

MASARYK UNIVERSITY  
FACULTY OF INFORMATICS



# **Normative Textual Representation of Mathematical Formulae**

MASTER'S THESIS

**Maroš Kucbel**

Brno, 2013

## **Declaration**

Hereby I declare, that this paper is my original authorial work, which I have worked out by my own. All sources, references and literature used or excerpted during elaboration of this work are properly cited and listed in complete reference to the due source.

Maroš Kucbel

**Advisor:** doc. RNDr. Petr Sojka, Ph.D.

## Acknowledgement

@todo Thanks!

## **Abstract**

@todo Abstract

## **Keywords**

MathML, StAX @todo more

## Contents

1	<b>Introduction</b>	2
2	<b>Motivation</b>	3
3	<b>Mathematical Markup Language - MathML</b>	4
3.1	<i>Structure</i>	4
3.1.1	Presentation MathML	5
3.1.2	Content MathML	6
3.2	<i>Possible Ambiguities</i>	6
3.3	<i>Creating MathML Markup</i>	7
3.3.1	L <sup>A</sup> T <sub>E</sub> XML	7
3.3.2	Tralix	7
3.3.3	MaxTract	7
3.3.4	InftyReader	7
3.4	<i>Canonicalization</i>	7
3.5	<i>Status Quo of Current Tools</i>	7
3.5.1	MathPlayer	7
4	<b>Analysis and Solution Proposal</b>	8
4.1	<i>Recursive Approach</i>	8
4.2	<i>Division of Mathematical Operations</i>	8
5	<b>Converter Implementation</b>	9
5.1	<i>Java and XML</i>	9
5.1.1	StAX	9
5.1.2	Custom XML Tree Representation	9
5.2	<i>Presentation MathML</i>	9
5.3	<i>Content MathML</i>	9
5.3.1	Element apply	9
6	<b>Results</b>	10
6.1	<i>Gensim</i>	10
6.2	<i>MREC Corpus</i>	10
7	<b>Conclusion</b>	11

## **Chapter 1**

### **Introduction**

1 page Estimated length is 30-40 pages of text, which will be supplemented with samples (possibly images) and maybe some source code snippets.

## **Chapter 2**

### **Motivation**



## Chapter 3

### Mathematical Markup Language - MathML

World Wide Web pages and their main publishing language, HTML, provide many ways to present desired information to the user. However, presenting mathematics is not so easy and straightforward. There aren't any special tags in the HTML specification for including mathematics. In most cases mathematical equations are presented in the form of an image. On one hand this approach renders the equation the same way in every web browser, on the other hand there is no way to copy the equation for further use, not to mention editing the equation.

MathML was specifically created to circumvent this obstacle. It was therefore designed to be used in web pages alongside HTML. It follows that MathML is an application of XML with special set of tags used to capture the content, structure and even presentation details of mathematical equations. We will take closer look at how this is achieved in the following sections. In April 1998 MathML became the recommendation of the W3C working group for including mathematics into web pages.

#### 3.1 Structure

MathML as an application of XML consists of a tree of nodes. Each node is either empty, has textual value or has a list of descendant nodes. To provide additional information, a node can have an arbitrary amount of attributes, key – value pairs. Each MathML tree has to have a root node named `math` that belongs to the MathML namespace. Also it is important not to forget to include XML declaration and MathML doctype declaration.

MathML comes two distinct sets of tags. For the visual form of equations there are Presentation MathML tags, Content MathML tags focus on the semantics and meaning of formulas. Presentation tags are being used primarily by web browsers to display mathematical expressions to the users, Content tags are important for further processing of expressions by specialized mathematical programs.

### 3.1.1 Presentation MathML

The main focus of the Presentation MathML is displaying the equation. For this purpose there are around thirty elements, all starting with the prefix “m”. Elements are divided into two classes, first of which is called tokens. It consist of elements that represent individual content and do not contain other nested elements. These include:

- `<mi>x</mi>` - identifiers,
- `<mn>2</mn>` - numbers,
- `<mo>+</mo>` - operators, fences, separators,
- `<mtext>free text</mtext>` - text.

The content of the token can of course be expressed in more than one character (`<mo>sin</mo>`) or with an XML and HTML character entities, for example `&#62;` and `&gt;` have the same meaning as `>`, greater than. It is completely up to the user which notation he will choose.

Even though nesting other elements in tokens is not allowed, there are exceptions. For example HTML5 allows almost any HTML inline tags inside the `mtext` element. So `<mtext><b>free</b> text</mtext>` would be rendered with the bold word free. The other class of elements is layout schemata. This collection of elements is further divided into following groups:

- General Layout
  - `<mrow>` - general horizontal grouping,
  - `<mfrac>` - fractions and binomial numbers,
  - `<msqrt>`, `<mroot>` - radicals
- Script and Limit - superscripts, subscripts,
- Tabular Math - tables and matrices,
- Elementary Math - notation for lower grades mathematics.

We can thought of the layout schemata as a form of expression constructors, that specify the way in which sub-expression are constructed into larger expressions, starting with tokens and ending with the `math` element. Therefore elements belonging to layout schemata class do not contain any characters, only other nested elements (layout schemata or tokens).

@todo image

MathML also provides over fifty attributes for fine tuning of expressions, like setting colors, dimensions, alignments and many others.

### 3.1.2 Content MathML

Mathematical notation, rigorous as it is, is still not standardized around the world and there are many different ways of writing mathematical expressions based on cultural customs. Even simple multiplication operation can be written as  $x*y$ ,  $xy$ ,  $x$  times  $y$ ,  $x \times y$ . In many situations there's no need to actually render the expression, only the underlying meaning is important. For this reason Content MathML provides a framework and a markup language that capture the semantics of mathematical expressions.

The structure of Content MathML tree is based on a very simple principle – applying an operator to sub-expressions. For example the quotient  $x/y$  can be thought of as applying the division operator to two arguments  $x$  and  $y$ . It follows that the cornerstone of Content MathML is the function application element `<apply>`. The first child of the `<apply>` element signifies the operator, which can be an `<apply>` element again, and the rest of the `<apply>` element direct descendants are arguments to which the operator is applied. Token elements `<ci>` and `<cn>` are available to represent numbers and variables respectively.

Content MathML provides two ways of defining the operator. The first one is a set of more than 100 elements, each corresponding to some mathematical operator. For example for the addition operator there is `<plus>`, for logical xor `<xor>` and so on. Then there is the second way, the element `<csymbol>`. This element contains a textual representation of the operator, that is bound to a definition in a content dictionary referenced by either the attribute `cd` or `definitionURL`. External content dictionaries are important for communication between agents and there exist several public repositories of mathematical definitions. Most notable is the OpenMath Society repository.

In MathML 3, a subset of Content MathML elements is defined: Strict Content MathML. This uses only minimal, but sufficient, amount of elements, most importantly allows only the usage of `<csymbol>` element for defining operators. Strict Content MathML is designed to be compatible with OpenMath standard.

## 3.2 Possible Ambiguities

1-2 pages

### 3.3 Creating MathML Markup

Creating MathML documents is very simple. All that is needed is a word processor available on every operating system. However, this approach is prone to errors, be it the wrong letter case or unclosed tags. Since MathML is an XML language, the usage of some sophisticated word processor that supports tags highlighting, code completion and XML validation will greatly improve the efficiency of creating MathML documents. But the whole process is still very time consuming and impractical. As is the case with most XML documents, even writing simple mathematical expressions takes up a lot of space and time. Fortunately there are many dedicated MathML editors, that provide some degree of abstraction from the actual MathML markup.

Another way of creating MathML documents is a conversion from different formats. Among scientist, mathematicians particularly,  $\text{\TeX}$  is the format of choice. It then comes as no surprise that there are many sources of mathematical texts written in  $\text{\TeX}$ . However, not every publication comes with the source files. Often only print-ready PDF files are released. Also many academic writings, especially older ones, are available only in printed form. These have to be scanned using an Optical Character Recognition (OCR) software.

#### 3.3.1 $\text{\LaTeX}$ XML

#### 3.3.2 Tralix

#### 3.3.3 MaxTract

#### 3.3.4 InftyReader

### 3.4 Canonicalization

1-2 pages

### 3.5 Status Quo of Current Tools

2 pages total

#### 3.5.1 MathPlayer

## **Chapter 4**

### **Analysis and Solution Proposal**

#### **4.1 Recursive Approach**

#### **4.2 Division of Mathematical Operations**

## **Chapter 5**

### **Converter Implementation**

#### **5.1 Java and XML**

3-4 pages Options of working with XML in Java, selecting the most suitable and why

##### **5.1.1 StAX**

1-2 pages More details about StAX

##### **5.1.2 Custom XML Tree Representation**

1-2 pages Creating custom tree with similar structure as DOM but less additional info and therefore lower space and time requirements

#### **5.2 Presentation MathML**

4-5 pages Specifics of implementing converter for presentation MathML

#### **5.3 Content MathML**

4-5 pages Specifics of implementing converter for content MathML, possibly longer

##### **5.3.1 Element apply**

## **Chapter 6**

### **Results**

#### **6.1 Gensim**

#### **6.2 MREC Corpus**

## **Chapter 7**

### **Conclusion**

1 page