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

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 `>` 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>free text</mtext> would be rendered with the bold word free. The other class of elments 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 x 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 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, TEX is the format of choice. It then comes as no surprise that there are many sources of mathematical texts written in 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.

3.2.1 LATEXML

The lack of a suitable tool for converting LATEX to XML prompted the participants of the Digital Library of Mathematical Functions project to develop their own solution. Main goals of LATEXML design contain the aspiration to faithfully emulate TEX behavior, the ease of extensibility and the preservation of both semantic and presentation aspects of original documents. To this end LATEXML provides two main commands, `latexml` and `latexmlpost`. `latexml` converts the initial TEX document to XML based on a set of LATEXML-bindings files, that define the mapping of TEX macros to XML.`latexmlpost` then processes resulting XML output by converting mathematics and graphics, cross-referencing and applying an `XSLT` stylesheet.

Both commands come with a multitude of parameters that allow users to customize the whole process, such as loading user-defined bindings or choosing the output format. Based on the requested output format, mathematics is converted to graphics (png images for HTML) or Presentation MathML in case of XHTML or HTML5.

LATEXML is freely available online as an installation package for Linux systems as well as Windows and MacOS. An extensive and detailed manual is available online or as a PDF document.

3.2.2 Tralics

Tralics is a freeware software designed to translate LaTeXsources into XML documents, that can be further converted into either PDF or HTML. The generated XML is conforming to the local ad-hoc DTD (a simplification of the TEI) with mathematical formulas conforming to the Presentation MathML 2.0 recommendations.

Similarly to LaTeXML, Tralics provides many ways for customization of resulting documents, among them the possibility to change element and attribute names in the XML file. Besides Presentation MathML, mathematical formulae can be translated to LaTeX-like elements as well.

Tralics is readily available online in the form of source files or binaries for Linux, Windows and MacOS operating systems. An extensive documentation regarding customization and usage is also available online.

3.2.3 MaxTract

MaxTract is a tool that through spacial analysis of symbols and fonts in PDF document reverse-engineers its source files in the form of LaTeX or XHTML + MathML documents. The conversion process requires valid PDF files to work correctly as it needs the information about symbols, font encoding and width of objects contained in the PDF file. PDF documents created via LaTeX fulfill these requirements, therefore MaxTract is able to process most of the scientific and mathematical material.

3.2.4 INFTY Reader

INFTY is an Optical Character Recognition (OCR) software especially created for mathematical documents. INFTY reads scanned pages (images) and yields their character recognition results in multiple formats, including LaTeX and MathML. It does so in four steps: layout analysis, character recognition, structure analysis of mathematical expressions, and manual error correction (optional).

The most important phase, character recognition, is responsible for distinguishing mathematical expressions and running a character recognition engine originally developed for mathematical symbols together with non-specialized commercial OCR engine.

INFTY Reader is available free of charge for limited amount of time, then it is necessary to purchase a license.

3.3 Possible Ambiguities

1-2 pages

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 impementing converter for presentation MathML

5.3 Content MathML

4-5 pages Specifics of impementing 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