CS490y Undergraduate Thesis Presentation

Organizing the Structure of Mathematical Expressions

Clare M. So
clare@scl.csd.uwo.ca

This thesis is supervised by **Dr. Stephen M. Watt**

April 1, 2003

mathml @ORCCA

What is the problem?

- Re-associate symbols and characters in *single-line* mathematical expressions while preserving expressions' implicit semantics
- Generate *Presentation MathML* as the result of the analysis

Keywords: MathML, Mathematical Handwriting Recognition

Why is this problem interesting?

- Need to recognize and treat mathematical expressions by computers in a meaningful manner
  - Mathematical handwriting recognizer
  - TEX/MathML converter
- Capture the semantics of mathematics so that the mathematical expressions can be:
  - Computed using a Computer Algebra System
  - Stored in databases

An example of (bad) character re-association

- A TEX markup of $\int_a^b (x+3)^2 dx$:

```
$ \int_{a}^{b}\! {(x+3} {)}^{2} {dx} $
```

- Curly brackets indicate implicit groupings in TEX
  - Do we actually mean to apply the exponent to ")" in "`{)}^{2}`"?
  - Do we actually mean to group "$(x+3$" together in "`{(x+3}`"?
  - etc.

We can see...

- If we input math by handwriting
  - How to associate characters and symbols using two-dimensional information provided?
  - How to make handwriting recognizers to recognize two-dimensional data?
  - A picture of an expression does not capture the semantics!
  - ...

How can one extract the semantics of mathematics from the representation of the expressions?

Previous works

- In University of Western Ontario
  - Bo Wan, a former member of ORCCA, developed a mathematical handwriting recognizer for the pocket PC
- In Université de Nice, Sophia-Antipolis, France
  - Stéphane Lavirotte developed an OFR (Optical Formula Recognition) to recognize mathematics in documents
  - A set of *graph grammars* defines permitted two-dimensional relationships between characters and symbols in mathematical expressions

Both of these works contain similar discussions

A brief overview of MathML

- A W3C recommendation to put mathematics on the web
- Looks a bit like HTML
- XML (eXtensible Markup Language)
- Natively supported by Netscape 7 and Mozilla
- Two kinds of markup:

  Presentation MathML encodes how the mathematical expressions look

  Content MathML encodes the semantics and the meaning of mathematical expressions

- We work with *Presentation MathML* in this project

- Example: MathML markup for "x^3":
  - Content MathML

```
<math xmlns="http://www.w3.org/MathML">
   <apply>
      <exp/>
      <ci> x </ci>
      <cn> 3 </cn>
   </apply>
</math>
```

  - Presentation MathML

```
<math xmlns="http://www.w3.org/MathML">
   <mrow>
      <msup>
         <mi> x </mi>
         <mn> 3 </mn>
      </msup>
   </mrow>
</math>
```

Properties of mathematical expressions

- *Presentation* VS *Content*
  - Presentation
    - concerns how the expressions look
  - Content
    - concerns the semantics of expressions

We intuitively draw the relationships between the presentation and the content of mathematics

- Example: "x^3" written in TeX, and OpenMath:
  - TeX:

```
$ {x}_{}^{3} $
```

  - OpenMath:

```
<OMOBJ>
   <OMA>
      <OMS cd="tranc1" name="exp"/>
      <OMV name="x"/>
      <OMI> 3 </OMI>
   </OMA>
</OMOBJ>
```

- Two-Dimensional
  - How can we know that "x^3" does not equals to "$x3$"?
  - Two-dimensional relationships are vital to determine the content of mathematical expressions
  - The relationships are:
    - *Subscript* (Example: x_y)
    - *Superscript* (Example: x^y)
    - *Underscript* (Example: $\underset{\cdot}{x}$)
    - *Overscript* (Example: $\dot{x}$)
    - *Presuperscript* (Example: ${}^yF$)
    - *Presubscript* (Example: ${}_yF$)
    - *Inline* (Example: xy)
    - *Include* (Example: $\sqrt{x}$)

- Uses of notations are arbitrary
  - Precedence
    - **"BEDMAS"**
      - **B**racket
      - **E**xponent
      - **D**ivision
      - **M**ultiplication
      - **A**ddition
      - **S**ubtraction
  - Number of arguments
    - Unary, binary, and n-ary
  - Some of the arguments are compulsory
    - Why $\int(x+3)^2dx$ is valid and $\int_0(x+3)^2dx$ is not?

- Location of Operators
  - *Prefix* (Example: $-x$)
  - *Infix* (Example: $1+2+3$)
  - *Postfix* (Example: $3!$)
  - *Bounding* (Example: $[a, b]$)
  - *Implicit* (Example: x^y)
  - *Two-dimensional* (Example: $\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$)
  - *Include* (Example: $\sqrt{x}$)

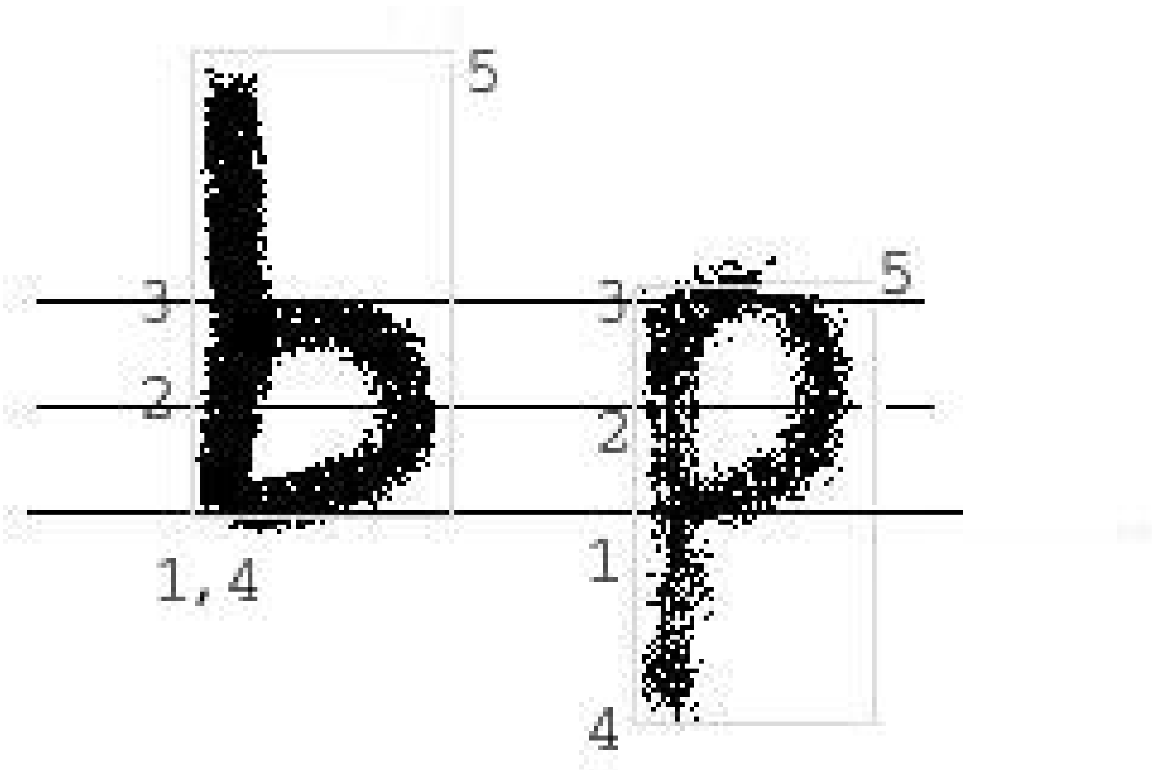
- Meanings must be determined globally
  - Example: a dot in "3.5", "$3 \cdot 5$" and "$\ldots$"

- Variations and ambiguities in notations
  - Variations
    - Example: decimal point
      - English: 3.5
      - French: $3,5$
  - Ambiguities
    - Example: Does $\sum_{x=1}^{10} x + 1$ mean
      "$1 + 2 + 3 \ \ldots + 10 \ + 1$"
      or
      "$(1 + 1) + (2 + 1) + (3 + 1) + \cdots + (10 + 1)$"?

Recognizing the semantics of mathematical expressions is very complex

Requirements for recognizing mathematical expressions

- Every character is enclosed by a *bounding box*



- Information provided by a *bounding box*:
  1. Absolute reference point
  2. Body midline
  3. Body topline
  4. Lower left corner of the *bounding box*
  5. Upper right corner of the *bounding box*

- Why having the "body" lines?
  - To determine superscript relationship
  - Superscripts are written relative to the "body" lines
  - We understand "2" is the superscript of in:

$$b^2 \qquad p^2 \qquad a^2$$

- The order of the characters in the inputs must be in a certain order
  - To reduce the size of the problem

- Only **single-line** mathematical expressions is covered in this project:
  - No array, fractions, or table constructs allowed
  - For example, $|x| = \begin{cases} -x & \text{if } x < 0 \\ 0 & \text{if } x = 0 \\ x & \text{if } x > 0 \end{cases}$ and $\begin{bmatrix} x^2+1 & x-1 \\ x-1 & 1 \end{bmatrix}$ are obviously not single-line mathematical expressions

Structual Analysis and Presentation MathML Generation

- Remember:
  - A piece of MathML markup is a n-way tree
  - Every node of the tree is a tag (Example: `<math>`)
  - Every tag *may* contain text
  - *Offline* process is assumed for this project
  - We get the characters and their associated bounding boxes in a input file, one expression per file

- Organizing characters in mathematical expression take five steps:
  1. Generate “flat” Presentation MathML
  2. Merge digits and predefined character sequences
  3. Recognize two-dimensional relationships
  4. Replace parenthesis with `<mfenced>`
  5. Indicate implicit grouping by adding `<mrow>`

Idea:
Put all symbols and characters in an internal tree
and re-arrange the tree nodes
as new relationships are identified

An example of organizing characters in an expression

Let's try to do $\int_0^\infty \sin^{12} x \, dx$

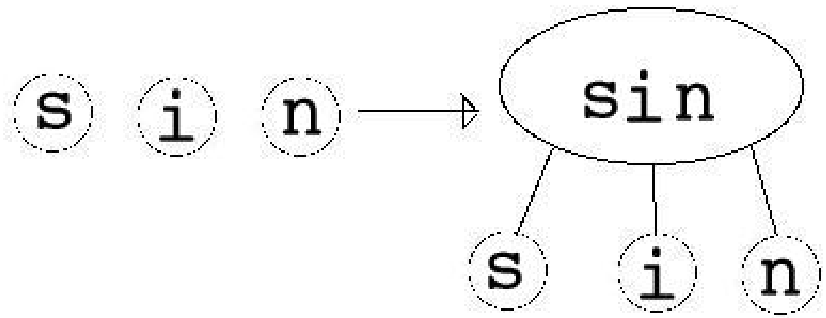
1. Generate "flat" MathML
   - Assume that all individual characters are in the same row
   - Put the characters in different categories:
     - number → `<mn>`
     - identifier → `<mi>`
     - operators → `<mo>`

- "Flat" MathML of $\int_0^\infty \sin^{12} x \; dx$:

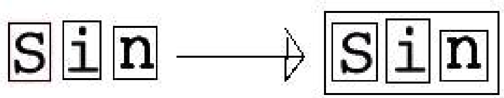
```
<math xmlns="http://www.w3.org/MathML/">
  <mrow>
    <mo> &int; </mo>
    <mi> &infin; </mi>
    <mn> 0 </mn>
    <mi> s </mi>
    <mi> i </mi>
    <mi> n </mi>
    <mn> 1 </mn>
    <mn> 2 </mn>
    <mi> x </mi>
    <mi> d </mi>
    <mi> x </mi>
  </mrow>
</math>
```

2. Merge digits and predefined character sequences

- In $\int_0^\infty \sin^{12} x \, dx$:
  - "1" and "2" can be grouped together to form "12"
  - "s", "i", and "n" together formed "sin"
  - A new node is created to store the merged characters:



  - A new *bounding box* is created upon the creation of a new node:



- Merged digits are still <mn> (a number)
- Merged character sequences are changed to <mo> (an operator)
- MathML generated so far for $\int_0^\infty \sin^{12} x$:

```
<math xmlns="http://www.w3.org/MathML/">
   <mrow>
      <mo> &int; </mo>
      <mi> &infin; </mi>
      <mn> 0 </mn>
      <mo> sin </mo>
      <mn> 12 </mn>
      <mi> x </mi>
      <mo> dx </mo>
   </mrow>
 </math>
```

3. Recognize two-dimensional relationships (Partially implemented)

- In $\int_0^\infty \sin^{12} x \, dx$:
  - "12" is the superscript of "sin"
  - Make a new node ("$<\texttt{msup}>$") to indicate the relationship
  - "12" and "sin" become the children of the new node
  - A new *bounding box* is created to surround the children
  - We do the same for all superscripts and subscripts

– Expected MathML generated so far:

```
<math xmlns="http://www.w3.org/MathML/">
   <mrow>
      <munderover>
         <mo> &int; </mo>
         <mn> 0 </mn>
         <mi> &infin; </mi>
      </munderover>
      <msup>
         <mo> sin </mo>
         <mn> 12 </mn>
      </msup>
      <mi> x </mi>
      <mo> dx </mo>
   </mrow>
</math>
```

4. Replace parenthesis with `<mfenced>`
   (Partially implemented)

   - The MathML should be rendered correctly before this step
   - Making sure that parenthesis are well nested

5. Indicate implicit groups by adding `<mrow>`
   (not implemented)

   - Just like "{" and "}" in LaTeX
   - Need to write rules to group characters and symbols together

Future works

- Better method to tolerate noise in input data
- Better approaches to determine the two-dimensional relationships between characters and symbols
  - Over and superscript
  - Under and subscript
- Insert more semantic information in MathML
  - Collect more predefined character sequences
  - Insert `<mrow>`s intelligently

Conclusion

- Recognizing and preserving semantics of mathematical expressions is not easy
- We try to reduce the complexities of the problem by:
  - defining information fetched from each *bounding boxes*
  - suggesting the steps to associate symbols and characters

Acknowledgements

- Dr. Stephen Watt
- Ms. Bethany Heinrichs, Administrative assistant of ORCCA
- Mr. Laurentiu Dragan and Mr. Igor Rodionov, System administrators of ORCCA
- ...and the members of the ORCCA lab :)

References

1. David Carlisle, Patrick Ion, Robert Miner, Nico Poppelier, Editors. Ron Ausbrooks, Stephen Buswell, David Carlisle, Stéphane Dalmas, Stan Devitt, Angel Diaz, Roger Hunter, Patrick Ion, Robert Miner, Nico Poppelier, Bruce Smith, Neil Soiffer, Robert Sutor, Stephen Watt, Principal Authors. **Mathematical Markup Language (MathML) Version 2.0 (2nd Edition)**, W3C Recommendation. Available: http://www.w3.org/TR/MathML2/, December 19, 2002

2. *OpenMath Website* Available: http://www.openmath.org

3. Stéphane Lavirotte. **Reconnaissance structurelle de formules mathématiques typographié et manuscrites**, Doctoral Thesis, École Doctoale des Sciences et Technologies de l'Information et de la Communication, Université de Nice - Sophia Antipolis, France, June, 2000. (In French)

4. Bo Wan. **An Interactive Mathematical Handwriting Recognizer for the Pocket PC**, MSc Thesis. Deparment of Computer Science, University of Western Ontario, December, 2001.