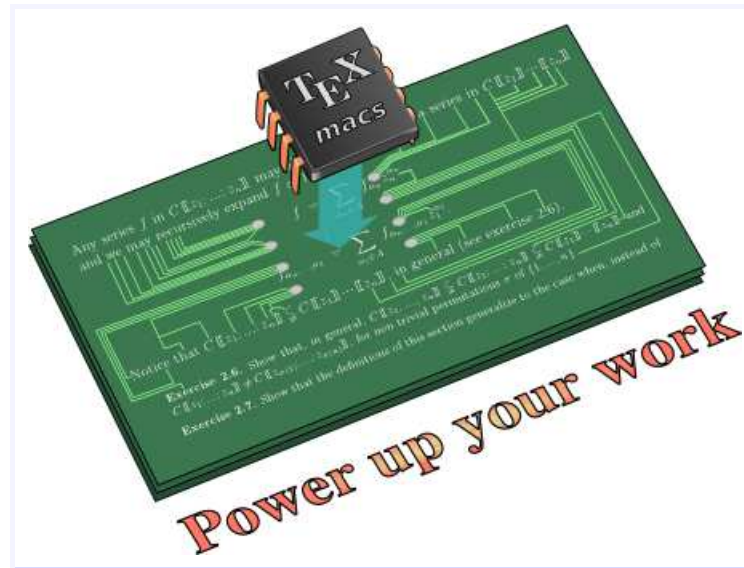


Preserving Syntactic Correctness While Editing Mathematical Formulas

Joris van der Hoeven, Grégoire Lecerf, Denis Raux

CNRS, École polytechnique



Kalamata, July 20, 2015

<http://www.TEXMACS.org>

1 2 3 4 5 6 7 8 9 10 11

No semantics

L^AT_EX

Code
`$a+bc$`

1 2 3 4 5 6 7 8 9 10 11

No semantics

L^AT_EX

Code
\$a+bc\$

Presentation semantics

Presentation MathML
Classical T_EX_{MACS}

Specification
a+b*c

1 2 3 4 5 6 7 8 9 10 11

No semantics

L^AT_EX

Code
\$a+bc\$

Presentation semantics

Presentation MathML
Classical T_EX_{MACS}

Specification
a+b*c

Content semantics

Content MathML

Syntax tree
(+ a (* b c))

1 2 3 4 5 6 7 8 9 10 11

No semantics	L ^A T _E X	Code \$a+bc\$
Presentation semantics	Presentation MathML Classical T _E X _{MACS}	Specification a+b*c
Content semantics	Content MathML	Syntax tree (+ a (* b c))
Concept semantics	OpenMath	Annotated tree + commutative, ...

1 2 3 4 5 6 7 8 9 10 11

No semantics	L ^A T _E X	Code \$a+bc\$
Presentation semantics	Presentation MathML Classical T _E X _{MACS}	Specification a+b*c
Content semantics	Content MathML	Syntax tree (+ a (* b c))
Concept semantics	OpenMath	Annotated tree + commutative, ...
Full semantics	Coq, Hol, ...	Proof script

1 2 3 4 5 6 7 8 9 10 11

No semantics	L ^A T _E X	Code \$a+bc\$
Presentation semantics	Presentation MathML Classical T _E X _{MACS}	Specification a+b*c
Syntactical semantics	Experimental T _E X _{MACS} Various CAS systems	Grammar enhanced
Content semantics	Content MathML	Syntax tree (+ a (* b c))
Concept semantics	OpenMath	Annotated tree + commutative, ...
Full semantics	Coq, Hol, ...	Proof script

- Computer algebra formulas and programs

Restricted grammars; possibility to design *ad hoc* editors for a fixed grammar

1 2 3 4 5 6 7 8 9 10 11

- Computer algebra formulas and programs

Restricted grammars; possibility to design *ad hoc* editors for a fixed grammar

- Formally correct mathematical texts

Flexible grammars; focus on proofs, not on presentation or readability

1 2 3 4 5 6 7 8 9 10 11

- Computer algebra formulas and programs

Restricted grammars; possibility to design *ad hoc* editors for a fixed grammar

- Formally correct mathematical texts

Flexible grammars; focus on proofs, not on presentation or readability

- Informal, general purpose mathematical texts

Very flexible notations; emphasis on user friendliness for authors

1 2 3 4 5 6 7 8 9 10 11

Classical T_EX_{MACS}

$$x_{1,2} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

1 2 3 4 5 6 7 8 9 10 11

Classical T_EX_{MACS}

$$x_{1,2} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

$$M = \begin{pmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_n \end{pmatrix}$$

1 2 3 4 5 6 7 8 9 10 11

Classical T_EX_{MACS}

$$x_{1,2} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

$$M = \begin{pmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_n \end{pmatrix}$$

$$\varphi = \frac{1}{1 + \frac{1}{1 + \frac{1}{1 + \frac{1}{1 + \frac{1}{1 + \ddots}}}}}$$

1 2 3 4 5 6 7 8 9 10 11

Semantically aware T_EX_{MACS}

$$x_{1,2} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

1 2 3 4 5 6 7 8 9 10 11

Semantically aware T_EX_{MACS}

$$x_{1,2} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

$$M = \begin{pmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_n \end{pmatrix}$$

1 2 3 4 5 6 7 8 9 10 11

Semantically aware $\text{T}_{\text{E}}\text{X}_{\text{MACS}}$

$$x_{1,2} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

$$M = \begin{pmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_n \end{pmatrix}$$

$$\varphi = \frac{1}{1 + \frac{1}{1 + \frac{1}{1 + \frac{1}{1 + \frac{1}{1 + \ddots}}}}}$$

Enforcing semantic correctness

$$x_{1,2} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

Enforcing semantic correctness

$$x_{1,2} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

$$M = \begin{pmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_n \end{pmatrix}$$

1 2 3 4 5 6 7 8 9 10 11

Enforcing semantic correctness

$$x_{1,2} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

$$M = \begin{pmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_n \end{pmatrix}$$

$$\varphi = \frac{1}{1 + \frac{1}{1 + \frac{1}{1 + \frac{1}{1 + \frac{1}{1 + \ddots}}}}}$$

1 2 3 4 5 6 7 8 9 10 11

- **Presentation oriented editing**
 - Making editor as graphical as possible
 - Design efficient and easy to remember keyboard shortcuts
 - Include powerful macro system for new notations

1 2 3 4 5 6 7 8 9 10 11

- **Presentation oriented editing**
 - Making editor as graphical as possible
 - Design efficient and easy to remember keyboard shortcuts
 - Include powerful macro system for new notations

- **Checking syntactic correctness**
 - Design of a “universal grammar” for informal mathematical notations
 - Compatibility with macro system for the introduction of new notations

1 2 3 4 5 6 7 8 9 10 11

- **Presentation oriented editing**

- Making editor as graphical as possible
- Design efficient and easy to remember keyboard shortcuts
- Include powerful macro system for new notations

- **Checking syntactic correctness**

- Design of a “universal grammar” for informal mathematical notations
- Compatibility with macro system for the introduction of new notations

- **Enforcing syntactic correctness while editing**

- How to add/remove “transient markup” in order to maintain correctness?
- Guarantee same editing behaviour as for the presentation oriented mode
- Guarantee correctness for *all* editing operations

1. Write a procedure “correct”, taking any formula on input and returning its correction, modulo inserting/removing transient markup only.
2. For each editing operation, apply the procedure “correct” to all modified formulas in all documents.

1. Write a procedure “correct”, taking any formula on input and returning its correction, modulo inserting/removing transient markup only.
2. For each editing operation, apply the procedure “correct” to all modified formulas in all documents.

Problems

- Writing the procedure “correct” is non trivial

1. Write a procedure “correct”, taking any formula on input and returning its correction, modulo inserting/removing transient markup only.
2. For each editing operation, apply the procedure “correct” to all modified formulas in all documents.

Problems

- Writing the procedure “correct” is non trivial
- How to select “best” correction among all possible corrections

1. Write a procedure “correct”, taking any formula on input and returning its correction, modulo inserting/removing transient markup only.
2. For each editing operation, apply the procedure “correct” to all modified formulas in all documents.

Problems

- Writing the procedure “correct” is non trivial
- How to select “best” correction among all possible corrections
- Strict application violates transparency w.r.t. presentation oriented editing:

$$a + \square | \xrightarrow{\text{—}} \begin{array}{l} a + \square \square \\ a + \square \square \end{array}$$

1. \forall editing action, determine a list of eligible correction schemes.
2. Try each eligible correction scheme until obtaining a correct formula.
3. If no schemes succeeded, then cancel the editing action.

1. \forall editing action, determine a list of eligible correction schemes.
2. Try each eligible correction scheme until obtaining a correct formula.
3. If no schemes succeeded, then cancel the editing action.

Challenges

- Design suitable correction schemes for the most common editing actions.

1. \forall editing action, determine a list of eligible correction schemes.
2. Try each eligible correction scheme until obtaining a correct formula.
3. If no schemes succeeded, then cancel the editing action.

Challenges

- Design suitable correction schemes for the most common editing actions.
- Completeness (so that step 3 is never reached).

1. \forall editing action, determine a list of eligible correction schemes.
2. Try each eligible correction scheme until obtaining a correct formula.
3. If no schemes succeeded, then cancel the editing action.

Challenges

- Design suitable correction schemes for the most common editing actions.
- Completeness (so that step 3 is never reached).
- Compatibility with “undo/redo” mechanism.

1. \forall editing action, determine a list of eligible correction schemes.
2. Try each eligible correction scheme until obtaining a correct formula.
3. If no schemes succeeded, then cancel the editing action.

Challenges

- Design suitable correction schemes for the most common editing actions.
- Completeness (so that step 3 is never reached).
- Compatibility with “undo/redo” mechanism.
- Correctness under all circumstances (e.g. editing operations that modify several formulas).

1 2 3 4 5 6 7 8 9 10 11

Basic insertion scheme

- Remove transient markup around cursor
- Apply insertion
- Insert transient box at cursor position if needed

| \xrightarrow{A} a | $\xrightarrow{+}$ a + | □ \xrightarrow{B} a + b |

1 2 3 4 5 6 7 8 9 10 11

Basic insertion scheme

- Remove transient markup around cursor
- Apply insertion
- Insert transient box at cursor position if needed

$| \xrightarrow{A} a | \xrightarrow{+} a + | \square \xrightarrow{B} a + b |$

Starting a prime or right script after a transient box

- Move to the left of transient markup around cursor
- Apply insertion
- Insert transient box at cursor position if needed

$x + | \square \xrightarrow{-1} x + | \square \xrightarrow{-2} x + | \square \xrightarrow{-3} x + | \square$

1 2 3 4 5 6 7 8 9 10 11

Insert content in the middle of an operator

- Remove transient markup around cursor
- Insert transient “explicit spaces” before and after the cursor
- Perform the insertion
- Add further transient boxes if needed



1 2 3 4 5 6 7 8 9 10 11

Insert content in the middle of an operator

- Remove transient markup around cursor
- Insert transient “explicit spaces” before and after the cursor
- Perform the insertion
- Add further transient boxes if needed



Removal of actual infix operators

- Remove transient markup around cursor
- Perform the deletion
- Add transient version of the deleted infix operator after the cursor
- Add further transient boxes if needed



1 2 3 4 5 6 7 8 9 10 11

- Informal content

$$Z = \{i \in I: f_i(x) = 0 \text{ and } g_i(x) = 0 \text{ almost everywhere}\}$$

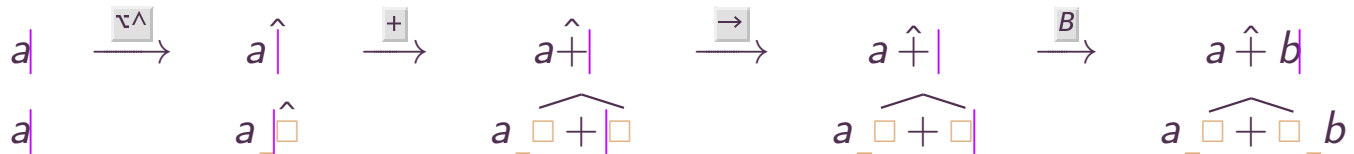
$$\blacksquare = \{i \in I: (f_i^2 + g_i^2)(x) = 0 \text{ almost everywhere}\}.$$

- Informal content

$$Z = \{i \in I: f_i(x) = 0 \text{ and } g_i(x) = 0 \text{ almost everywhere}\}$$

$$\blacksquare = \{i \in I: (f_i^2 + g_i^2)(x) = 0 \text{ almost everywhere}\}.$$

- Missing schemes for “weird” editing operations



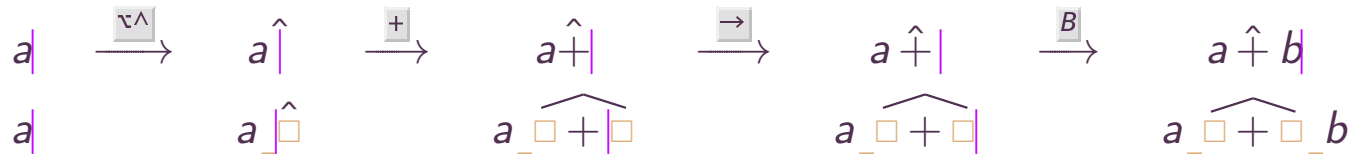
1 2 3 4 5 6 7 8 9 10 11

- Informal content

$$Z = \{i \in I: f_i(x) = 0 \text{ and } g_i(x) = 0 \text{ almost everywhere}\}$$

$$\blacksquare = \{i \in I: (f_i^2 + g_i^2)(x) = 0 \text{ almost everywhere}\}.$$

- Missing schemes for “weird” editing operations



- Wildly varying notations for quantified expressions

$\forall x, \exists y, P(x, y)$
 $\forall x \exists y: P(x, y)$
 $(\forall x)(\exists y)P(x, y)$
 \vdots



1 2 3 4 5 6 7 8 9 10 11

- **Implicit zeros and such**

$$\begin{pmatrix} \square & \square \\ \square & \square \end{pmatrix} \text{ versus } \begin{pmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_n \end{pmatrix}$$

1 2 3 4 5 6 7 8 9 10 11

- **Implicit zeros and such**

$$\begin{pmatrix} \square & \square \\ \square & \square \end{pmatrix} \text{ versus } \begin{pmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_n \end{pmatrix}$$

- **How permissive should the universal grammar be?**

$$f_n; = f_n z^n + f_{n+1} z^{n+1} + \dots$$

$$L_{\times f}(g) = L(fg)$$

$$f_n \xrightarrow{i} ???$$

1 2 3 4 5 6 7 8 9 10 11

- **Implicit zeros and such**

$$\begin{pmatrix} \square & \square \\ \square & \square \end{pmatrix} \text{ versus } \begin{pmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_n \end{pmatrix}$$

- **How permissive should the universal grammar be?**

$$f_n; = f_n z^n + f_{n+1} z^{n+1} + \dots$$

$$L_{\times f}(g) = L(fg)$$

$$f_n; \xrightarrow{i} ???$$

- **Undoing “save the current selection as an image”**

1 2 3 4 5 6 7 8 9 10 11

- **Implicit zeros and such**

$$\begin{pmatrix} \square & \square \\ \square & \square \end{pmatrix} \text{ versus } \begin{pmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_n \end{pmatrix}$$

- **How permissive should the universal grammar be?**

$$f_n; = f_n z^n + f_{n+1} z^{n+1} + \dots$$

$$L_{\times f}(g) = L(fg)$$

$$f_n \xrightarrow{i} ???$$

- **Undoing “save the current selection as an image”**
- **Increased semantics?** Replace y by $a + b$ in $x \cdot y \rightsquigarrow x \cdot a + b$ or $x \cdot (a + b)$?

1 2 3 4 5 6 7 8 9 10 11

- **Implicit zeros and such**

$$\begin{pmatrix} \square & \square \\ \square & \square \end{pmatrix} \text{ versus } \begin{pmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_n \end{pmatrix}$$

- **How permissive should the universal grammar be?**

$$f_n; = f_n z^n + f_{n+1} z^{n+1} + \dots$$

$$L_{\times f}(g) = L(fg)$$

$$f_n \xrightarrow{i} ???$$

- **Undoing “save the current selection as an image”**
- **Increased semantics?** Replace y by $a + b$ in $x \cdot y \rightsquigarrow x \cdot a + b$ or $x \cdot (a + b)$?
- **Unclear semantics for certain expressions**

Differences between $a + bc - d$ and $a + bcy$ yield $a + bc - dy$ in versioning tool.