Improving Mathematics Retrieval

Shahab Kamali and Frank Wm. Tompa

David R. Cheriton School of Computer Science
University of Waterloo
Waterloo, ON, Canada
skamali,fwtompa@cs.uwaterloo.ca

Abstract. Despite the popularity of storing mathematical objects on the web, searching for mathematical expressions is extremely limited. Conventional retrieval systems are inadequate for mathematical expressions, because they are not tuned for text with complex structures that include only a few distinct terms. Surprisingly current approaches to the problem of retrieving mathematical information do not include a formal definition of the similarity between two expressions, and thus fail to find many relevant documents.

In this paper, we present steps to advance mathematics retrieval to incorporate best practices from modern information retrieval. We first review encodings of mathematical expressions currently found on the web, and present the results of our efforts to create an experimental testbed. We formally define the similarity between two mathematical expressions and present the problem of searching for similar mathematical expressions.

1 Introduction

Two text documents are distinguished primarily by the words they contain. Therefore conventional text retrieval systems emphasize their ability to find documents that include certain words and consider the structures in which the words are used secondarily or as a filter. On the other hand, mathematical expressions are objects with complex structures consisting of few distinct mathematical symbols, and they are distinguished at least as much by their structures as by the choices of symbols. As a result, current text retrieval systems cannot effectively search for mathematical expressions. Even the few existing mathematical search systems either completely ignore the structure of expressions, or perform very simple searches only.

To publish, process, and retrieve mathematical expressions, their structure should be encoded as well as their symbolic content. Despite the existence of standards for encoding mathematical expressions, e.g. MathML, our experiments show they are still rarely used to publish mathematical knowledge on the web. Other techniques, e.g. putting the image of an expression in a web page, are still widely used. As a result, recognizing, processing, and retrieving them is a nontrivial task.

In part because of the above difficulties, the problem of mathematical expression retrieval is not effectively addressed. The research in this field is still in the preliminary stage. Even the basic concept of similarity between two mathematical expressions is not formally defined.

In this paper we first describe various encoding schemes for mathematical expressions in Section 2. Then, we present our experience in gathering expressions from the web, with some statistics about the published mathematical data. In Section 4 we formally define the similarity between two mathematical expressions. Based on our definition, we review and classify current mathematical search systems and propose a new approach.

2 Encoding mathematical expressions

Encoding mathematical expressions is a crucial precursor for designing a framework for mathematics retrieval. In order to collect and process mathematical expressions on the web, detailed knowledge about all possible encoding schemes is required. Naturally we prefer an encoding scheme that allows more efficient retrieval. In this section we briefly describe some encoding schemes for mathematical expressions.

2.1 Interchange Formats

Interchange formats are encoding schemes with the capability to create, convert, and manage data on the web.

mathML Mathematical Markup Language (MathML) [7] is a W3C recommended XML application for describing mathematical notations. Since MathML is an application of XML, its syntax is governed by XML syntax rules. A mathematical expression has two different, though related, aspects: the visual appearance and the mathematical content. Based on this observation, MathML defines two standard markup frameworks for representing mathematical expressions: presentation markup and content markup. In Presentation MathML, the two dimensional visual appearance of an expression is encoded. Content markup deals with the functional structure of mathematical expressions.

OpenMath: OpenMath [6] is a standard for representing mathematical expressions, allowing them to be exchanged, stored in databases, or published on the World Wide Web. OpenMath was proposed to provide a standard way of communication between mathematical applications, so unlike MathML, OpenMath is solely concerned with the content of mathematical expressions and not its presentation form.

2.2 Typesetting formats

Typesetting formats, such as LaTeX, are the encoding schemes used by typesetting systems to express mathematical expression. They are not able to encode

the meaning of an expression, but instead they offer a set of symbols and provide ways to put them together in an expression.

2.3 Syntactic computation formats

Symbolic computation software packages, e.g. Maple [3] and Mathematica [4], define languages that offers a syntax for encoding mathematical expressions. These formats are mostly designed to encode the semantics of algebraic equations and are limited to a set of predefined algebraic symbols.

3 Mathematics on the Web

In this section we present the results of searching for some samples from the Web. Starting from Wikipedia [1] and Wolfram [2] root pages, we crawled a set of web pages totaling 60GB in size.

We extracted only 4000 mathematical expressions encoded with MathML (content and presentation), 3000 of them from the MathML test suite by W3C. We conclude that although MathML is considered as a standard for publishing mathematical information on the web, it is still not popular.

Some main publishers of mathematical information including Wikipedia and Wolfram, present mathematical expressions as images. They also describe the expressions with pieces of TEX, but since these pieces are used just to describe the expressions, there are many TEX errors in them. We could collect a set of 300,000 expressions encoded this way. After refining the expressions to fix their TEX errors, using a TEX to presentation MathML translator [9], we translated the collected expressions into MathML.

To get an estimate of the average size of mathematical expressions, we counted the number of nodes in each MathML tree. In Figure 1 the number of expressions for each size from 1 to 250 is shown. In Figure 2, the cumulative distribution function of expressions's sizes is shown. It seems that the size of mathematical expressions on the Web follows an exponential distribution.

As Figure 1 shows, about 10% of the collected samples have very small sizes. They are usually very simple expressions like x^2, and $1+2$. The majority of the expressions have sizes between 10 and 130 nodes. These are some examples:

$$\tilde{e}_i(b \otimes b') = \begin{cases} \tilde{e}_i b \otimes b', \text{ if } \phi_i(b) \geq \epsilon_i(b'), \\ b \otimes \tilde{e}_i b', \text{ if } \phi_i(b) < \epsilon_i(b'), \end{cases}$$

$$\gamma\gamma^* = \gamma^*\gamma,\ \alpha\gamma = \mu\gamma\alpha,\ \alpha\gamma^* = \mu\gamma^*\alpha,\ \alpha\alpha^* + \mu\gamma^*\gamma = \alpha^*\alpha + \mu^{-1}\gamma^*\gamma = I,$$

Expressions with sizes larger than 150 nodes are very rare. These are examples of large expressions:

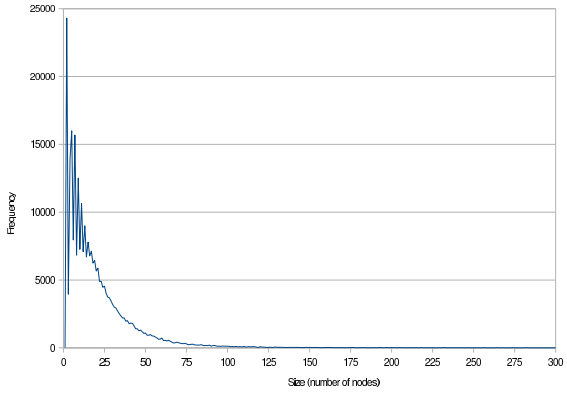

Fig. 1. The number of expressions of size n.

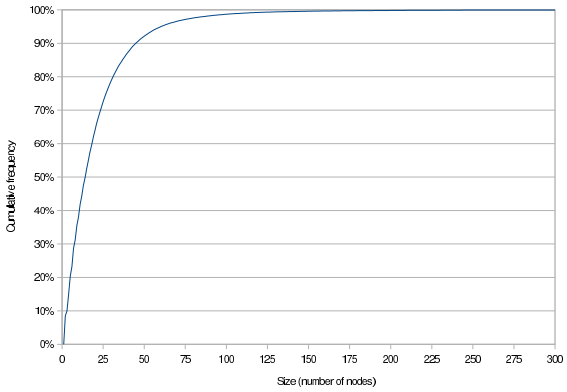

Fig. 2. The percentage of expressions with size at most n.

$$t = \begin{cases} \sqrt[3]{-\frac{q}{2} + \sqrt{\frac{q^2}{4} + \frac{p^3}{27}}} + \sqrt[3]{-\frac{q}{2} - \sqrt{\frac{q^2}{4} + \frac{p^3}{27}}} \\ \left(-\frac{1}{2} + i\frac{\sqrt{3}}{2}\right) \sqrt[3]{-\frac{q}{2} + \sqrt{\frac{q^2}{4} + \frac{p^3}{27}}} + \left(-\frac{1}{2} - i\frac{\sqrt{3}}{2}\right) \sqrt[3]{-\frac{q}{2} - \sqrt{\frac{q^2}{4} + \frac{p^3}{27}}} \\ \left(-\frac{1}{2} - i\frac{\sqrt{3}}{2}\right) \sqrt[3]{-\frac{q}{2} + \sqrt{\frac{q^2}{4} + \frac{p^3}{27}}} + \left(-\frac{1}{2} + i\frac{\sqrt{3}}{2}\right) \sqrt[3]{-\frac{q}{2} - \sqrt{\frac{q^2}{4} + \frac{p^3}{27}}} \end{cases}$$

$$G_{ik} = \frac{1}{4\pi\mu} \begin{bmatrix} \frac{b}{r} + \frac{x^2}{r^3} - \frac{ax^2}{r(r+z)^2} + \frac{az}{r(r+z)} & \frac{xy}{r^3} - \frac{axy}{r(r+z)^2} & \frac{xz}{r^3} - \frac{ax}{r(r+z)} \\ \frac{yx}{r^3} - \frac{ayx}{r(r+z)^2} & \frac{b}{r} + \frac{y^2}{r^3} - \frac{ay^2}{r(r+z)^2} + \frac{az}{r(r+z)} & \frac{yz}{r^3} - \frac{ay}{r(r+z)} \\ \frac{zx}{r^3} + \frac{ax}{r(r+z)} & \frac{zy}{r^3} + \frac{ay}{r(r+z)} & \frac{b}{r} + \frac{z^2}{r^3} \end{bmatrix}$$

Our observations show that large expressions are usually composed of several smaller expressions in the form of matrices or arrays.

4 Similar expressions and mathematics retrieval

The definition of similarity between two mathematical expressions is a key concept that significantly affects a mathematical information retrieval system, but a formal definition of similarity is missing in the literature. Here we formally define the similarity between two mathematical expressions.

We mentioned earlier that a mathematical expression can be viewed from two perspectives, its content, i.e. the meaning, and its presentation, i.e. the appearance. A fundamental question is whether the content or the presentation should be considered while searching for a set of similar expressions to a mathematical expression.

Usually there are many different ways to represent a mathematical concept. In different fields of math, different symbols might have the same meaning. Often numbers and names of variables are less important or irrelevant to meaning. If two expressions have different appearance but the same meaning (i.e. synonymy), they should be treated the same. In this way the search results are more comprehensive, yielding a high recall factor. On the other hand, occasionally different expressions with the same appearance have different meanings (i.e. polysemy), which can yield low precision. These factors favour content-based retrieval over retrieval based on presentation form.

Despite these advantages, however, the content based approach has two severe shortcomings. First, as different fields of science evolve, new mathematical concepts and symbols are defined. Thus creators and users of mathematical expressions must agree on a common dictionary of terms. Updating such a dictionary to include definitions of new symbols or new definitions of existing symbols is a nontrivial task and will almost certainly need manual management. Second, most mathematical resources on the Web (including presentation MathML, TEX, postscript files, and image-in-HTML) are presentation-based encodings of mathematical expressions and provide little information about meaning. In this approach there is no need to specify (or even understand) the meaning of each symbol, and document authors will be loathe to spend the time needed to add the extra semantics required for better retrieval.

Based on the above discussion, we conclude that presentation-based search is more robust for dealing with today's information on the Web. So by default we assume no information about the content of an expression is provided. Since MathML is widely accepted as a standard for publishing mathematical information, we base our definitions on the assumption that all expressions are encoded with presentation MathML.

Definition 1. *For an expression X, $T_X(V, E)$ is the XML tree corresponding to the presentation MathML encoding of X. V and E represent the sets of nodes and edges of T_X respectively.*

In the remainder of this paper, we assume an expression, X, and its corresponding tree, T_X, are the same entities, and we use X and T_X interchangeably. The terms "expression" and "tree" (also "subexpression" and "subtree") are also

used interchangeably. We also assume the root of a MathML tree has a unique label which is the same for all trees.

Definition 2. *Given a tree, $T(V, E)$, to each node $v \in V$ we assign a weight, $\omega(v)$, which is by default 1. We also define a weight function $\omega(T) = \sum_{v \in V} \omega(v)$.*

Definition 3. *Given a labeled ordered tree, $T(V, E)$, a* part *of T is a connected labeled ordered tree comprising a node in T and some of its descendants in T. A* complete part *of T is a subtree comprising a node in T and all of its descendants.*

Definition 4. *Two trees* match *if they have the same structure and the corresponding nodes have the same labels. Given two trees, T and U, $U \lhd T$ means U matches one or more parts of T.*

Definition 5. *Given two labeled ordered trees, $T_1(V_1, E_1)$ and $T_2(V_2, E_2)$, a common part, C, is a tree such that $C \lhd T_1$ and $C \lhd T_2$. A common part of T_1 and T_2 is a* common subtree *if it matches a complete part of T_1 and a complete part of T_2. The* common part set *is a set composed of all common parts and is denoted $T_1 \bigcap T_2$. We define the weight of $T_1 \bigcap T_2$ to be the maximum of the weights of its members, $\omega(T_1 \bigcap T_2) = max\{\omega(c), c \in T_1 \bigcap T_2\}$.*

Definition 6. *To each node in an expression, depending on its type, a set of attributes is assigned, e.g. the name of an operator or the value of a number. Normalization is the process of removing some unnecessary information (i.e. insignificant attributes) from an expression. An expression is normalized if all insignificant attributes are removed.*

For example the value of a number is usually insignificant and can be removed during the normalization but the name of an operator should be conserved, Figrue 3. To normalize an expression, a set of rules must be defined to specify which attributes are insignificant. We assume, for example, that by default names of variables is insignificant. Alternatively, a feature can be added to an encoding scheme, like MathML, that allows indicating the significance of an attribute at publication time. Normalization can also be broadened to include some standardization of mathematical expressions based on equivalence. For example, we might replace $x + y \cdot 0$ by x.

Definition 7. *For two expressions E_1 and E_2, $Sim(E_1, E_2)$ quantifies their similarity:*

$$Sim(E_1, E_2) = \frac{\omega(E_1 \bigcap E_2)}{\omega(E_1) + \omega(E_2)} \tag{1}$$

Given two expressions, E_1 and E_2, several possible similarity patterns between them can be defined:

– Mathematical equivalence: E_1 and E_2 are mathematically equivalent if it could be mathematically proven that they are the same, e.g. $x + y$, $y + x$ and x^{-1}, $\frac{1}{x}$.

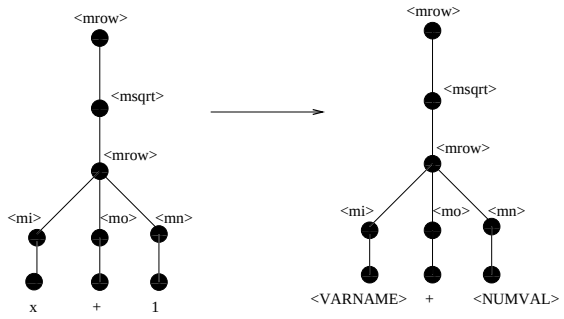

Fig. 3. The MathML tree of $x + 1$ on the left and the normalized tree on the right.

– Identical: E_1 and E_2 are identical if they are exactly the same, down to the same variable names and numbers.

– Syntactic equivalence: E_1 and E_2 are syntactically equivalent if they are identical after normalization, e.g. $sin(x)$ and $sin(y)$.

– n-similarity: Normalized expressions E_1 and E_2 are n-similar if $sim(E_1, E_2) \geq n$, where n is a parametric value that determines a threshold on the similarity between two expression. N-similarity can be classified into three categories:
  1. Subexpression n-similarity: There is subexpression n-similarity between E_1 and E_2, if they are n-similar and a subtree (i.e., a complete common part) $S \in E_1 \bigcap E_2$ exists such that $\omega(S) = \omega(E_1 \bigcap E_2)$, e.g. there is subexpression similarity between $sin(x)^2$ and $\sqrt{sin(x)}$ for any value of $n \leq \frac{9}{24}$, Figure 4.

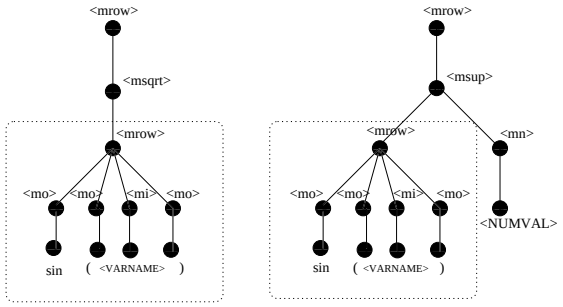

Fig. 4. Normalized MathML trees of $\sqrt{sin(x)}$ and $sin(x)^2$.

  2. Structural n-similarity: There is structural n-similarity between E_1 and E_2 if they are n-similar and a common part, C, exists such that $Label(root(C)) = Label(root(E_1)) = Label(root(E_2))$ and $\omega(C) = \omega(E_1 \bigcap E_2)$, e.g. there is subexpression similarity between $\sqrt{x + x^2}$ and $\sqrt{2x}$ for any value of $n \leq \frac{5}{19}$, Figure 5.

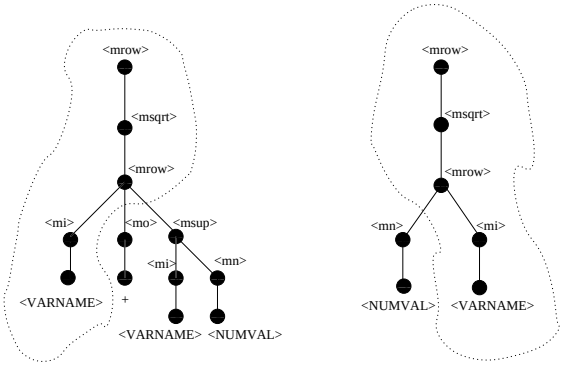

Fig. 5. Normalized MathML trees of $\sqrt{x+x^2}$ and $\sqrt{2x}$.

3. Irregular n-similarity: There is irregular n-similarity between E_1 and E_2 if there is n-similarity but neither structural nor subexpression n-similarity between them, e.g. $(sin(x)+1)^2$ and $\sqrt{(sin(x)+y)}$

Definition 8. *Given a mathematical expression, Q, and a set of mathematical expressions, S, the search problem is to find the k most similar expressions to Q. k is a parameter that is specified by the user.*

5 Current mathematics retrieval systems

In this section we review the architecture of the mathematics retrieval systems proposed so far and explain the shortcomings of each approach.

5.1 MathWebSearch

MathWebSearch [12] is a search engine for mathematical expressions based on semantics, i.e. it does not consider the visual appearance of expressions. In this scheme mathematical expressions are interpreted as prefix forms, i.e. first the name of a function is presented and then its arguments, e.g. $f(x, y)$. A set of mathematical expressions is stored in a tree-based data structure, called a *substitution tree*, in a way that the common parts of the expressions are shared. Each node of this data structure is called a substitution that corresponds to a function. A substituition also serves as an argument of its parent's corresponding function. A mathematical expression could be represented as paths starting from the root in the tree. The advantage of this approach is that parts that are in common among multiple expressions are stored only once. The search for an expression could be performed simply by traversing the substitution tree, i.e. starting from the root and trying to find the sub-terms of the queried expression.

MathWebSearch can process and index expressions encoded with content MathML or OpenMath. The schemes that encode the appearance of expressions,

e.g. presentation MathML and TEX, are not well suited for use by MathWebSearch. This is a major drawback because the majority of the mathematical expressions on the web are encoded based on their appearance. The other shortcoming of MathWebSearch is that it only supports exact matching between expressions. It is suggested that while adding an expression to the substitution tree, all its sub-expressions should also be added as independent expressions. This can partially address the problem but the number of expressions will grow exponentially and still there are other similarity patterns that are not covered.

5.2 Searching for integral tables

Einwohner and Fatman [8] propose a data structure and an algorithm for storing and retrieving symbolic indefinite or definite integrals. In this approach, many pre-calculated integrals are stored in a table. A requested integral matches an entry in the table if its integrand agrees with that of the table entry up to a choice of parameters (e.g. $\frac{1}{x^2+1}$ would match $\frac{1}{x^2+1}$, $\frac{1}{x^2+a}$, and $\frac{1}{x^2+a^2}$). A match is sought through a sequence of successive approximate matches, which is controlled by a set of keys associated with the integrand. Each key is related to a fragment of the integrand, and is determined on insertion of the integrand in the table and matched when the table is searched for a given integrand. On searching, the first key of the queried integrand is looked up in the appropriate hash-table, yielding a list of acquisition numbers (each acquisition number uniquely represents an integrand). If the list is empty, the integral is not in the table. Otherwise, the list of acquisition numbers corresponding to the second key is intersected with the list of acquisition numbers of the first key. If the intersection is empty, the integrand is not in the table, otherwise the algorithm continues similarly with successive keys. The above search strategy is based on many assumptions about structures of expressions that make its application limited to specific cases. This is especially a drawback for a mathematical search engine that is supposed to retrieve recent mathematical information on the web.

5.3 DLMF search system

The search system for the Digital Library of Mathematical Functions, DLMF [13], follows two different approaches. In the first approach [15], a textual language, TexSN, is defined to normalize queries and the math content of the digital libraries into canonical forms. After that a search is performed to find the mathematical expressions that exactly match a query.

In the other approach [16], a mathematical expression is regarded as a set of mathematical terms, e.g. operators. Following an algorithm based on the vector space model as used in text information retrieval, the objects are ranked regarding a given query. The only difference is that instead of term frequency (tf) and inverted document frequency (idf), other weighting schemes are considered. This approach treats a mathematical expression as a document consisting of a set of mathematical symbols and does not take into account its structure.

5.4 MathFind

MathFind [14] is a math-aware search engine developed at Design Science Incorporation. The general idea is to encode mathematical expressions as text objects and use the same techniques for ranking text documents. In MathFind, a layer is added to a typical text search engine to analyze expressions in MathML and decomposes each expression into a sequence of text encoded math fragments that are analogous to words in text documents. No further details about this search engine and its performance are published.

5.5 Formal math search

The proposals for formal math search [10, 5] mostly focus on very detailed and formal query languages. Each expression is a set of symbols and the expressions containing a symbol are selected and then intersected using relational database operations. The structure of mathematical expressions is not considered in this approach.

6 Challenges in mathematics retrieval

In none of the work to date is the problem of mathematics retrieval clearly defined. Also none of the proposed approaches cover all the defined similarity patterns. Many do exact matching only and others do not consider the structure of mathematical expressions.

Like any information retrieval system, a mathematics retrieval system should collect mathematical expressions that appear on the Web, it should process and store them in a data structure, and finally it should process queries and retrieve and rank expressions based on similarity to a given query.

To collect mathematical expressions, we must harvest them from Web pages. Detecting mathematical expressions in some cases in a nontrivial task, e.g. mathematical expressions may be hidden in postscript files. To process collected mathematical expressions, we should translate them into an encoding scheme that facilitates the indexing and ranking algorithms, e.g. presentation MathML. This translation is another challenging task in cases where limited information about published expressions is provided and there are many ambiguities, e.g. images of expressions are difficult to process. We also wish to capture semantic information wherever possible so that the system can perform better by applying content-based search.

After this step, an indexing algorithm stores the expressions in a data structure. This data structure should allow fast and accurate retrieval of similar expressions to a given query. It should also preserve the structure of the expressions. There is a tradeoff between the size of the index, the speed of the ranking, and the accuracy of the search, which in part depends on how much of the structure is preserved. The complexity of directly comparing the structure of two trees is quite high [11]. Therefore, storing whole trees in the index and

comparing them one by one to a given query is inefficient. On the other hand, breaking the structure into parts and indexing the parts has the risk of loosing some structural information. Cases in between that allow fast retrieval and conserve all structural information result in very large indexes. Considering these facts, designing the indexing scheme is a critical and nontrivial step.

The performance of a mathematics retrieval engine must be measured not only with respect to efficiency, but also with respect to effectiveness. Thus an important part of this research is to design and evaluate ranking algorithms. The definitions we provide for n-similarity reflect the notion of matching expressions, but they must be tested to see if they capture humans' perception of relevancy in practice. Experiments also need to be conducted to decide how to weight components of expressions (e.g., should variable names be weighted less than numbers? should simple operands be weighted less than operators? does the inclusion of inverse document frequency (idf) improve mathematics retrieval?). Such evaluations requires the creation of a testbed of expressions together with a widely accepted set of relevancy judgments. Our initial crawling of the Web is a starting point for such a testbed.

There are other problems to address: dealing with polysemy and synonymy, allowing embedded text in mathematical expressions (e.g. text comments in an expression), and supporting more sophisticated query languages (e.g. allowing a mixture of text and math).

References

1. http://www.wikipedia.org.
2. http://www.wolfram.com.
3. Maple learning guide. Maplesoft, a division of Waterloo Maple Inc, 2003.
4. Mathematica 6. Wolfram Research Documentation Center, 2008.
5. G. Bancerek. Information retrieval and rendering with mml query. In *Proc. of MKM 2006, Lecture Notes in Artificial Intelligence 4108*, pages 266–279. Springer Verlag, 2006.
6. O. Caprotti, D. Carlisle, and A. Cohen. The OpenMath standard. *The OpenMath Esprit Consortium*, 2002.
7. D. Carlisle, P. Ion, and R. Miner. Mathematical Markup Language (MathML) version 3.0. In *W3C Working draft*, 2007.
8. T. H. Einwohner and R. J. Fateman. Searching techniques for integral tables. In *International Symposium on Symbolic and Algebraic Computation*, pages 133–139, 1995.
9. J. Grimm. Tralics, a latex to xml translator. In *IRNIA, Institut National De Recherche en Informatique et en Atomatique*, 2008.
10. F. Guidi and I. Schena. A query language for a metadata framework about mathematical resources. In *In Asperti et al*, pages 105–118, 2003.
11. M. Halldrsson and K. Tanaka. Approximation and special cases of common subtrees and editing distance. In *Proc. 7th Ann. Int. Symp. on Algorithms and Computation, Lecture Notes in Comput. Sci. 1178, Springer-Verlag*, 1996.
12. M. Kohlhase and I. A. Sucan. A search engine for mathematical formulae. In *Artificial Intelligence and Symbolic Computation, LNCS*, pages 241–253, 2006.

13. B. Miller and A. Youssef. Technical aspects of the digital library of mathematical functions. *Ann. Math. Artificial Intelligence*, 2002.
14. R. Munavalli and R. Miner. Mathfind: a math-aware search engine. In *SIGIR*, page 735, 2006.
15. A. Youssef. Search of mathematical contents: Issues and methods. *IASSE*, 2005.
16. A. Youssef. Methods of relevance ranking and hit-content generation in math search. *Calculemus/MKM*, 2007.