Development of Information-Retrieval Tool for MathML-based Math Expressions

Yoshinori MIYAZAKI[a], **Yoshihide IGUCHI**[b]
[a]*Faculty of Informatics, Shizuoka University, Japan*
[b]*Graduate School of Informatics, Shizuoka University, Japan*
yoshi@inf.shizuoka.ac.jp

Abstract: In this research, it's our objective to develop an information-retrieval tool for mathematical expressions to enhance usability of contents including math. Our target is MathML (Mathematical Markup Language) -based expressions, which was developed for describing math on the web.

Keywords: MathML, Information Retrieval, Regular Expression, Fuzzy Search

Introduction

Search engines, or information-retrieval tools, have made it possible to search information you request from enormous amount of web contents in the world. Google and Yahoo are good examples. They, however, cannot search math expressions as of now. They are used not only in math itself but various fields such as physics, economics, and even in education from elementary levels. The contents on the web are also the case, so is in educational systems such as e-learning systems.

In spite of its universality, it's difficult to implement math-finding functions to conventional search engine systems because the way math expressions are described is different from those of text data, and also because they can mean different depending on the fields in which they are used.

Our objective is to enable information retrieval with the inclusion of arbitrary math expressions stated by MathML, and also to pursue convenience toward web contents treating math expressions. Similar research has been conducted by [5] and [6]. Taking these two pieces of literature, the authors don’t find them completely satisfactory in that the former does not fully consider math structures by searching only frequency of MathML tags, and the latter is indexing tag information too rigidly to realize fuzzy searching (no flexibility). Our research is expected to conquer these problems as well.

1. MathML

1.1 What is MathML

Math expressions have unique forms, combined by alphanumerics and symbols in a two-dimensional plane. For example, fractions allocate two numbers (namely, a denominator and numerator) in two dimensions, or above and below a line for separation. HTML, used most generally in describing web contents, however, does not have such formats. As a compromise, it has been a common practice to substitute by plain texts (in case of fractions, for instance, "(numerator)/(denominator)") or to paste math expressions converted into image files. They are of course quite restrictive in terms of their layouts and secondary use.

On the other hand, MathML has recently been gaining attention, which was generated for the purpose of encoding math expressions on the web. The markup language was recommended by W3C (World Wide Web Consortium) [Version 1.01 in June 1999, and version 2.0 in October 2003]. Currently MathML 2.0 has been used widely. Its corresponding working group also has been established for MathML 3.0 on June 28th, 2006[2]. The new version is expected to improve and expand MathML in the areas of internationalization, accessibility, and mathematical richness.

There are two categories in MathML, or, Presentation MathML and Content MathML. The former deals with the visual presentation while the latter even deals with their structures and meanings.

1.2 MathML Versus LaTeX

LaTeX is probably counted as another widely known markup language describing math expressions. The following codes show a set of examples by LaTeX, Presentation MathML, and Content MathML for depicting the following 2×2 matrix:

$$\begin{bmatrix} x & 3 \\ -1 & y \end{bmatrix}$$

- An example by LaTeX

```
\begin{eqnarray}
\left[
\begin{array}{cc}
x & 3 \\
-1 & y \\
\end{array}
\right]
\end{eqnarray}
```

- An example by Presentation MathML

```
<math xmlns="http://www.w3.org/1998/Math/MathML">
<mo>[</mo>
<mrow><mtable align="right" width="80%">
<mtr><mtd><mrow><mrow><mo>[</mo>
<mtable><mtr><mtd columnalign="center">
<mrow><mi>x</mi></mrow></mtd>
<mtd columnalign="center"><mrow><mn>3</mn></mrow></mtd>
</mtr><mtr><mtd columnalign="center">
<mrow><mo>-</mo><mn>1</mn></mrow></mtd>
<mtd columnalign="center"><mrow><mi>y</mi></mrow></mtd>
</mtr></mtable><mo>]</mo></mrow></mrow>
</mtd></mtr></mtable></mrow>
<mo>]</mo></math>
```

- An example by Content MathML

```
<math xmlns="http://www.w3.org/1998/Math/MathML">
```

```
<mrow>
<matrix><matrixrow><ci>x</ci><cn>3</cn></matrixrow>
<matrixrow><apply><minus/><cn>1</cn></apply>
<ci>y</ci></matrixrow></matrix>
</mrow></math>
```

LaTeX has an advantage over MathML in terms of (shorter) length of codes, this is because LaTeX is focusing only on making documents with nice-looking layouts. MathML are superior to LaTeX from the point of view that additional semantic information can be added (especially Content MathML).

MathML has already been implemented on a certain number of browsers, such as Mozilla, Netscape, Amaya, and the like. Several LMS (Learning Management System) have incorporated MathML technology as a measures of handling math expressions (such as BlackBoard, eCollege, Maple TA)[3]. For instance, BlackBoard, adopting WebEQ[4], has made it possible to create math contents taking advantage of GUI.

The table below briefly shows several tags of Presentation MathML.

Table 1: MathML (Presentation MathML)

Tag	Meaning	Math Sample	Corresponding sample code
mn	Numeric	0	<mn>0</mn>
mi	String	a	<mi>a</mi>
mo	Symbol	$+$	<mo>+</mo>
msup	Power	x^2	<msup><mrow><mi>x</mi></mrow> <mrow><mn>2</mn></mrow></msup>
mrow	Group together any number of sub-expressions		
mtable mtr mtd	Make tables	$\begin{matrix} x & 3 \\ -1 & y \end{matrix}$	<mtable><mtr><mtd><mi>x</mi></mtd> <mtd><mn>3</mn></mtd></mtr> <mtr><mtd><mrow><mo>-</mo> <mn>1</mn></mrow></mtd> <mtd><mi>y</mi></mtd></mtr></mtable>

Note that mrow tag is not necessarily required in the case of x^2 in the table (not vital if only to show on the web). For the speed tuneup, it is considered to ignore these tags when search command is executed. One problem of doing so, however, is that different math expressions can be treated likewise (for example, $\int_{ab}^{c} x^2 dx$ and $\int_{a}^{bc} x^2 dx$ are considered the same expressions). After all, they have trade-offs (between high speed and high precision).

2. Development of Information Retrieval System

In respect of information retrieval, adopting Content MathML seems a better strategy. This is because Content MathML has tags with mathematical meanings, which makes information retrieval easier and full of varieties. Regardless of this, the reality is that BrEdiMa[7], MathBlackBoard[8], and others as well as WebEQ mentioned before have all adopted Presentation MathML. Furthermore, so has the converter from LaTeX source codes to MathML codes, which the authors used in this research. For this reason, it is thought to be more realistic to develop the system (search engine) intended for contents described by Presentation MathML. Of course, it is natural to shift the system to Content MathML-intended when it gains more popurality on the web in the future.

This system has been created as a web application, assuming the use on the web.

2.1 Environment for Development

The system has been implemented by Java (1.6.0_01) and MySQL (4.1.20).
And its platform environment is Fedora 8 server machine with Apache, Tomcat (5.5.12).

2.2 Targeted Math Contents

For the target math contents on the web, IMED Linear Algebra[9] has been adopted this time. This site was developed as an international project between UCLA and Japanese Universities team, and the first author was one of the project members.

It has to be admitted that up to the present, MathML contents on the web is not yet widely spread. Neither is the way IMED Linear Algebra was developed (this site is written in HTML originally created by LaTeX). Try out TtM[10] and WebEQ[4] were chosen as converters from LaTeX to MathML.

2.3 Storage to DataBase

Converted MathML contents are stored in DB (DataBase) beforehand. This process is progressed with lexical analysis, designed by the authors (let the system be called "Data Processing System"). There are two types of DBs, one of which is for information retrieval, created by extracting and modifying the original math expressions, and another of which is for storing original contents, for output results.

The structures of DB for storage and retrieval are shown in Table 2 and Table 3, respectively.

Table 2: DB for storage (IMED Linear Algebra)

Chapter (c)	Paragraph (p)	File Name (file_name)	Title (title)	HTML Data (data)
1	1	math1-1.xhtml	The Object …	<html>…</html>
1	2	math1-2.xhtml	Matrices	…
…	…	…	…	…
2	13	math2-1.xhtml	Catalogue of …	…
…	…	…	…	…

p	: int, not null unique
c	: int, not null
file_name	: varchar(255), not null
title	: varchar(255), not null
data	: longblog, not null

Table 3: DB for retrieval (IMED Linear Algebra + page numbers)

Id (id)	MathML Data (data)
1	<math>…</math>
…	…

id	: int, not null unique
data	: text, not null

The Data Processing System is made up by two modules at present. The first one is an interface module, and the second is a processing module for lexical analysis and storing in DB (see Figure 1). JDBC API has been used for the access from Java to MySQL.

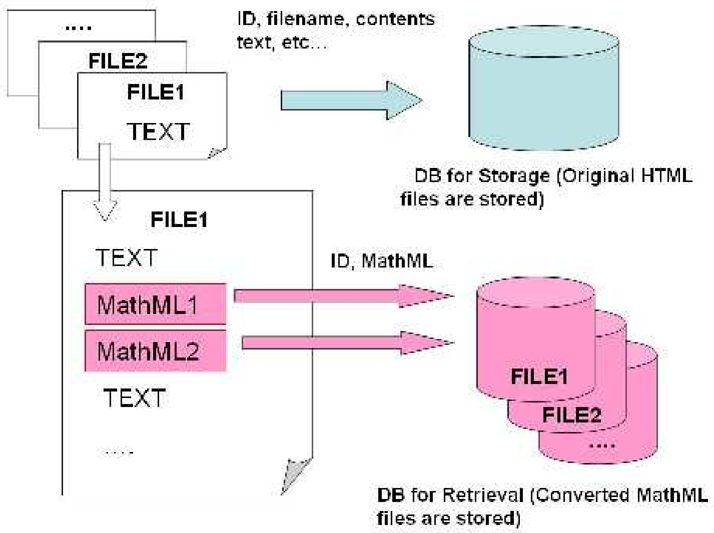
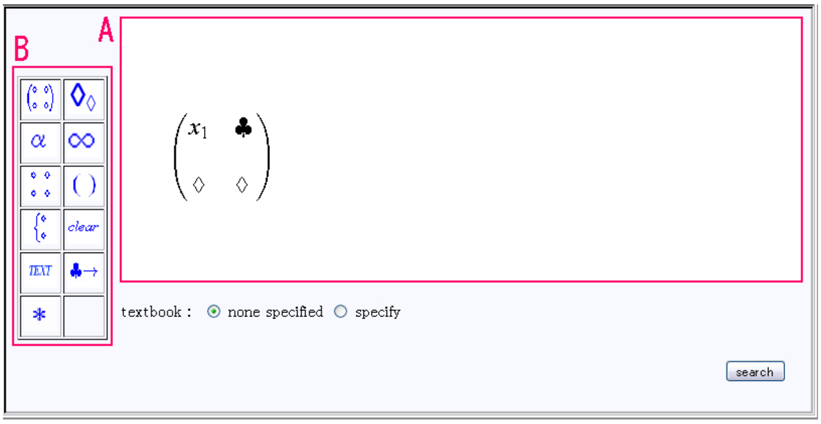


Figure 1: How contents are stored in DBs

2.4 Interface

One of the sales points of this system is that the authors made a GUI (Graphical User Interface) which is easy to use math expressions without knowledge of markup language. A palette is placed on the screen with buttons for various functions. Examples are the buttons for special symbols specific to math, Greek letters, editing, and the like. Figure 2 is a sample screenshot of the interface being used. Spot A is for inputting math expressions for retrieval, and Spot B is the palette mentioned above.



textbook : ◉ none specified ○ specify

Figure 2: Interface of the retrieval system

2.4.1 Math Canvas

More concretely, Spot A is an example of inputting 2×2 matrix with (1,1) element (x_1) already inserted. Club-suit symbol is representing a prompt symbol, and diamond-suit symbol implies the elements supposed (but not yet) to be input.

2.4.2 Input Palette

Spot B is for editing math expressions. Its functions are, roughly speaking, divided into 4 kinds:

- describing math-specific expressions

- inputting alphanumerics using textboxes
- inputting math symbols and Greek letters
- moving prompt and resetting all the input.

Examples of math-specific expressions are fractions, matrices, powers, etc. These (two-dimensional) input are realized by GUI buttons. After clicking a corresponding button in Spot B of Figure 2, a pop-up window comes up (Figure 3) and enables inputs. Spot A is for various types of math expressions. A module for matrices has been developed separately, since the entries of matrices can be varied depending on its size (number of rows and columns) (Spot B of Figure 3). Likewise, simultaneous equations are separated as well (Spot C of Figure 3). Others than them are all stipulated in another module.

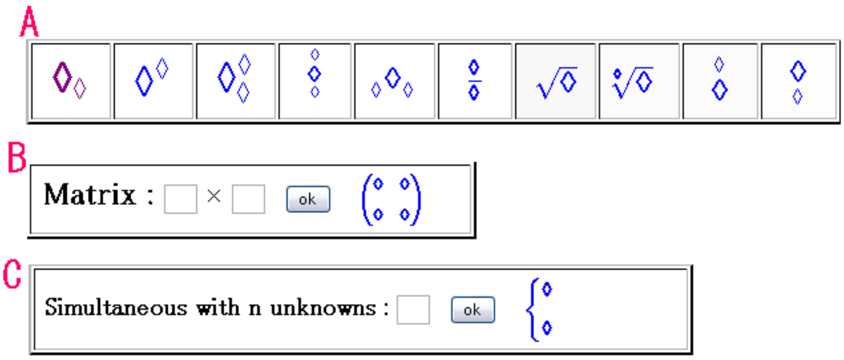


Figure 3: Pop-up window of retrieval system (1)

Next is for inserting alphanumerics with textboxes. More exactly, using the textboxes, operators and identifiers are expected to be input (Figure 4). In MathML, even numbers (e.g. "0") are enclosed by tags, (e.g. "<mn>0<mn>"). Therefore, they have to be output with addition of tags for numerics, operators, or identifiers through a simple lexical program.

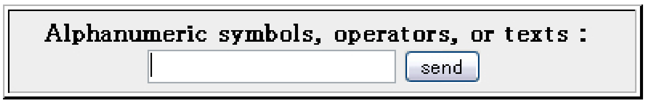


Figure 4: Pop-up window of retrieval system (2)

The last one is about the special (math) symbols and Greek letters. Clicking its corresponding button pops up a window for the palette. Figure 5-A is for math symbols (such as ∞ and ±), and Figure 5-B for Greek letters.

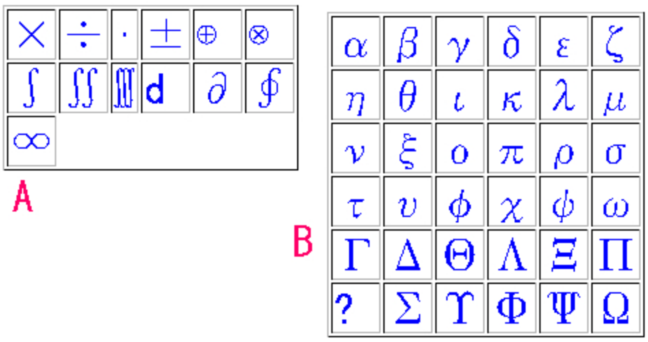


Figure 5: Pop-up window of retrieval system (3)

2.5 *Processing Retrieval*

In processing retrieval, data matching is processed using regular expressions in MySQL, the DBMS adopted in this research. The following table (Table 4) is the list of regular expressions for retrieval.

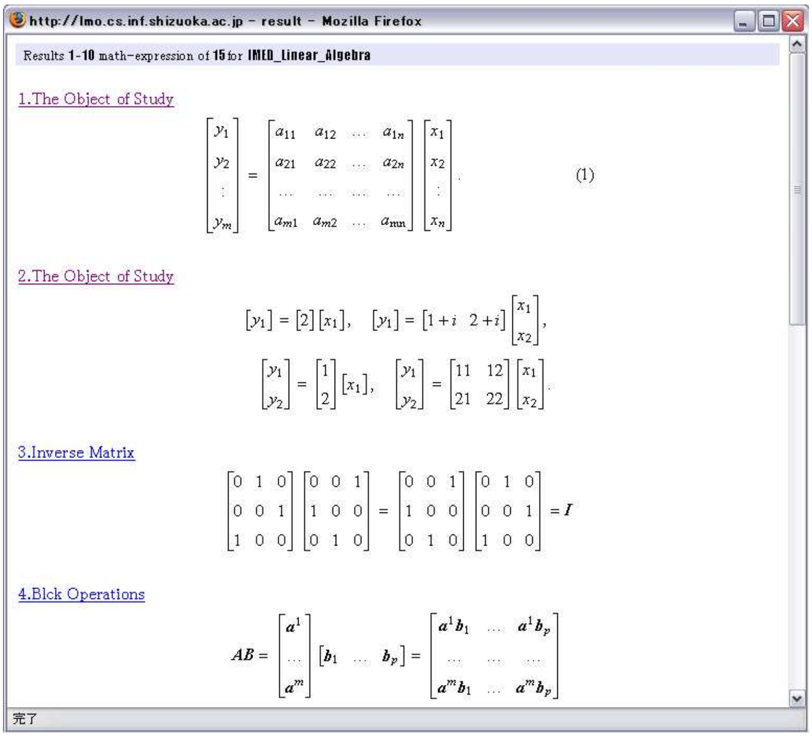
By using regular expressions, fuzzy search has been made possible. For example, the users can combine multiple number of wildcards, and make information retrieval for “A^2+B^2”-formed math expressions (A and B are both wildcards). This ambiguity is quite useful since no one knows what variables are to be used in each math text introducing Pythagorean theorem or related topics (such as length of two points).

Table 4: Regular expressions of MySQL used for retrieval

Expression	Meaning
.	Arbitrary character
?	Repetition of previous character 0 or 1 time(s)
(abc|def)	Either “abc” or “def”
[^a]	Characters other than “a”
*	Repetition of previous character more than or equal to 0 times

3. Experiment and Discussion

After completing the system development, several retrievals were executed for verifying its performance. Figure 6 is a search result of inquiring math expressions having “=(matrix)(matrix)” form, namely, the multiplication of matrices on the right-hand side. This form frequently appears in linear algebra especially for the purpose of decomposing matrices. As in the figure, the title, the link, and retrieved math expressions are displayed all together.



http://lmo.cs.inf.shizuoka.ac.jp - result - Mozilla Firefox

Results **1-10** math-expression of **15** for **IMED_Linear_Algebra**

1.The Object of Study

$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} . \qquad (1)$$

2.The Object of Study

$$[y_1] = [2][x_1], \quad [y_1] = [1+i \;\; 2+i] \begin{bmatrix} x_1 \\ x_2 \end{bmatrix},$$

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \end{bmatrix} [x_1], \quad \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} 11 & 12 \\ 21 & 22 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} .$$

3.Inverse Matrix

$$\begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix} = I$$

4.Blck Operations

$$AB = \begin{bmatrix} \boldsymbol{a}^1 \\ \dots \\ \boldsymbol{a}^m \end{bmatrix} [\boldsymbol{b}_1 \;\; \dots \;\; \boldsymbol{b}_p] = \begin{bmatrix} \boldsymbol{a}^1\boldsymbol{b}_1 & \dots & \boldsymbol{a}^1\boldsymbol{b}_p \\ \dots & \dots & \dots \\ \boldsymbol{a}^m\boldsymbol{b}_1 & \dots & \boldsymbol{a}^m\boldsymbol{b}_p \end{bmatrix}$$

完了

Figure 6: Example of retrieved result

Since this tool allows fuzzy search, by using ‘*’ for arbitrary form, flexible search is realized. For instance, if the mathematical expression

$$\begin{bmatrix} a_{11} & * & * \\ * & * & * \\ * & * & a_{33} \end{bmatrix}$$

is input, this tool retrieves 3×3 matrices whose (1,1)-entry is a_{11} and (3,3)-entry is a_{33}.

For exact matching, only the first chapter of [9] was dealt with this time, since the number of the math expressions is quite large in total. Table 5 is the result of the experiment, with the number of math expressions and the number of hits by the tool. Further experiments in depth are of course necessary.

Table 5: Result of exact matching

Sec # of Chap 1 →	1	2	3	4	5	6	7	8	9	10	11
(1) # of hits	10	22	8	17	20	24	5	10	8	61	34
(2) # of expressions	12	22	8	20	20	29	5	10	12	61	38
(1) / (2) * 100 (%)	83	100	100	85	100	83	100	100	67	100	89

As in Table 5, some exceptions were observed that math expressions could not be found as expected. After the investigation, we found out that this was because of the converted MathML files not formed correctly. It seems that sometimes in Presentation MathML, codes can be made even illogically (since more emphasis is put on displaying math, rather than structuring logical data). And that was how the converter did this time. Another exception was that some math expressions including '[' or '(', both of which can be used as meta-characters in regular expressions, could not be found.

4. Concluding Remarks and Future Plan

Information retrieval system was developed intended for math contents described by (Presentation) MathML. The use of regular expressions made it possible to search with a wider variety (fuzzy search).

The future plan goes as follows:

- Speed-ups by indexing DB contents
- Compacting stored data applying DB normal forms
- Fixation of imperfect results in Section 3 (such as problems of meta characters)
- More flexible search taking advantage of JLink of Mathematica (for example, search for "1+x" is interpreted as "x+1" as well)

References

[1] Y. Miyazaki, and Y. Iguchi, Development of Search Engine for MathML-based Mathematical Descriptions, *Information Technology Letters*, pp. 339-342 (2007).

[2] W3C:Mathematical Markup Language(MathML) Version 2.0 (Second Edition), (2003).

[3] Hans Cuypers, Karin Poels, Rikko Verrijzer, Olga Caprotti, and Jouni Karhima, State of the Art in Mathmatical, *E-learning D1.1*, pp. 9-11 (2005).

[4] WebEQ : Edit Live! WebEQ Euqtion Editor, http://www.ephox.com/products/equationeditor/

[5] T. Nakanishi, S. Kishimoto, M. Murakata, T. Otsuka, T. Sakurai, and T. Kitagawa, An Implementation Method of Composite Association Retrieval System for Data of Mathematical Formulas, *DBSJ Letters*, Vol. 4, No. 1, pp. 29-32 (2005).

[6] H. Hashimoto, Y. Hijikata, and S. Nishida, A Survey of index formats for the search of MathML objects, *IPSJ SIG Notes* 2007-DBS-142-(10), Vol.2007, No.54(20070531), pp. 55-59 (2007).

[7] Y. Nakano, and Y. Murao, Math Input System on the Web, *Risa/Asir Conference 2006* (2006).

[8] H. Deguchi, MathBlackBoard, *J.JSSAC*, Vol.11, No.3,4, pp.77-88 (2005).

[9] IMED : IMED Linear Algebra, http://128.97.76.119/linearalgebra/

[10] TTM : Try out TtM, http://hutchinson.belmont.ma.us/tth/mml/ttmmozform.html