MathML

Presenting and Capturing Mathematics for the Web

http://www.cs.cmu.edu/~kohlhase/talks/mathml-tutorial

⚠ Mathematics $\widehat{=}$ Anything Formal ⚠

Michael Kohlhase

School of Computer Science

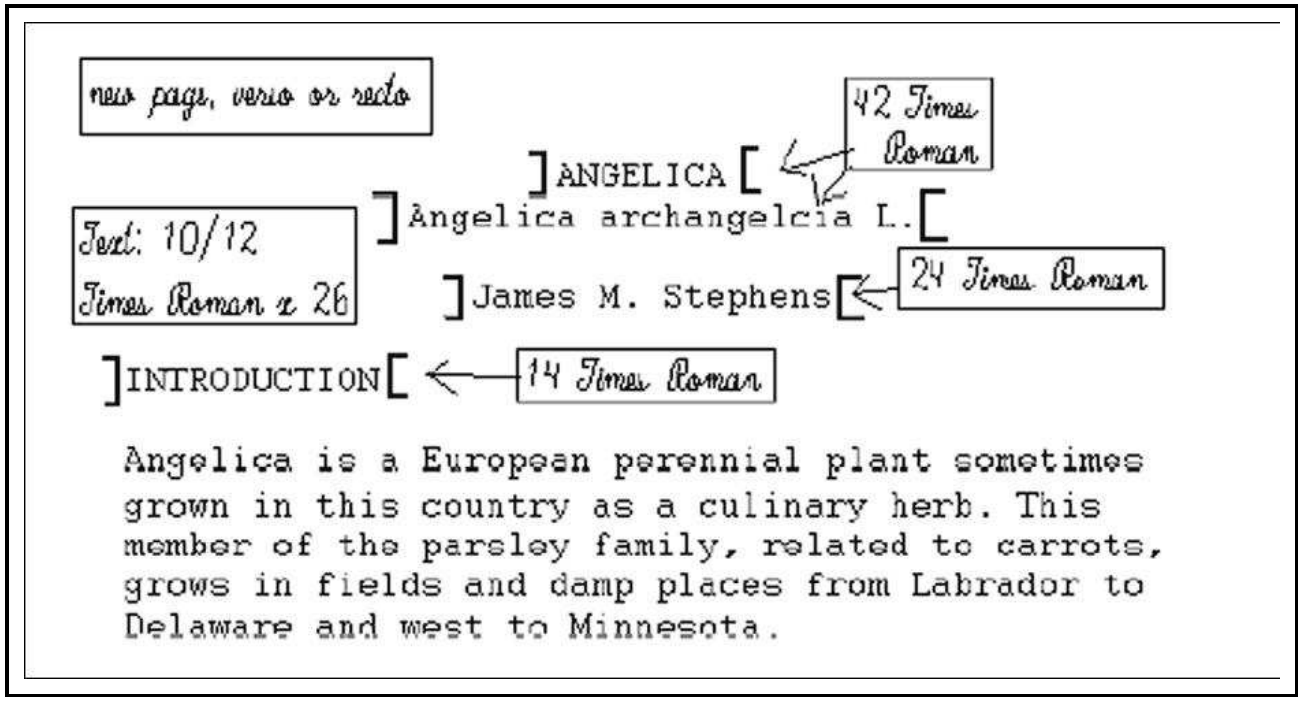
Carnegie Mellon University

http://www.cs.cmu.edu/~kohlhase

Carnegie Mellon

Document Markup for Mathematics

- Problem: Mathematical Vernacular and mathematical formulae have more structure than can be expressed in a linear sequence of standard characters
- Definition (Document Markup)
  Document markup is the process of adding codes to a document to identify the structure of a document or the format in which it is to appear.



Document Markup Systems for Mathematics

- M$ Word/Equation Editor: WYSYWIG, proprietary formatter/reader
  - \+ easy to use, well-integrated
  - – limited mathematics, expensive, vendor lock-in
- TEX/LATEX: powerful, open formatter (TEX), various readers (DVI/PS/PDF)
  - \+ flexible, portable persistent source, high quality math
  - – inflexible representation after formatting step
- HTML+GIF: server-side formatting, pervasive browsers
  - \+ flexible, powerful authoring systems LATEX/MATHEMATICA/...
  - – limited accessibility, reusability

Styles of Markup

- Definition (Presentation Markup)

  A markup scheme that specifies document structure to aid document processing by humans

  – e.g. `*roff`, Postscript, DVI, early MS Word, low-level TEX

  \+ simple, context-free, portable (verbatim), easy to implement/transform

  – inflexible, possibly verbose,

- Definition (Content Markup)

  A markup scheme that specifies document structure to aid document processing by machines or with machine support.

  – e.g. LATEX (if used correctly), Programming Languages, ATP input

  \+ flexible, portable (in spirit), unambiguous, language-independent

  – possibly verbose, context dependent, hard to read and write

Content vs. Presentation by Example

Language	Representation	Content?
LaTeX	`{\bf proof}:...\hfill\Box`	`\begin{proof}...\end{proof}`
HTML	`<font size="+2"><b>...</b></font>`	`<h1>...</h1>`
LISP	$8 + \sqrt{x}^3$	`(power (plus 8 (sqrt x)) 3)`
TeX	`$\{f\|f(0)> 0\;{\rm and}\;f(1)<0\}$`	$\{f \mid f(0) > 0 \text{ and } f(1) < 0\}$
TeX	`$\{f\|f(0)> 0$ and $f(1)<0\}$`	$\{f \mid f(0) > 0 \text{ and } f(1) < 0\}$

- We consider these to be representations of the same content (object)

Web Standards for Formula Markup

lanugage	MathML	OpenMath
by	W3C Math WG	OpenMath society
origin	math for HTML	integration of CAS
coverage	cont+pres; K-14	content; extensible
status	Version 2 (II 2001)	standard (IV 2001)
activity	maintenance	maintenance
Info	`http://w3c.org/Math`	`http://www.openmath.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*

- Plan of the talk
  - Philosophy (What is Math? Presentation vs. content, Why now?)
  - Concrete Language Elements (What you need to generate)
  - Tools, Deployment (Authoring, Browsers, Computation...)

Excursion: XML (Extensible Markup Language)

- XML is language family for the Web
  - tree representation language (begin/end brackets)
  - restrict instances by Doc. Type Def. (DTD) or XML Schema (Grammar)
  - Presentation markup by style files (XSL: XML Style Language)
- XML= extensible HTML∩ simplified SGML
- logic annotation (markup) instead of presentation!
- many tools available: parsers, compression, data bases, ...
- conceptually: transfer of directed graphs instead of strings.
- details at `http://www.w3c.org`

MathML Language Elements

Presentation first

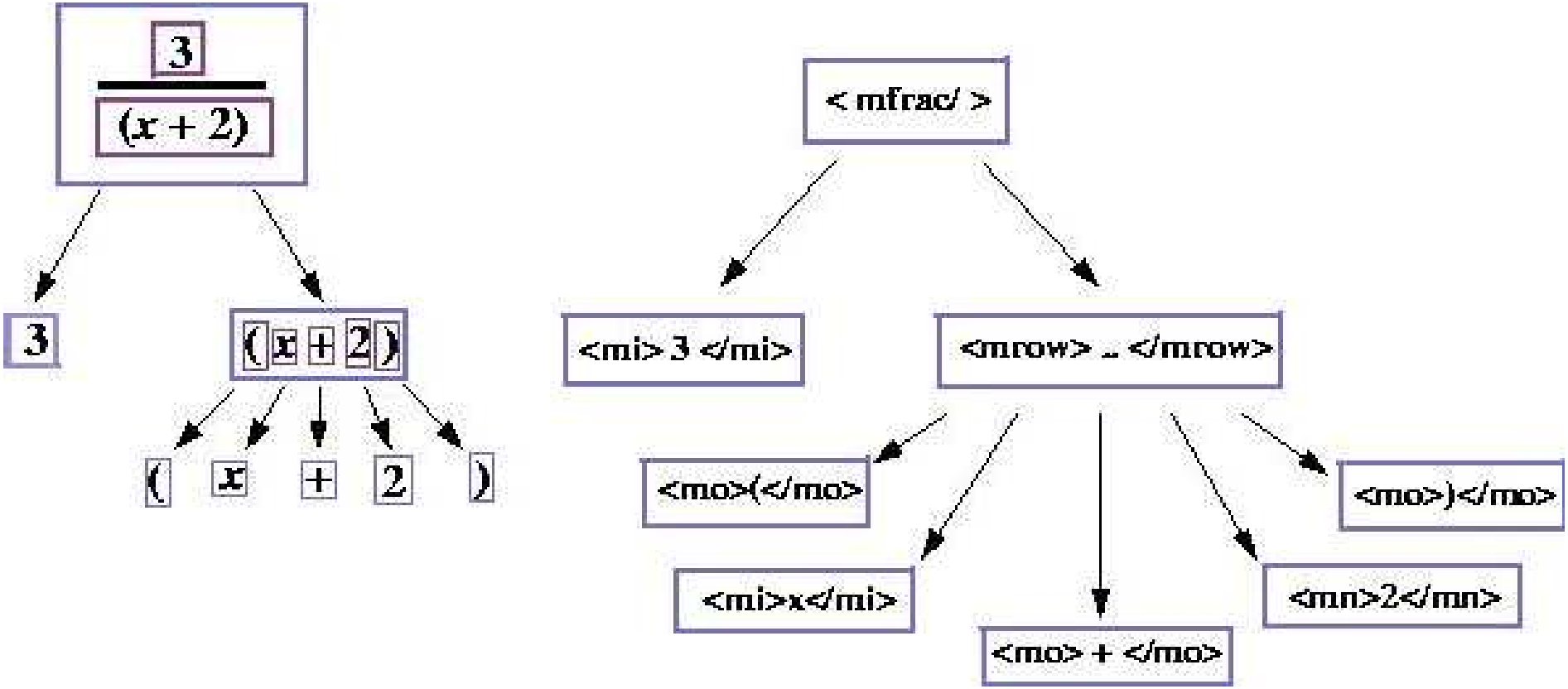
Representation of Formulae as Expression Trees

- Mathematical Expressions are build up as expression trees
  - of layout schemata in Presentation-MathML
  - of logical subexpressions in Content-MathML
- Example: $(a+b)^2$

```
<msup>
  <mfenced>
     <mi>a</mi>
     <mo>+</mo>
     <mi>b</mi>
   </mfenced>
   <mn>2</mn>
</msup>
```

```
<apply>
  <power/>
  <apply>
    <plus/>
    <ci>a</ci>
    <ci>b</ci>
  </apply>
  <cn>2</cn>
</apply>
```

Layout Schemata and the MathML Box model



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: `<mnumbers> ... </mi>` (∼ variables, italicized)
- 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).

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)

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`

$f\left(x+y\right)$	$f\left(x+y\right)$
<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

```
<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 $x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$

```
<mrow>
  <mi>x</mi>
  <mo>=</mo>
  <mfrac>
    <mrow>
      <mrow><mo>-</mo><mi>b</mi></mrow>
      <mo>&PlusMinus;</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, $R^i_j \ldots$
  - Super: `<msup> base script </msup>`
  - Subs: `<msub> base script </msub>`
  - Both: `<msubsup> base superscript subscript</msub>`

    (vertical alignment!)
- Bars and Arrows: $\overline{X}$, $\underbrace{Y}$, $\overline{Z}$,$\ldots$
  - Under: `<munder> base script</munder>`
  - Over: `<mover> base script</mover>`
  - Both: `<munderover> base underscript overscript </munderover>`
- Tensor-like: `<mmultiscripts> base sub1 sup1 ... [<mprescripts/> psub1 psup1 ...] </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

```
<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>
```

$$\sum_{i=1}^{\infty} x^i + \Sigma_{i=1}^{\infty} x^i$$

MathML Language Elements

Content MathML

Expression Trees in Prefix Notation

- 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>`
- Operators and Functions: ~ 100 empty elements `<sin/>`, `<plus/>`, `<eq/>`, `<compose/>`,...
- Token elements: `ci`, `cn` (identifiers and numbers)
- Extra Operators: `<csymbol definitionURL="...">...</csymbol>`

Containers aka Constructors

- sets: `<set> <elt1> <elt2> ... </set>` or
  `<set> <bvar>...</bvar> <condition> ...</condition> </set>`
- intervals: `<interval> <pt1> <pt2> </interval>`
  Attribute: `closure` (one of `open`, `closed`, `open-closed`, `closed-open`)
- vectors: `<vector> <elt1> <elt2> ... </vector>`
- matrix rows: `<matrixrow> <elt1> <elt2> ... </matrixrow>`
- matrices: `<matrix> <row1> <row2> ... </matrix>`

Examples of Content Math

Expression	Markup

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

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

`<apply><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>`
`</apply>`
	$\sum_{n=0}^{\infty} x^n$

`<apply><diff/>`
` <bvar><ci>x</ci><degree><cn>3</cn></degree></bvar>`
` <apply><ci>f</ci><ci>x</ci></apply>`
`</apply>`
	$\frac{d^3}{dx^3} f(x)$

`<set>`
` <bvar><ci>x</ci></bvar>`
` <bvar><ci>y</ci></bvar>`
` <condition>`
`  <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>`
`  </apply>`
` </condition>`
`</set>`
	$\left\{ x, y \;\middle|\; \begin{array}{l} 0 < x < 1, \\ 3 \leq y \leq 10 \end{array} \right\}$

Expression	Markup
<apply><eq/> <set> <bvar><ci>x</ci></bvar> <condition> <apply><geq/><ci>x</ci><cn>0</cn></apply> </condition> </set> <interval closure='closed-open'> <cn>0</cn> <ci>&infty;</ci> </interval> </apply>	$\{x\|x \geq 0\} = [0, \infty)$
<apply> <eq/> <apply><times/> <vector><cn>1</cn><cn>2</cn></vector> <matrix> <matrixrow><cn>0</cn><cn>1</cn></matrixrow> <matrixrow><cn>1</cn><cn>0</cn></matrixrow> </matrix> </apply> <apply> <transpose/> <vector><cn>2</cn><cn>1</cn></vector> </apply> </apply>	$(1,2) \times \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} = (2,1)^t$

Mixing Presentation and Content MathML

```
<semantics>
 <mrow>
  <mrow><mo>(</mo><mi>a</mi> <mo>+</mo> <mi>b</mi><mo>)</mo></mrow>
  <mo>&InvisibleTimes;</mo>
  <mrow><mo>(</mo><mi>c</mi> <mo>+</mo> <mi>d</mi><mo>)</mo></mrow>
 </mrow>
 <annotation-xml encoding="MathML-Content">
  <apply><times/>
   <apply><plus/><ci>a</ci> <ci>b</ci></apply>
   <apply><plus/><ci>c</ci> <ci>d</ci></apply>
  </apply>
 </annotation-xml>
 <annotation-xml encoding="openmath">
  <OMA><OMS cd="arith1" name="times"/>
   <OMA><OMS cd="arith1" name="plus"/><OMV name="a"/><OMV name="b"/></OMA>
   <OMA><OMS cd="arith1" name="plus"/><OMV name="c"/><OMV name="d"/></OMA>
  </OMA>
 </annotation-xml>
</semantics>
```

Parallel Markup in MathML

```
<semantics>
 <mrow id="E">
  <mrow id="E1">
   <mo id="E11">(</mo><mi id="E12">a</mi><mo id="E13">+</mo><mi id="E14">b</mi><mo id="E15">)</mo>
  </mrow>
  <mo id="E2">&InvisibleTimes;</mo>
  <mrow id="E3">
   <mo id="E31">(</mo><mi id="E32">c</mi><mo id="E33">+</mo><mi id="E34">d</mi><mo id="E35">)</mo>
  </mrow>
 </mrow>
 <annotation-xml encoding="MathML-Content">
  <apply xref="E">
   <times xref="E2"/>
   <apply xref="E1"><plus xref="E13"/><ci xref="E12">a</ci><ci xref="E14">b</ci></apply>
   <apply xref="E3"><plus xref="E33"/><ci xref="E32">c</ci><ci xref="E34">d</ci></apply>
  </apply>
 </annotation-xml>
</semantics>
```

Practical Considerations

I can write all this MathML now, how can I include it in my web-page or course materials.

Including MathML in your web page

- which browsers? (After all, we want to see it)
  - Internet Explorer ≥ 5.5 with Mathplayer/Techexplorer (behaviors)
  - Amaya (W3C Exploratory Browser) (native support of editing)
  - Mozilla/Netscape 7 (native presentation MathML)
  - all others with applets like WebEQ, CSS rendering
- Problem: how to identify the math markup to our browser, plug-in, or applet.
- Solution: insert MathML markup between `<math>` and `</math>`tags to distinguish MathML from HTML.
- Question: Does this really work cross-platform? (no!)

Including MathML in Web pages (Details)

- Use XHtML! MathML is an Xml application which does not mix well with HtML (tag soup parser)
- Use the MathML namespace! `http://www.w3.org/1999/xhtml`

```
<?xml version="1.0"?>
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>...</head>
  <body>
    <h1>Example</h1>
    ....
    <math
      xmlns="http://www.w3.org/1998/Math/MathML">
      <mi>x</mi><mo>+</mo><mn>3</mn>
    </math>
  </body>
</html>
```

This is the theory (according to the MathML Spec)

does it work?

not in practice!

Problems with 95% of the Browser Market

- M$ Intenet Explorer does not render XML (but XHTML is XML!)
- I.E. also does not implement MATHML natively (market too small)
- MathPlayer (Design Science), Techexplorer (IBM) plugins must be registered by magic incantations in the document head

```
<object id="mmlFactory"
        classid="clsid:32F66A20-7614-11D4-BD11-00104BD3F987"/>
```

Solution: David Carlisle's Universal XSLT style sheet.

- Idea: Make use of XSLT transformer in the browser
  - Amaya, Mozilla, Netscape 7: do nothing (Identity trafo)
  - M$ Internet Explorer: insert `<object>` element, adjust namespaces prefixes, transform to HTML.
- In practice? Add a stylesheet processing instruction

```
<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl"
   href="http://www.w3.org/Math/XSL/mathml.xsl"?>
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>...</head>
  <body>...</body>
</html>
```

Practical Considerations

- This will work in most cases, except if
- you are off-line ⇒ use a local copy of the style sheet

```
<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl" href="mathml.xsl"?>
<html xmlns="http://www.w3.org/1999/xhtml">
```

- the style sheet is not on the same server as XHTML+MATHML document
  - set IE security preferences to "low" (not recommended)
  - copy the style-sheets there
  - use off-line approach or embed style sheets into document

Coping with multiple renderers

- If a machine has more than one renderer, use XML namespaces for preferences.

```
<?xml-stylesheet type="text/xsl" href="pmathml.xsl"?>
<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:pref="http://www.w3.org/2002/Math/preference"
      pref:renderer="css">
```

- applicable values
  - `css`: render the equation through the use of CSS (no plug-in required)
  - `mathplayer-dl`: prompt to install MathPlayer if necessary.
  - `mathplayer`: use the MathPlayer behavior.
  - `techexplorer-plugin`: use the Techexplorer plug-in.
  - `techexplorer`: the Techexplorer rendering is preferred.

Back to the Content

Can't I just generate the presentation from other formats?

Content vs. Semantics in Math

- Content: logic-independent infrastructure
  Identification of abstract syntax, "semantics" by reference for symbols.

```
<apply>
  <plus/>
  <csymbol definitionURL="mbase://numbers/perfect#the-smallest"/>
  <cn>2</cn>
</apply>
```

- Semantics: establishing meaning by fixing consequences
  adds formal inference rules and axioms.
  - Mechanization in a specific system, or (Theorem Prover or Proof Checker)
  - logical framework (specify the logic in the system itself)

Semantics vs. Representation

e.g. Disjunction in Strong Kleene Logic

$\vee$	F	U	T
T	F	U	T
U	U	U	T
T	T	T	T

The binary minimum function on $\{T, U, F\}$, where $F \leq U \leq T$.

- We will consider these to be representations of different content objects
- That they are the same mathematically, is another matter.

Boolean Algebra	field of clopen subsets of a topological space	Stone Representation Theorem
Multi-modal K	Description Logic $\mathcal{ALC}$	[IJCAI93]
Solution of $f' = f$	$f(x) = 1 + \frac{x}{1!} + \frac{x^3}{2!} + \frac{x^3}{3!} + \cdots$	Complex Analysis II

From Content to Presentation

- Idea: manage the source in content markup scheme and generate presentation markup from that! (style sheets)
  - e.g. LaTeX with style/class files or HTML with Cascading Style Sheets, ...
  - device/language independence, personalization,.....
- Questions: (every content language has to deal with this somehow)
  - fixed or extensible set of language features?
  - specify presentation information (in what language?)
  - need a style sheet execution engine (client-side or server-side)
  - late binding problems, (where is my style sheet? what will it look like?)
- Problems: How to get back? (transformation loses information)

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)

OpenMath

- Extensible framework for specifying the content of mathematical objects
- Objects as (Deliberately uncommitted wrt. object semantics)
  - constants (OMS) (symbols: semantics given in theory)
  - variables (OMV) (local objects)
  - applications (OMA) (for functions)
  - bindings (OMBIND) (λ, quantification)
  - attributions (OMATTR) (e.g. for types)

```
<OMOBJ>
 <OMBIND>
  <OMS cd="quant1" name="forall"/>
  <OMBVAR><OMV name="a"/><OMV name="b"/></OMBVAR>
  <OMA><OMS cd="relation1" name="eq"/>
   <OMA><OMS cd="arith1" name="plus"/><OMV name="a"/><OMV name="b"/></OMA>
   <OMA><OMS cd="arith1" name="plus"/><OMV name="b"/><OMV name="a"/></OMA>
  </OMA>
 </OMBIND>
</OMOBJ>
```

C-MathMl and OpenMath are equivalent (almost)

OpenMath

```
<OMBIND>
 <OMS cd="quant1" name="forall"/>
 <OMBVAR>
  <OMATTR>
   <OMATP>
    <OMS cd="sts" name="type"/>
    <OMS cd="setname1" name="N"/>
   </OMATP>
   <OMV name="a"/>
  </OMATTR>
 </OMBVAR>
 <OMA>
  <OMS cd="relation1" name="geq"/>
  <OMV name="a"/>
  <OMI>0</OMI>
 </OMA>
</OMBIND>
```

MathMl

```
<apply>
 <forall/>
 <bvar>
  <ci type="nat">a</ci>
 </bvar>
 <apply>
  <geq/>
  <ci type="nat">a</ci>
  <cn>0</cn>
 </apply>
</apply>
```

Added-value services facilitated with Math Content

D1 cut and paste (cut output from web search engine and paste into CAS)

D2 automatically proof-checking formal argumentations (bridge verification?)

- math explanation (e.g. specialize a proof to a simpler special case)

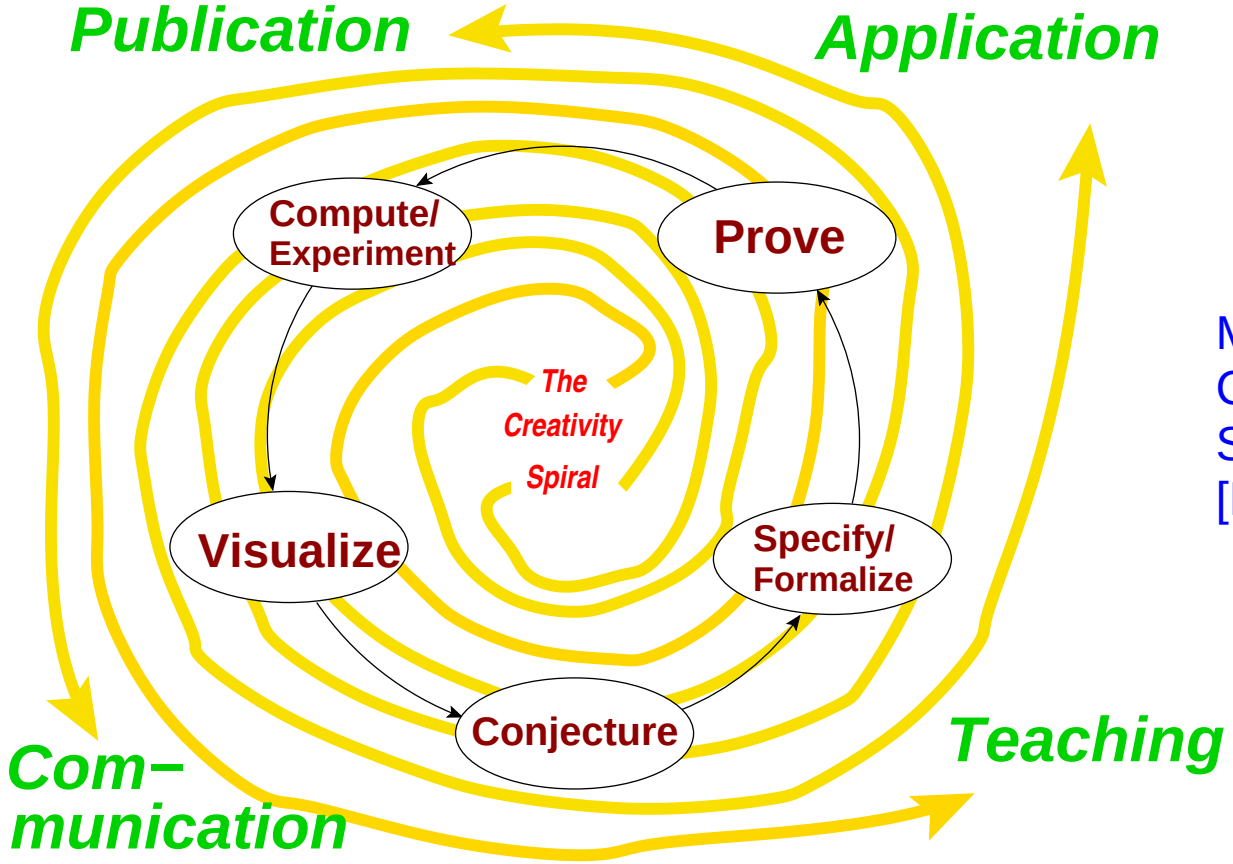
D3 semantical search for mathematical concepts (rather than keywords)

- data mining for representation theorems (find unnoticed groups out there)
- classification (given a concrete math structure, is there a general theory?)

D4 personalized notation (implication as $\rightarrow$ vs. $\supset$, or Ricci as $\frac{1}{2}\mathcal{R}^{ij}$ vs. $2\mathcal{R}^{ij}$)

D5 user-adapted documents (ActiveMath, Course Capsules)

Conclusion: The way we do math will change dramatically



Mathematical Creativity Spiral [Buchberber '95]

- Every step will be supported by mathematical software systems
- Towards an infrastructure for web-based mathematics!