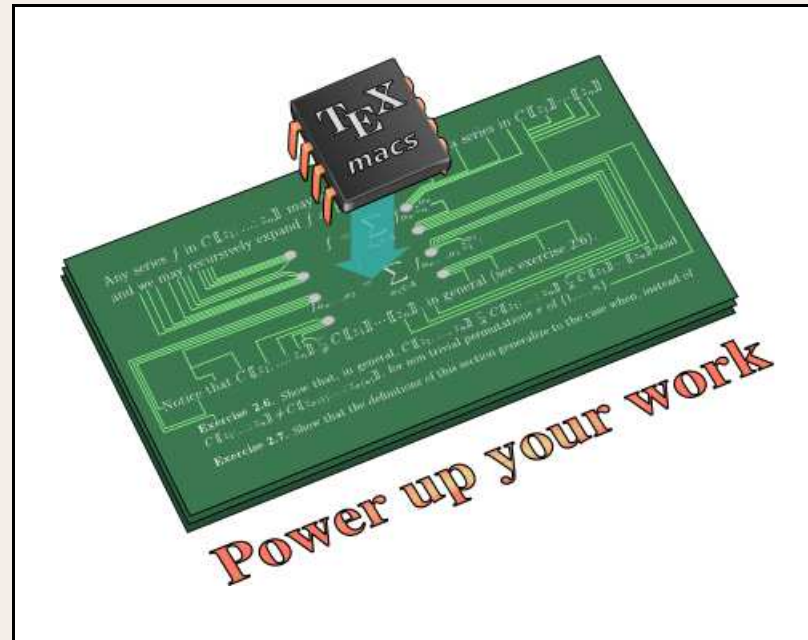


Towards semantic mathematical editing



Joris van der Hoeven, Palaiseau 2011
<http://www.TEXMACS.org>



Three levels of mathematical documents



Informal mathematics.

Software: text editors, mathematical user interfaces

Formats: L^AT_EX, presentation MATHML

Syntactically correct documents.

Software: computer algebra systems, scientific computation systems

Formats: content MATHML, software specific languages

Semantically correct documents.

Software: automatic theorem provers/checkers

Formats: OPENMATH, OMDOC, software specific languages



Three levels of mathematical documents



Informal mathematics.

Software: text editors, mathematical user interfaces

Example: $a(b+c)$

L^AT_EX

Syntactically correct documents.

Software: computer algebra systems, scientific computation systems

Formats: content MATHML, software specific languages

Semantically correct documents.

Software: automatic theorem provers/checkers

Formats: OPENMATH, OMDOC, software specific languages



Three levels of mathematical documents



Informal mathematics.

Software: text editors, mathematical user interfaces

Example:

Presentation MATHML

```
<mrow>
  <mi>a</mi> <mo>&InvisibleTimes;</mo>
  <mo>( </mo> <mi>b</mi> <mo>+</mo> <mi>c</mi> <mo>)</mo>
</mrow>
```

Syntactically correct documents.

Software: computer algebra systems, scientific computation systems

Formats: content MATHML, software specific languages

Semantically correct documents.

Software: automatic theorem provers/checkers

Formats: OPENMATH, OMDoc, software specific languages



Three levels of mathematical documents



Informal mathematics.

Software: text editors, mathematical user interfaces

Example: $a^*\langle\text{around}(|b+c|)\rangle$

TEX_{MACS}

Syntactically correct documents.

Software: computer algebra systems, scientific computation systems

Formats: content MATHML, software specific languages

Semantically correct documents.

Software: automatic theorem provers/checkers

Formats: OPENMATH, OMDoc, software specific languages



Three levels of mathematical documents



Informal mathematics.

Software: text editors, mathematical user interfaces

Example: $a * \langle \text{around} | (|b+c|) \rangle$

TEX_{MACS}

Syntactically correct documents.

Software: computer algebra systems, scientific computation systems

Example:

Content MATHML

```
<apply>
  <times/>
  <ci>a</ci>
  <apply>
    <plus/> <ci>b</ci> <ci>c</ci>
  </apply>
</apply>
```

Semantically correct documents.

Software: automatic theorem provers/checkers

Formats: OPENMATH, OMDOC, software specific languages



Three levels of mathematical documents



Informal mathematics.

Software: text editors, mathematical user interfaces

Example: $a * \langle \text{around} | (|b+c|) \rangle$

TEX_{MACS}

Syntactically correct documents.

Software: computer algebra systems, scientific computation systems

Example: $(* a (+ b c))$

Scheme

Semantically correct documents.

Software: automatic theorem provers/checkers

Formats: OPENMATH, OMDoc, software specific languages



Three levels of mathematical documents



Informal mathematics.

Software: text editors, mathematical user interfaces

Example: $a^*\langle\text{around}(|b+c|)\rangle$

TEX_{MACS}

Syntactically correct documents.

Software: computer algebra systems, scientific computation systems

Example: $(* a (+ b c))$

Scheme

Semantically correct documents.

Software: automatic theorem provers/checkers

Example: $a(b+c)$, where $a, b, c \in \mathbb{Z}$ and $+, \cdot: \mathbb{Z}^2 \rightarrow \mathbb{Z}$



The challenge



General purpose user interfaces.

Presentation oriented, no syntactic or semantic correctness.

Improved general purpose interfaces.

Presentation oriented interface *while* enforcing syntactic correctness

Interfaces for special systems.

May enforce syntactic or semantic correctness, application specific



The challenge



General purpose user interfaces.

Presentation oriented, no syntactic or semantic correctness.

Improved general purpose interfaces.

Presentation oriented interface *while* enforcing syntactic correctness

Interfaces for special systems.

May enforce syntactic or semantic correctness, application specific



User friendly extreme.

Rely on intelligent software to make sense out of presentation markup.

Programmer friendly extreme.

Let the user provide the explicit content markup.

Compromise.

- Make as much sense out of presentation markup as possible:
 - Automatic upgrading
 - Automatic syntax correction
 - Packrat parsing
- Provide markup for correcting the default interpretation, when needed.

Central technique.

- Presentation oriented interface
- Background converter presentation markup \rightsquigarrow content markup



Common ambiguities



Invisible operators.

- Invisible separators: $A = (a_{ij})$
- Invisible addition: $17\frac{3}{8}$
- Invisible “wildcards”: $+1$ (also denoted by $\cdot + 1$)
- Invisible ellipses: $\begin{pmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & & \vdots \\ a_{n1} & \cdots & a_{nn} \end{pmatrix}$
- Invisible zeros: $\begin{pmatrix} a_{11} & & \\ & \ddots & \\ & & a_{nn} \end{pmatrix}$
- Invisible brackets, for forced matching if we want to omit a bracket

Vertical bars.

- Brackets: absolute values $|x|$ or “ket” notation $\langle A|$
- “Such that” separators: $\{a \in X | a \geq 0\}$
- “Divides” predicate: $11 | 1001$
- Separators: $\langle a_1 | \cdots | a_n \rangle$.
- Restricting domains or images: $\varphi|_D, \varphi|^I$.

Punctuation.

- Commas: $f(x, y), 3,14159\dots, 1,000,000$

- Periods: 3.14159 , $\lambda x.x^2$, data access $a.b$, period.
- Semicolons: $\{x \in X : x \geq 0\}$, $P = (x : y : z) \in \mathbb{P}^2$, $a: \text{Int}$.
- Moreover, in the formula

$$a^2 + b^2 = c^2, \tag{1}$$

punctuation is used in the traditional, non-mathematical way.

Miscellaneous homoglyphs.

- Backslash \backslash : separator or “subtraction” of sets $X \backslash Y$
- Dot \cdot : multiplication $a \cdot b$ or wildcard $|\cdot|_p$
- Wedge \wedge : logical and $P \wedge Q$ or exterior product $dx \wedge dy$.

Good news.

THAT IS ABOUT IT! ...but inconsistent support in Unicode



Syntax correction



Invisible tag correction.

Breaks: $\begin{math}a+\end{math}\begin{math}b\end{math}$.

Redundancies I: $\begin{math}a+\begin{math}b\end{math}\end{math}$.

Redundancies II: $\begin{math}\text{hi}\end{math}$.

Bracket matching. Match in decreasing order of likeliness

Example: $f(|x|) = g(|x| + |y|)$

Match $(|x|)$ and $(|x| + |y|)$ at an early stage.

Match $|x|$, $|x|$ and $|y|$ at a later stage, with higher security

Bracket motion. Let $y=f(x)$. \rightsquigarrow Let $y=f(x)$.

Superfluous invisible removal. $\frac{a^*}{b}$ \rightsquigarrow $\frac{a}{b}$

Missing invisible insertion. $2x$ \rightsquigarrow $2*x$, Lf \rightsquigarrow L_f .

Homoglyph substitution. $a:=b$ \rightsquigarrow $a:=b$

Miscellaneous corrections. $a^{\wedge x}$ \rightsquigarrow a^x



Experimental results



BPR: Algorithms in Real Algebraic Geometry (Basu, Pollack, Roy)

COL: Collection of class notes (Evan Chou)

LN: Transseries and Real Differential Algebra (vdH)

Document	BPR	BPR ₁	BPR ₂	COL	COL ₃	LN
Total # of formulas	30394	883	2693	13048	2092	12626
Initial # of errors	2821	63	221	4158	607	629
# after correction	705	35	53	543	37	98
Number of pages	585	16	48	357	56	233

Table. Performance of the $\text{T}_{\text{E}}\text{X}_{\text{MACS}}$ syntax corrector on various documents.

Demo 1: [BPR₂](#)

Demo 2: [Arxiv₁](#)



Experimental results



BPR: Algorithms in Real Algebraic Geometry (Basu, Pollack, Roy)

COL: Collection of class notes (Evan Chou)

LN: Transseries and Real Differential Algebra (vdH)

Document	BPR	BPR ₁	BPR ₂	COL	COL ₃	LN
Invisible tag correction	92	0	3	10	5	56
Bracket matching	676	14	67	1236	94	282
Bracket motion	16	0	4	6	0	4
Superfluous invisible removal	382	1	25	1046	305	149
Missing invisible insertion	873	11	56	1271	164	33
Homoglyph substitution	44	2	7	45	2	6
Miscellaneous corrections	16	0	6	1	0	1

Table. Numbers of corrections due to individual algorithms.



Experimental results



BPR: Algorithms in Real Algebraic Geometry (Basu, Pollack, Roy)

COL: Collection of class notes (Evan Chou)

HAB: Habilitation (vdH)

Document	BPR ₁	COL ₃	HAB ₁	HAB ₅
Invisible operator confusion	some	many		some
Informal list notation	several		several	some
Non marked text inside formulas	several	many		
Non marked formulas inside text	some			
Misinterpreted meaningful whitespace			several	
Miscellaneous misinterpretations		some		

Table. Manual determination of common sources of misinterpretation.



- **What kind of Parser?**

Packrat parsers are fast ($O(ns)$ parsing, n : #input, s : #grammar)

Packrat parsers are flexible (no preprocessing required)

Packrat parsers are general (better set of recognized languages)

- **What kind of grammar?**

Non-structured string grammar \Rightarrow flattening of structured texts

Universal grammar for mathematics

Might support other grammars for (e.g.) automatic theorem provers

- **How to allow for customization?**

Exploit $\text{T}_{\text{E}}\text{X}_{\text{MACS}}$ built-in macro system

Allow for “behaves as” annotations



Snippets of the universal grammar



```
(define Plus-symbol
  (:type infix)
  (:penalty 30)
  (:spacing default default)
  "+" "<amalg>" "<oplus>" "<boxplus>"
  "<dotplus>" "<dotamalg>" "<dotoplus>")
```

```
(define Plus-infix
  (:operator associative)
  (Plus-infix Post)
  (Pre Plus-infix)
  Plus-symbol)
```

```
(define Pre
  (:selectable inside)
  (:<lsub Script :>)
  (:<lsup Script :>)
  (:<lprime (* Prime-symbol) :>))
```

```
(define Post
  (:selectable inside)
  (:<rsub Script :>)
  (:<rsup Script :>)
  (:<rprime (* Prime-symbol) :>))
```

```
(define Sum
```

```
  (Sum Plus-infix Product)
```

```
  (Sum Minus-infix Product)
```

```
  Sum-prefix)
```



How much structure in the document?



Brackets.

Previously: no structure in document

$$f(x + y) + a(b + c)$$

Currently: **around** tag

$$f(x + y) + a(b + c)$$

Subscripts and superscripts.

Base not in document

↪ either requires parsing or hard to edit

Scripts physically glued to content on direct left

$$a + \lim_{i \rightarrow \infty} x_i$$

Big operators.

Scope not in document

Prefix operators, arguments variable priorities

$$\sum_{i=1}^n a_i + \sum_{i=1}^p b_i c_i$$



Grammar rules for big operators



```
(define Big-sum-symbol  
  "int" "oint" "intl" "ointl" "sum" "oplus" "triangledown")
```

```
(define Big-sum  
  (:operator)  
  (Big-sum Post)  
  (:<big Big-sum-symbol :>))
```

```
(define Prefixed  
  (Big-separator Expression)  
  (Big-or Conjunction)  
  (Big-and Negation)  
  (Big-union Intersection)  
  (Big-intersection Sum)  
  (Big-sum Sum-prefix)  
  (Big-product Power)  
  (Prefix-prefix Prefixed)  
  (Pre Prefixed)  
  (Postfixed Space-infix Prefixed)  
  Postfixed)
```



Customization



“Behaves as” annotation.

$$a_1 + \cdots + a_n$$

Macro expansion.

```
<assign|pt|<macro|<math-times|<superpose|+|×>>>>
```

$$a * b + x * y$$

Customizing the interface.

```
Scheme] (kbd-map (:mode in-math?) ("+ *" (make 'pt)))
```

```
Scheme]
```



Non parsable formulas



Meaningful whitespace or text.

$$\exists x \ P(x)$$

$$\left\{ b < \mathbb{N}_0 \mid b < d \text{ and } \frac{\det A}{d} \mathbb{Z} + b \mathbb{Z} + d \mathbb{Z} = \mathfrak{d}_1(A) \right\}$$

Using mathematics as a replacement for text.

$$a) \Rightarrow b)$$

Manual hyphenation on tables.

$$\text{posgcd}(\mathcal{P} \cup \{P\}) = \{(\text{Pol}(p(L)), \mathcal{C} \wedge \mathcal{C}_L) \mid (Q, \mathcal{C}) \in \text{posgcd}(\mathcal{P})$$

$$\text{and } L \text{ leaf of } \text{TRems}(P, Q)\}.$$

Unsuccessful automatic correction.

$$\text{T}_{*\nu}^j$$

$$\text{\$a+\$ \$b\$}$$

$$\mathcal{B} = \{a + i b \mid |b| \leq -\sqrt{3} a\}$$

Non standard operator precedence.

Given a formula $\Theta(Y) = (\exists X) \mathcal{B}(X, Y)$, where \mathcal{B} is...

Abusive visual twiddling.

$$\prod_{\substack{i < j, k < \ell \\ (i, j) < (k, \ell)}} (\alpha_{i, j, k, \ell} + Z \beta_{i, j, k, \ell})^2$$

Visual twiddling with no obvious alternative.

1. The signs of the polynomials in the Sturm sequence are $+ - + - +$
2. No other code word is of the form $0.z_1 \dots z_l(x) * \dots$
3. $g = e^{\sqrt{x} + e^{\sqrt{\log x} + e^{\sqrt{\log \log x} + \dots + \log \log \log x} + \log \log x} + \log x}$



Incorrectly parsed formulas



Incorrect invisible operators.

1. $b(12c + a^2)$

2. $\sim R(n) s$

Informal list notation.

1. $SRemS(P, Q) = SRemS_0(P, Q), \dots, SRemS_k(P, Q)$

2. The subsets of affine spaces \mathbb{R}^n for $n = 0, 1, 2, \dots$ that are...

3. $H(Y_i | Y^{i-1}, X^n)$

Non marked text inside formulas.

$$p(\hat{x}|x) = 1 \text{ if } \hat{x} = 0$$

Misinterpretation of meaningful whitespace.

$$P(f) = 0 \quad (f \prec \mathfrak{F}).$$

Miscellaneous misinterpretations.

$$\max_{p, q, r} -1 - p \log p - q \log q - r \log r - (1 - p - q - r) \log(1 - p - q - r)$$



Semantic selections



Selectable subexpressions.

Associative operators

$$a + b - c + d$$

Selections as unary operators when pasting

$$+a + b \quad \rightsquigarrow \quad x \longmapsto x + a + b$$

Second example

$$\frac{1}{1 +} \quad \rightsquigarrow \quad x \longmapsto \frac{1}{1 + x}$$

Problematic transformations.

$$a_1, \dots, a_n \in A \quad \rightsquigarrow \quad (a_1, \dots, a_n) \in A$$



Spacing.

1. Binary subtraction $x - y$
2. Unary negation $-x$
3. Abstract operators $\square \in \{+, -, \cdot\}$

Hyphenation transforms.

$$\frac{a + b + c}{x + y + z} \rightsquigarrow (a + b + c) / (x + y + z)$$

$$\frac{1}{a + b + c + x + y + z} \rightsquigarrow 1 / (a + b + c + x + y + z)$$