An Investigation of Index Formats for the Search of MathML Objects

Yoshinori Hijikata, Hideki Hashimoto and Shogo Nishida
Graduate School of Engineering Science, Osaka University
1-3 Machikaneyama, Toyonaka Osaka 560-8531, Japan
{hijikata,hashimoto,nishida}@sys.es.osaka-u.ac.jp

Abstract

Users cannot search information by mathematical formulas as queries in existing search engines. This is because mathematical formulas are not expressed as a sequence of characters. Some formulas are expressed in a complex structure like fractional numbers and index numbers. We present a search engine for MathML objects using the structure of mathematical formulas. The system makes the inverted indices by using the DOM structure of the MathML object. The indices are constructed from some paths of the DOM structure and are expressed in XPath. This paper describes the experiment conducted to study the effectiveness of those indices by using the mathematical contents which are open to the public on the Internet.

1. Introduction

Mathematical formulas are one of the best expressions to tell people the knowledge in various fields like science, engineering, agriculture, economics, and so on. However, people cannot search web documents by a mathematical formula as a search query in current search engines on the Web. The current search engines allow us to use a sequence of characters for a search query. Mathematical formulas are not necessarily expressed in a sequence of characters. They are sometimes expressed in a structure like fractional numbers and index numbers. For an example, xy and x^y have the same expressions if they are expressed in a sequence of characters. Both of them include x and y and they are arranged in the same order. However the meaning of xy and the meaning of x^y are totally different. Therefore people cannot search mathematical formulas by using a search engine which only uses the kind of characters and the order of them. If we want to search mathematical formulas, the system must consider their mathematical structures.

MathML (Mathematical Markup Language) [1] is being introduced worldwide as a standard to express mathematical formulas on the Web. People can express the structure of the mathematical formula by exploiting the hierarchy structure of XML. Nakanishi et al. have proposed a search method of mathematical formulas for MathML objects [2]. Its search type is the similarity search. It uses the vector space model, one of the popular search methods in the field of information retrieval. However the vector space model only has the information about the existence of characters and mathematical symbols. It does not express the structure of mathematical formulas. If two different formulas include similar characters and symbols, they are listed in the same search result. Furthermore, the search performance (speed) of the vector space model drastically becomes slow if the number of objects (documents) becomes large. The inverted file (or inverted index) is the better search method especially for commercial search engines. As adopted in major search engines like Google and Yahoo!, the inverted file provides a fast search performance.

In our research, we propose a search engine of mathematical formulas. We make an inverted index by using the DOM (one of the API to access the XML document [3]) structure of MathML of objects. We conduct an investigation of the discrimination capability according to the the type of index. Then we propose an indexing method of MathML objects. Finally, we conduct an experiment to evaluate the search results by using the proposed indices.

2. Overview of the Math Search Engine

Figure 1 shows the system structure of the proposed search engine of mathematical formulas. The system consists of two sub systems. One of them crawls the Web and collects web documents including MathML objects, and then it stores them with the search indices in an inverted file. The other system sends a query to the inverted file and outputs the search results when it receives the query from a user.

Web documents which this search engine targets are HTML documents or XML documents which include mathematical formulas expressed in MathML like Figure 2-(a). <msqrt> represents the root, <msup> represents the

0-7695-3028-1/07 $25.00 © 2007 IEEE
DOI 10.1109/WI-IATW.2007.121

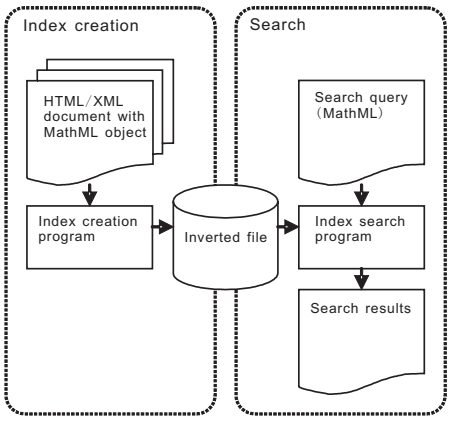
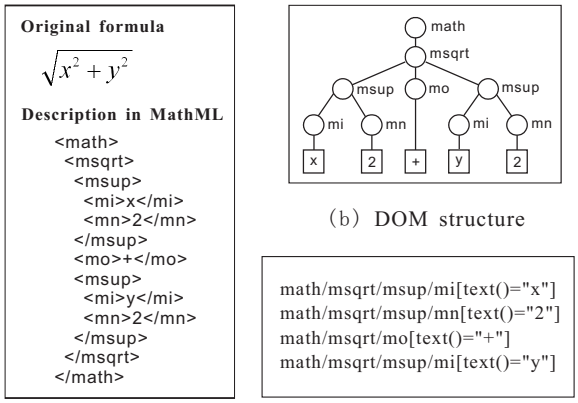


Figure 1. Overview of our proposed system.



(a) MathML description (b) DOM structure (c) XPath expression

Figure 2. An example of MathML description, dom and XPath expression.

square, <mi> represents the variable number, <mn> represents the numerical number, <mo> represents the mathematical operator.

When the system obtains a new web document, it retrieves each MathML object from the document. Then it acquires the DOM structure of the MathML object (see Figure 2-(b)). It generates indices by expressing every path from the root node to the leaf node and the value of the leaf node in XPath [4] expression (see Figure 2-(c)). It stores the ID of the document with these indices as keys. In this step, the system does not include the following tags in the indices because they are not relating to the content of the formula: <mrow>, <mstyle>, <semantics>, <annotation>. Note that <mrow> represents the horizontal location of the formula at the time of rendering, <mstyle> represents the rendering attribute like color and font, <semantics> and <annotation> represents the additional semantic annotation to the formula. Table 1 shows an example of inverted indices. The first row shows that the document 1, 52, 70, 271, ... include the MathML object which has a fraction including the symbol "δ."

Table 1. Examples of inverted index.

XPath description	List of document ID
/math/mfrac/mo[text()="\delta"]	1,52,70,271,,,
/math/msqrt/mfrac/mn[text()="1"]	2,16,55,102,,,
/math/mfrac/mfrac/mi[text()="\pi"]	22,93,181,,,

When the user wants to search web documents including a specific formula, he/she inputs the search query in a MathML object by using an editor of mathematical formula. Or when the user browses some web pages and finds a formula which he/she wants to study deeply, the MathML object is automatically extracted from the web document by the system by pointing the mathematical formula with the mouse pointer. Because the meaning of the mathematical formulars are not expressed in some specific keywords and are expressed in the mathematical structure, we need to carefully study the user interface for inputting the query. In this research, we will develop this editor and tool in the future (In this paper, we focus on only the index format for the search engine). Currently, we use MathType (software for inputting mathematical formulas) for inputting search queries. When we conduct a 'copy' operation in MathType, we can store the copy in MathML. We can input the MathML string in a HTML form by pasting it in the form.

The system acquires the DOM structure of the given MathML object of the search query and acquires a key which is expressed as a path from the root node to one leaf node. Then it acquires the document IDs whose indices match the key from the inverted file. Finally it checks the paths from the root node to the all leaf nodes between the MathML objects of the search query and the MathML objects in the acquired web document. It only outputs the web document in the search results which has a MathML object whose all paths match to all the paths of the MathML object of the search query. Here, we assume that the system stores one path of the MathML object in the web document in the inverted file and uses one path of the MathML object of the search query as a key. This is the simplest idea. In this research, we investigate what kinds of indices are useful from the viewpoint of memory cost and search performance.

3 Investigation of Indices

For finding the appropriate type of indices, we investigate 87000 MathML objects which are open to the public in a web site of mathematical formula collection which is

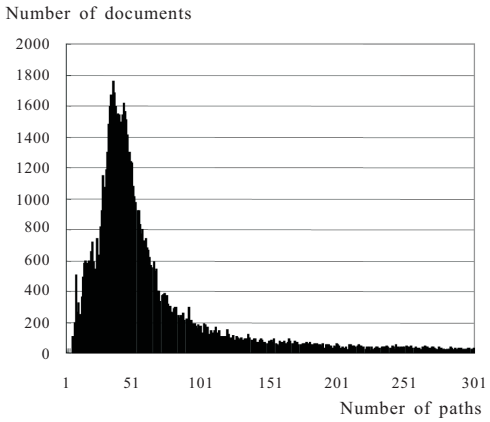


Figure 3. The number of paths in the DOM tree of MathML object.

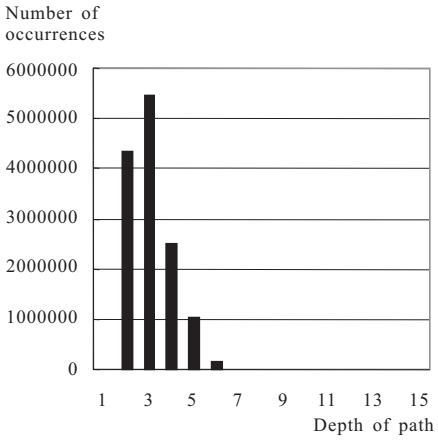


Figure 4. The number of occurrences for each depth of the DOM tree.

operated by WOLFRAM RESEARCH [5].

Figure 3 shows the number of MathML objects (actually documents) according to the number of paths of every MathML object. Figure 4 shows the number of paths in the 87000 MathML objects according to its depth. From Figure 3, the number of MathML objects is large in the range from 30 to 60 in the number of paths. The number of MathML objects is small in the range of 100 and bigger. From Figure 4, the deepest path is 6. We can see that MathML objects do not have deep hierarchical structure. Only one path is not enough for indexing the MathML object. We have to create an index by combining several paths.

We extracted all the paths from all the MathML objects and expressed them in XPath. We counted the number of occurrences of all the path patterns from the extracted paths.

Table 2. Ranking of XPath expressions in frequency (top 10).

Rank	XPath expression	Number (Ratio) of occurrences
1	math/mo[text()="⁢"]	857574(6.369%)
2	math/mo[text()="("]	540280(4.012%)
3	math/mo[text()=")"]	540280(4.012%)
4	math/mfrac/mo[text()="⁢"]	540280(4.012%)
5	math/msup/mo[text()="-"]	381608(2.834%)
6	math/msup/mo[text()="⁢"]	341416(2.535%)
7	math/mo[text()="-"]	272379(2.023%)
8	math/mo[text()="+"]	222359(1.651%)
9	math/mfrac/mn[text()="2"]	221889(1.648%)
10	math/msup/mo[text()=")"]	202749(1.506%)

Table 3. Ranking of XPath expressions in frequency (from 1001 to 1010).

Rank	XPath expression	Number (Ratio) of occurrences
1	math/msup/msup/mi[text()="sin"]	320(0.00238%)
2	math/msqrt/msup/mo[text()="-"]	320(0.00238%)
3	math/msub/mi[text()="ℬ𝒫"]	319(0.00237%)
4	math/mo[text()="\in"]	318(0.00236%)
5	math/mfrac/msup/mi[text()="sn"]	315(0.00234%)
6	math/mfrac/msub/msub/mi[text()="b"]	312(0.00232%)
7	math/munderover/mo[text()="\Sigma"]	311(0.00231%)
8	math/msup/mfrac/msqrt/mn[text()="1"]	311(0.00231%)
9	math/mfrac/msup/mi[text()="\beta"]	310(0.00230%)
10	math/mtable/mtr/mtd/mi[text()="\mu"]	309(0.00229%)

We obtained 8813 path patterns. Table 2 shows the top ten list in the number of occurrences and the XPath expression. Table 3 shows the list of the rank from 1001 to 1010. Figure 5 shows the graph whose axes are the number of occurrences of the path pattern and its ranking.

For example, the path pattern ranking at 1001 occurs 320 times. If we use this path pattern as an index, we can narrow down to 320 MathML objects. By comparing Table 2 and Table 3, the path patterns ranking below 1001 have comparatively deeper hierarchical structure and include mathematical symbol or operator representing the meaning of the original formula. The path patterns ranking in top 10 list have shallow hierarchical structure and only include common mathematical symbols or operators like "&InvisibleTime" (product with omitted product operator '·'), parenthesis and difference operators. From Figure 5, we can see that the number of occurrences is particularly high for patterns ranking under top 40.

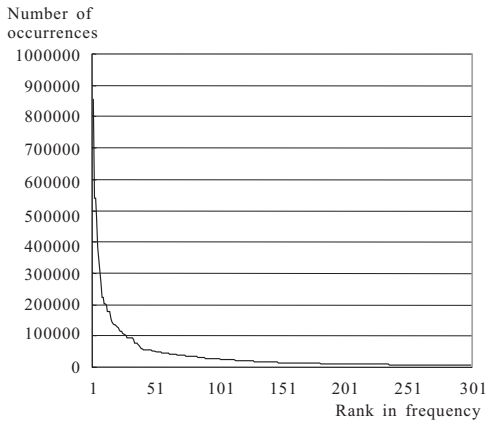


Figure 5. The number of occurrences of the path pattern for each rank in frequency.

We compare this result with previous findings in the area of the general text retrieval. It is reported in that area that the most frequent keyword only covers 0.01% in all the keywords [6]. The most frequent path pattern in this investigation covers 6.37% in all the path patterns. While the number of path patterns in this investigation is 8813, the number of words in the dictionary of natural language is more than 100000 [7]. The reason is that contents expressed in natural language cover various topics, but contents of MathML objects cover limited topics. For achieving the good memory cost and good search performance, we should use an index which represents the content of the formula. It is also reported in the area of conventional information retrieval that deleting the high frequent keywords achieves the better search performance [8]. We can expect the improved search performance by deleting the high frequent path pattern for generating indices in our research.

4 Experiment

From the result of the investigation in Section 3, we found that using only one path for an index is not enough for narrowing down the search result. However, if we make indices from all the paths of the MathML object, the inverted file becomes large and many MathML objects are associated to one index. In that case, the cost for checking all the path becomes large. Therefore, we decided to select two paths from all the paths in the MathML object and generate an index by combining these paths. We built a search engine which introduces the above indexing method and conducted an experiment for the evaluation.

We used 87000 MathML objects used in the investigation in Section 3 for this evaluation. We extracted all the paths in the DOM structure of each MathML object. We selected the first path and the deepest path and combined them as an index. By including the fist path in the index, we can represent how the formula started (e.g. The formula expresses the definition or the equation). We think that the deepest path shows the most characteristic part in the formula. Table 4 shows examples of the created inverted file. The format is just a pair of the first path and the deepest path (expressed in XPath and separated by a commna).

Table 4. Examples of created inverted index in the experiment.

Description combining two XPath descriptions	List of document ID
/math/mi[text()="cot"],/math/mover/ mi[text()="\infty"]	47206,47208, 47276,,,
/math/munderover/mo[text()="\Sigma"], /math/msup/mfrac/mn[text()="1"]	850,1211,1379,,,
/math/mo[text()="\int"], /math/mfrac/msqrt/msup/mi[text()="sin"]	18679,39824,39839,,,

We selected one MathML object from the 87000 MathML objects and searched other MathML objects by our system. Figure 6-(a) and 6-(b) are examples. In the search result in Figure 6-(a), the XPath of the first path of the search query represents that the formula starts by $\tan$. The XPath of the deepest path of the search query represents that the formula consists of hierarchical fraction and there is imaginary number i. There are four items in the search result. In this example, the system outputs the item in the search result if the index of the inverted file and the index of the search query just match. If the user wants to obtain the perfect-match result, the system checks all the paths between the MathML object of the search query and the matched MathML object in the inverted file. Then it outputs only the MathML object in which all the paths match with those of the search query.

In Figure 6-(b), the formula starts with π and includes the power of $\sin$. We can see that the search results include the formula with the square of $\sin$ and the cube of $\sin$. Also we can see that it includes the generalized formula that represents the n-power of $\sin$. If the system does not check all the paths, the search result becomes a similarity search with keeping the starting format of the formula and the complex structure the formula.

5 Conclusions

In this research, we built a search engine for the MathML object by introducing the XPath and the inverted file. Firstly we investigated the paths in the DOM structures of MathML objects for generating an index. We found that combining the fist path and the deepest path represent the characteris-

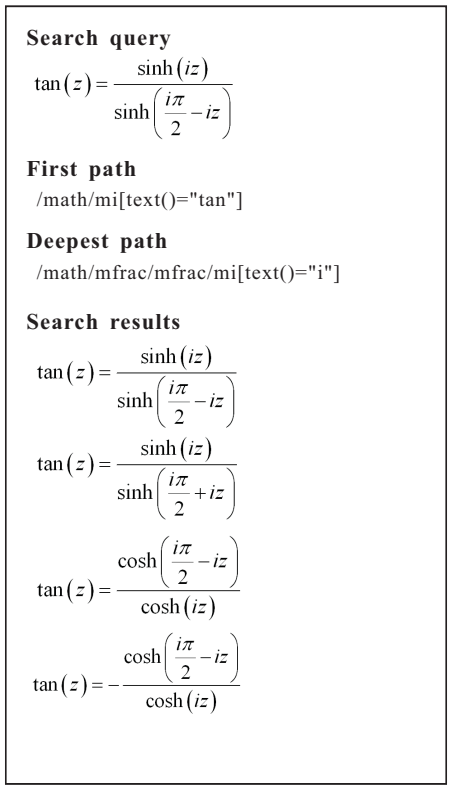


Figure 6. Example (1) of the search result in the experiment.

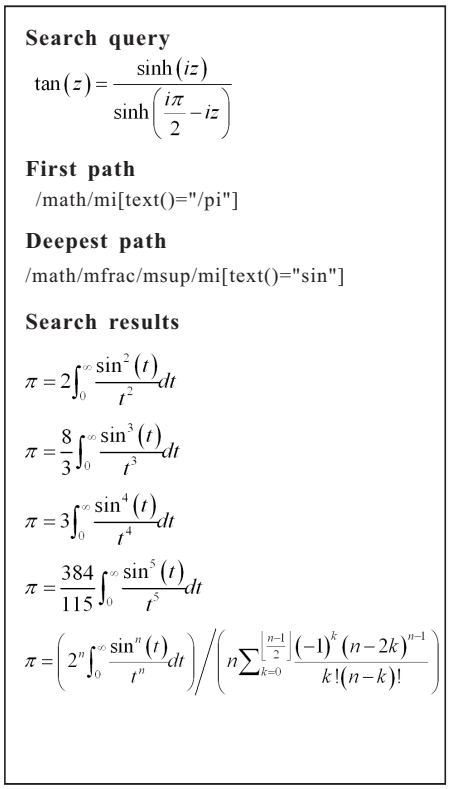


Figure 7. Example (2) of the search result in the experiment.

tics of the formula well. We conducted an experiment to see what kind of results will be produced by using this index type. The results showed that it is useful not only for exact-match retrieval but also for similarity-math retrieval. Furthermore the similarity-math retrieval can keep the constrain like how the formula stats and what is the most important structure.

The future work is weighing the path pattern according to the frequency and introducing AND or NOT operator. We also target the practical application of the search engine. We plan to provide SDK including this search module.

Acknowledgement

This research was partly supported by the NEDO Research Fund for Industorial Technology (Project ID: 06A14501d).

References

[1] W3C: M3C Math Home, http://www.w3.org/math/, 1998.

[2] T. Nakanishi, et al. An Implementation Method of Composite Association Retrieval System for Data of Mathematical Formulas *DBSJ Letters*, 4(1), 2005.

[3] W3C: M3C DOM Home, http://www.w3.org/dom/, 1998.

[4] W3C: M3C XPath Home, http://www.w3.org/TR/ xpath, 1999.

[5] WOLFRAM Research: MATHML CENTRAL, http://www.mathmlcentral.com/, 2007.

[6] NICT: The EDR Electronic Dictionary, http:// www2.nict.go.jp/r/r312/EDR/index.html, 1999.

[7] H. Kohno, et al. Search Engine and Agent University of Tokyo Press, 2002.

[8] K. Kita, et al. Information Retrieval Algorithm Kyoritsu Publisher, 2002.