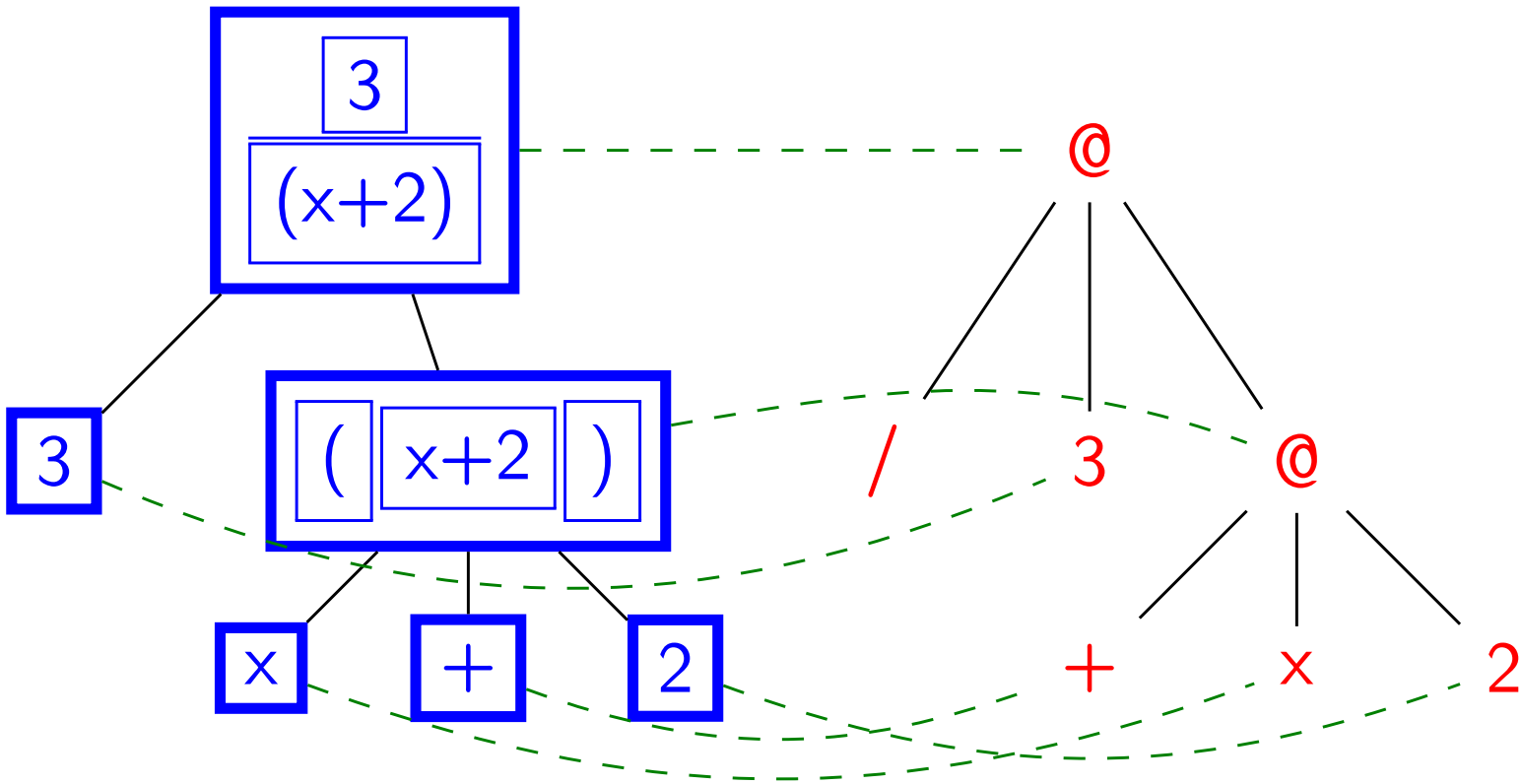
Function Application: <**apply**>function arg1 ... argn </**apply**>

Content Mathml: Expression Trees in Prefix Notation II

- Operators and Functions: $\sim$ 100 empty elements <**sin**/>, <**plus**/>, <**eq**/>, <**compose**/>,...
- Token elements: ci, cn (identifiers and numbers)
- Extra Operators: <**csymbol** cd="...">...</**csymbol**>

Parallel Markup e.g. in *MathML* I

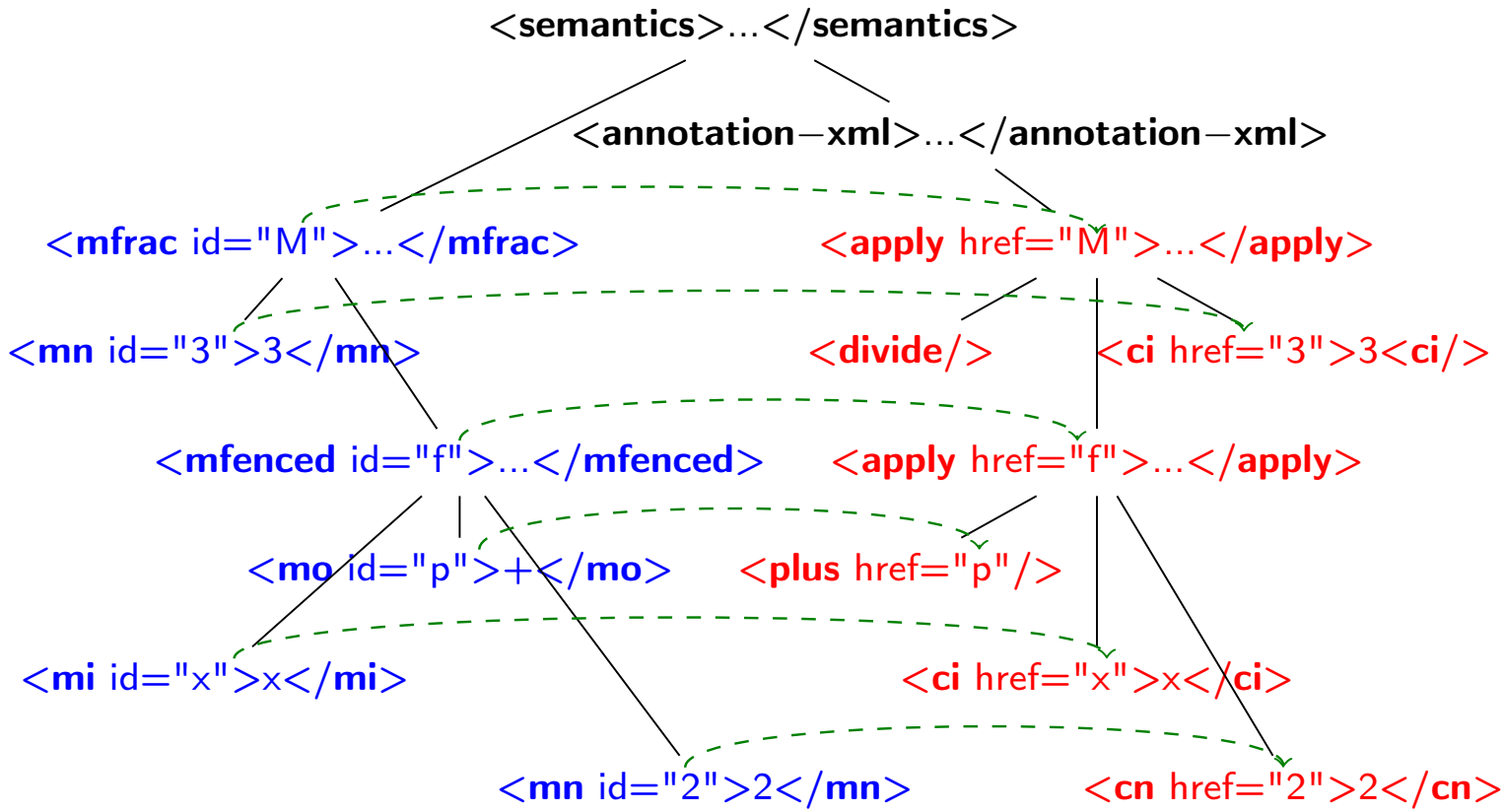
- Idea: Combine the presentation and content markup and cross-reference



- use e.g. for semantic copy and paste. (click o3n presentation, follow link and copy content)

Parallel Markup e.g. in *MathML* II

- Concrete Realization in *MathML*: semantics element with presentation as first child and content in annotation−xml child



Content Mathml: Expression Trees in Prefix Notation I

- Prefix Notation saves parentheses (so does postfix, BTW)

$(x-y)/2$	$x-(y/2)$
<apply> <divide/> <apply> <minus/> <ci>x</ci> <ci>y</ci> </apply> <cn>2</cn> </apply>	<apply> <minus/> <ci>x</ci> <apply> <divide/> <ci>y</ci> <cn>2</cn> </apply> </apply>

Function Application: <**apply**>function arg1 ... argn </**apply**>

Content Mathml: Expression Trees in Prefix Notation II

- Operators and Functions: $\sim$ 100 empty elements <**sin**/>, <**plus**/>, <**eq**/>, <**compose**/>,...
- Token elements: ci, cn (identifiers and numbers)
- Extra Operators: <**csymbol** cd="...">...</**csymbol**>

Examples of Content Math

Expression	Markup
<apply> <plus/> <apply><sin/><ci>x</ci></apply> <cn>9</cn> </apply>	$\sin(x) + 9$

Examples of Content Math

<apply><eq/><ci>x</ci><cn>1</cn></apply>	$x = 1$

Examples of Content Math

```
<apply><eq/>
  <bind><int/>
      <bvar><ci>x</ci></bvar>
      <apply><sin/><ci>x</ci></apply>
  </bind>
  <cos/>
</apply>
```

$$\int \sin(x)dx = \cos$$

Examples of Content Math

```
<bind>
   <apply>
      <csymbol cd="calculus1">defint</csymbol>
       <cn>0</cn>
       <csymbol cd="nums1">infinity</csymbol>
    </apply>
   <bvar><ci>x</ci></bvar>
   <apply><sin/><ci>x</ci></apply>
</bind>
```

$$\int_0^\infty \sin(x)dx$$

Examples of Content Math

```
<bind><sum/>
  <bvar><ci>n</ci></bvar>
  <lowlimit><cn>0</cn></lowlimit>
  <uplimit><ci>&infty;</ci></uplimit>
  <apply><power/><ci>x</ci><ci>n</ci></apply>
</bind>
```

$$\sum_0^\infty x^n$$

Examples of Content Math

```
<bind>
   <set/>
   <bvar><ci>x</ci></bvar>
   <bvar><ci>y</ci></bvar>
   <apply><and/>
     <apply><lt/>
        <ci>0</ci><ci>x</ci><ci>1</ci>
     </apply>
    <apply><leq/>
       <ci>3</ci><ci>y</ci><ci>10</ci>
    </apply>
</bind>
```

$$\left\{ x, y \;\middle|\; \begin{array}{l} 0 < x < 1, \\ 3 \leq y \leq 10 \end{array} \right\}$$

Examples of Content Math

Expression	Markup
<apply><eq/> <bind><set/> <bvar><ci>x</ci></bvar> <apply><geq/> <ci>x</ci><cn>0</cn> </apply> </bind> <interval closure='closed−open'> <cn>0</cn> <cn>&infty;</cn> </interval> </apply>	$\{x \mid x \geq 0\} = [0, \infty)$

Examples of Content Math

```
<apply><eq/>
  <apply><times/>
    <apply><vector/>
        <cn>1</cn><cn>2</cn>
    </apply>
    <apply><matrix/>
      <apply><matrixrow/>
          <cn>0</cn><cn>1</cn>
      </apply>
      <apply><matrixrow/>
          <cn>1</cn><cn>0</cn>
       </apply>
    </matrix>
  </apply>
  <apply>
    <transpose/>
    <apply><vector/>
        <cn>2</cn><cn>1</cn>
    </apply>
  </apply>
</apply>
```

$$(1,2) \times \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} = (2,1)^t$$

From Presentation to Content?

- Problem: Presentation Markup $\leftrightarrow$ Content Markup
  - many presentation for one concept (e.g. binomial coeff. $\binom{n}{k}$ vs. C_k^n vs. C_n^k)
  - many concepts for one presentation (e.g. m^3 is m cubed, cubic meter, upper index, footnote,...)
  - grouping is left implicit, invisible operators (e.g. $3a^2 + 6ab + b^2$)
  - disambiguation by context (e.g. $\lambda X_\alpha.X =_\alpha \lambda Y_\alpha.Y$)
  - notation is introduced and used on the fly.
- Content Recovery is a heuristic context/author-dependent process
  - There is little hope we can do it fully automatically in principle (AI-hard!)
  - for limited domains we can do a good job (e.g. in Mathematica 4)

Added-value services with Math Content

- cut and paste (cut output from web search engine and paste into CAS)
- automatically proof checking formal argumentations (bridge verification?)
- math explanation (e.g. specialize a proof to a simpler special case)
- semantical search for mathematical concepts (rather than keywords)
- data mining for representation theorems (find unnoticed groups out there)
- classification (given a concrete math structure, is there a general theory?)
- personalized notation (implication as $\rightarrow$ vs. $\supset$, or Ricci as $\frac{1}{2}\mathcal{R}^{ij}$ vs. $2\mathcal{R}^{ij}$)
- user-adapted documents (ActiveMath, Course Capsules)

References I