Methods to Access and Retrieve Mathematical Content in ActiveMath

Paul Libbrecht and Erica Melis

German Research Center for Artificial Intelligence, Saarbrücken, Germany

Abstract. This article describes how mathematical content items and formulæ are processed, retrieved, and accessed in ActiveMath. Central to the retrieval and access is a search tool which allows for searching text, attributes, relations and formulæ, and presenting items. The search tool has been evaluated according to the standard measures of precision and recall as well as for usability. We report results of these evaluations.

1 Introduction

Increasingly, (mathematical) content is enriched with semantic information in order to make it interoperable and better accessible for men and machines. To employ the semantics, new retrieval techniques have to be developed. Since our learning environment, ActiveMath, works with semantically represented maths content, we developed new techniques to make the semantics of items and of mathematical formulæ accessible with common information retrieval (IR) technologies. The developed techniques convert formulæ to indices and allow sub-expressions to be matched (with wild cards) while relying on the semantic representation of OpenMath [BCC+04]. Moreover, we contribute the implementation of a search tool that does not only retrieve maths content items but also provides access to related items, ranks them, and presents them in an advanced human-readable rendering.

1.1 ActiveMath

ActiveMath [MAB+01] is an integrated learning environment on the Web. Its content uses an extension of the `OMDoc` language [Koh00], which itself is an extension for mathematical documents of the OpenMath [BCC+04] mathematical objects encoding. `OMDoc`'s item granularity is that of definitions, examples, exercises etc., and it is the level which is mostly used for management, referencing, and search in ActiveMath. Each *content item* can contain text with links and semantically encoded formulæ and is annotated with mathematical and pedagogical attributes and relations [MAF+03].

ActiveMath supports learners in many ways including:

– generation of courses/books adapted to the user's learning goals, scenarios and knowledge

– interactive exercises with mathematical input, evaluation, and feedback
– various learning support tools
– an open learner model.

The presentation of courses in the form of books provides an intuitive navigation paradigm. This and the search facility are two methods to access the content items on an ActiveMath server, allowing a learner to see the content item, reference it in communication, and find it.

A new ActiveMath facility for multi-dimensional search is presented in this paper. It searches through large `OMDoc` content repositories for text, formulæ, and items' characteristics.

The paper first recapitulates the learning and authoring situations in which content items are presented, searched for. This is followed by a description of the search tool's components and evaluation results. Finally related research and future works are presented.

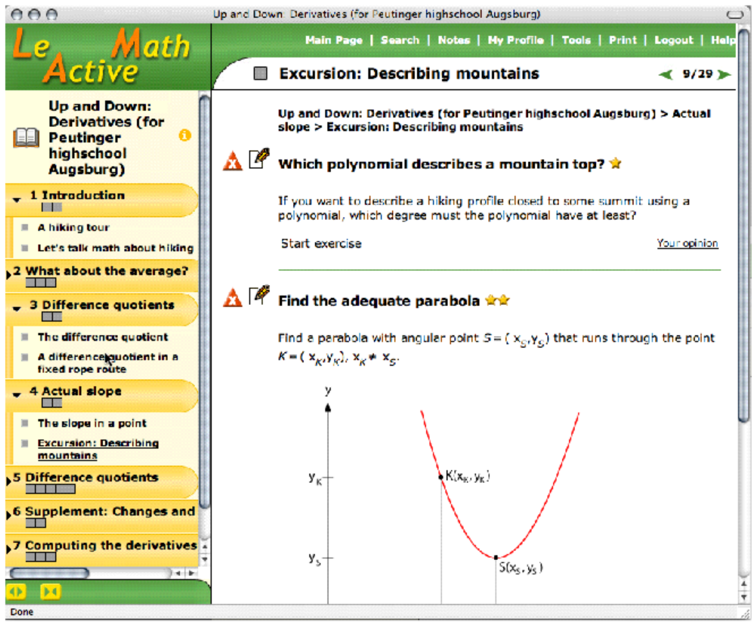
2 Access to Content Items

This section reviews common practices for mathematical knowledge management and indicates what ActiveMath is doing in this direction. We use the term *access* very broadly for the ability of a user to reach a given item, symbol, or formula. Accessing an item means to have it presented in a browser, to be able to reference it or to let other programmes download it. Access can be granted through linking or through search.

The search for mathematical texts or formulæ has the same purposes as general search and can be included into general search engines with additional components or via preprocessing. In ActiveMath, search can serve the system, e.g. adaptive course generation, or it can serve the learner to retrieve information for learning, to learn about the relationship among knowledge items, mathematical symbols, etc., to communicate search results, or to enquire about communicated information and mathematical expressions. Search can also serve authors and tutors which will search content with textual, formal, and attribute queries in many situations. For example, when authors are writing new content, are reviewing it, or are assembling new new courses from existing materials they search for items by their content, pedagogical attributes, or relations.

In ActiveMath, access to mathematical content may start when opening a "book" that contains items previously assembled by an author. An example page of a book is in Figure 1. While the learner is reading she may be wondering about a concept or symbol that is present on the page. Content items for this concept or symbol, e.g. a definition of it, may be directly linked by the author in which case the learner could simply click on it and the item would be displayed. The concept behind a mathematical symbol is presented via a link. The concept can also be searched by its name.

Situations in which a learner may wish to use links or search include the process of active learning exercising in which, e.g., a rule, definition, or theorem has to be recalled and applied. Such links can be offered in feedback of an exercise step. With her own initiative, the learner could use a search tool for textual search or formulæ search in order to recall and/or copy semantics to a particular application, e.g. a concept mapping tool.



Up and Down Derivatives (for Peutinger highschool Augsburg)
LeActiveMath
Main Page | Search | Notes | My Profile | Tools | Print | Logout | Help
Up and Down: Derivatives (for Peutinger highschool Augsburg)
1 Introduction
A hiking tour
Let's talk math about hiking
2 What about the average?
3 Difference quotients
The difference quotient
A difference quotient in a fixed rope route
4 Actual slope
The slope in a point
Excursion: Describing mountains
5 Difference quotients
6 Supplement: Changes and
7 Computing the derivatives
Done
9/29
Start exercise
Your opinion
y
y_K
y_S
$K(x_K, y_K)$
$S(x_S, y_S)$

Excursion: Describing mountains

Up and Down: Derivatives (for Peutinger highschool Augsburg) > Actual slope > Excursion: Describing mountains

Which polynomial describes a mountain top?

If you want to describe a hiking profile closed to some summit using a polynomial, which degree must the polynomial have at least?

Find the adequate parabola

Find a parabola with angular point $S = (x_S, y_S)$ that runs through the point $K = (x_K, y_K)$, $x_K \neq x_S$.

Fig. 1. A book presentation in ActiveMath.

3 Ingredients of a Search Tool

In the sketch of the ingredients needed for a search tool we follow [You05] but our focus is on the mathematical content items and formulæ as opposed to the mere function orientation of [You05]. We present additional ingredients that are required for a system that does not only search but also presents the content.

Identifiers and References Content items need to be addressable in order to be extracted and referenced. References to items should be exchangeable in e-mail communications. Therefore, they need to be context independent and short.

Storage, Extraction, and Presentation Mathematical content items are embedded within larger documents. The documents need to be stored in repositories and ways for their extraction are needed, so that they can be queried and presented.

Query Input Queries for textual fragments are well known, being the focus of Information Retrieval. Queries for item attributes can be input using form-based interfaces or using a dedicated syntax. Queries for mathematical expressions is less developed. A usage of a facility to input the formulæ is important.

Indexing, Analysis, and Back-End Query Information Retrieval as in [vR79] provides powerful methods to convert rows of tokens to an index for which search results can be computed efficiently. This relies on a tokenization process called *analysis* which, among others, *stems* words (e.g. remove plural), removes too frequent words, etc.

We developed a special treatment for the highly structured nature of mathematical formulæ in the indexing process as well as in the analysis process transforming (queries for) mathematical expressions into (back-end queries for) rows of tokens.

Results Presentation The way a search tool displays a result is very important. For users who often only visit the first few matches, information retrieval has introduced the notion of relevance of a match, a score that is assigned to a document matching a query; results with highest relevance should be presented first. The results' list should provide visual hints about the type of mathematical item presented, its title, its ranking.

4 The ActiveMath Search Tool

The search tool of ActiveMath searches (an index produced from) the `OMDoc` sources. It provides a friendly user-interface that combines plain-text search with item-attributes and formulæ search. It presents the results of the queries as well as individual items and their relations.

A coarse architecture of the tool is depicted in Figure 2 which presents the flow of information at indexing time (on the left), at query time (right bottom), and at item presentation time (right top).

The description of the ActiveMath search tool follows the structure of §3.

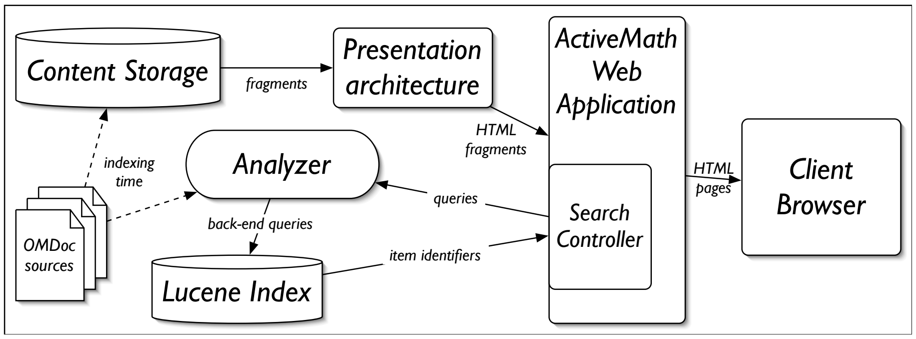
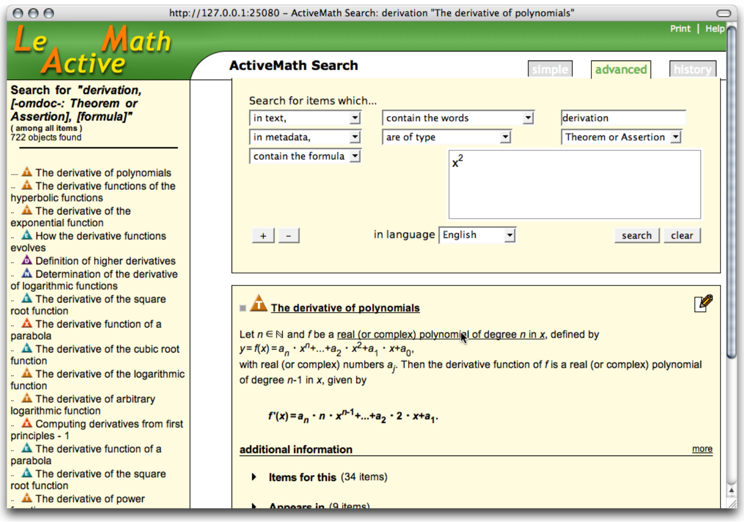


Fig. 2. Architecture of the ActiveMath search tool



http://127.0.0.1:25080 – ActiveMath Search: derivation "The derivative of polynomials"

Search for "derivation, [-omdoc-: Theorem or Assertion], [formula]"
(among all items)
722 objects found

- The derivative of polynomials
- The derivative functions of the hyperbolic functions
- The derivative of the exponential function
- How the derivative functions evolves
- Definition of higher derivatives
- Determination of the derivative of logarithmic functions
- The derivative of the square root function
- The derivative function of a parabola
- The derivative of the cubic root function
- The derivative of the logarithmic function
- The derivative of arbitrary logarithmic function
- Computing derivatives from first principles - 1
- The derivative function of a parabola
- The derivative of the square root function
- The derivative of power

Search for items which...

in text, | contain the words | derivation
in metadata, | are of type | Theorem or Assertion
contain the formula | x^2

in language English

The derivative of polynomials

Let $n \in \mathbb{N}$ and f be a real (or complex) polynomial of degree n in x, defined by
$y = f(x) = a_n \cdot x^n + \ldots + a_2 \cdot x^2 + a_1 \cdot x + a_0$,
with real (or complex) numbers a_j. Then the derivative function of f is a real (or complex) polynomial of degree n-1 in x, given by

$$f'(x) = a_n \cdot n \cdot x^{n-1} + \ldots + a_2 \cdot 2 \cdot x + a_1.$$

additional information more

- **Items for this** (34 items)
- **Appears in** (9 items)

Fig. 3. The advanced search user interface.

Identifiers and References Each content item is marked by a content identifier. References to content items are presented as links within the browser presentations which makes them exchangeable by most desktop applications. Since these URIs are also downloadable, they provide an entry point to Web-robots. Among others, we used this in a comparison between the Google and ActiveMath search engines described in §5.

Storage and Extraction The `OMDoc` files are loaded in a content storage which splits the items and prepares them to be served individually. The search and presentation engines request the content of items, their metadata and the relations between them.

Query Input The user interface of the ActiveMath search tool allows for a set of queries which are expanded to low-level queries referring to the index. ActiveMath offers several kinds of queries: text queries, attribute queries, and formulæ queries. Simple queries allows the input of a conjunction of text and attribute queries.

In the advanced search mode users edit a Boolean combination of text, attribute, and formulæ queries: simple text fields are used for text queries, item attributes' queries are input using pop-up-menus, while formulæ queries are input with the Wiris input-editor.[1] A screenshot of the advanced search user-interface is in Figure 3.

Indexing, Analysis, and Back-end Queries The indexing process follows the Information Retrieval approach: the content is read from the `OMDoc` sources and

[1] See `http://wiris.com/` for more information about the OpenMath input-editor that is provided in ActiveMath.

decomposed by the *analysis process* in parallel streams of tokens. These streams are indexed by the Lucene library.[2] Each token is stored on the disk along with its position in the stream in a way that allows queries for (rows of) tokens to be efficiently matched.

We have developed an analysis process which converts the title, annotations metadata, formal and textual content (including formulæ) to tokens as follows:

- Annotations' attributes are stored as key-value pair tokens in their own fields.
- Word analysis applies the classical stemming and stop-words filtering following Martin Porter's algorithm [Por05].
- Words are also converted into a row of *phonetic tokens* which converts two phonetically equivalent words in the target language to the same phonetic tokens. This is a language-dependent process.
- Mathematical formulæ are converted from OpenMath to a row of tokens following the depth-first walk of the expression tree. This enables sub-terms to be matched.

4.1 Example Tokenization and Queries

Consider the following content item:

Trigonometric exercise

Let us assume $a + b = k$.

Our analysis process decomposes its content in named fields each populated with a sequence of tokens passed to the Lucene library for indexing. For the content item above, the following tokens are provided to the index:

```
id:                 trigExo
attr:               type:exercise
title-en:           trigonometr exercis
text-en:            let us assum _(_1 _OMS_relation1/eq
                     _(_2 _OMS_arith1/plus _OMV_a _OMV_b _)_2 _OMV_k _)_1
text-phonetic-en :  LT US ASMN
```

With these token-streams in the index, queries for exact text, fuzzy text, item attributes, simple formulæ and formulæ with wild cards can be performed. They match the item each with a particular relevance score computed on the basis of the field and type of match (e.g., matches in titles are *boosted* by a factor of 10 as we expect them to be more relevant than matches in text):

- If the user enters "trigonometry" while working in English, the analysis converts this word to a query for token `trigonometr`, which is exactly matched to the field `title-en` of our item yielding score 10.0. If the user enters the word "assume", the analysis converts it the token "assum" which is exactly matched to the field `text-en` yielding a score of 1.0.

[2] Lucene is a Java library for high-performance retrieval at the Apache Software Foundation, `http://lucene.apache.org/`.

- If the user enters "asuming", the tokenization converts it to a query for the token "ASMN" in the *text-phonetic-en* field which is matched to the content of the phonetic english field (with score 0.8).
- A query for the type exercise would be reformulated as an index-query for the token `type:exercise` in the field `attr` which is matched to our item with score 1.0.
- If the user inputs the formula $a + b$ as formula query, it is translated to a query for the row of tokens

  `_(_`i `_OMS_arith1/plus _OMV_a _OMV_b _)_`i

  where i ranges from 1 to the maximum-depth in the index. This is exactly matched, with $i = 2$, to the tokens of our OpenMath representation of $a+b$ and thus yields a score of 1.0.
- the formula $? = k$ can be input as query by assigning the wild card role to the ? sign. It is translated to a query for the token sequence

  `_(_`i `_OMS_relation1/eq _? _OMV_k _)_`i

  where i ranges from 1 to the maximum-depth in the index and where `_?` is a wild card match of any row of tokens with the exception of the token `_(_`i. This can be exactly matched, with $i = 1$ to the OpenMath representation of our formula. The score of such a match is 1/6 which is computed on the length of the matched wild card.

Results Presentation The ActiveMath search tool returns the first page of results. Each result is displayed with bullets indicating the score, an icon of its type, and its title. The user can click it to obtain a display of the item.

The plain-text search, by default, returns conceptual content items of the current book with a sufficient relevance; a click can generalize this query. For a mathematical symbol only its definitions is presented. The tool is complemented by links that trigger an equivalent query to external sources of mathematical content on the Web.

Clicking on an item in the result list presents the item view. The ActiveMath presentation architecture converts the semantic `OMDoc` source into a format that is highly readable and is linked to other functionalities of ActiveMath; for example references to other items in the `OMDoc` source are transformed to HtML anchors linked to its item view. The system uses XSLT, caches, and a templating language to provide these presentations. It can render in HtML, xHtML +MathML, and pdf. To support the rendering of mathematical symbols, authorable notations are provided, see [MLUM06]. The *item view* that presents single items is accessible in each presentation of the item.

The description we have provided above shows that the ActiveMath search tool can be used to search for text with reasonable tolerance, for item attributes, as well as for formulæ with wild cards, that is, placeholders that are matched with any term.

5 Evaluation of the Search Tool

The search tool of ActiveMath has been evaluated along two methodologies. The first is a formative user testing. The second is a typical search evaluation with measures for precision and recall which is additionally complemented by a comparison to the Google search engine.

The tests were preformed with the LeActiveMath calculus content [LG06] a corpus equivalent to a typeset book of about 500 pages in English with full translations to German and Spanish. The index contains 2761 documents made of 560'259 tokens: 182765 words and 377'494 mathematical tokens spread in 36'389 formulæ, and 13'681 attribute tokens. On disk, this index takes about 10% of the size of its `OMDoc` sources.

Performance The simple text queries for learners are the slowest, ranging from 500 to 1500 milliseconds which is due mostly to the time taken to verify types of each result and replace symbols by their definitions. The advanced queries for mathematical terms, attributes, and words (both fuzzy and not) take between 50 and 150 milliseconds.

Formative evaluations At the University of Edinburgh 12 mathematics students were invited to discover and use the ActiveMath learning environment under the supervision of an expert. The *Think-aloud* protocols were taken. The evaluation indicated the learners found ActiveMath search tool quite useful and enjoyed an acceptable ease of use. It suggested the importance of labelling content items by types and indicated that learners start grasping the structure of content items, when using the search tool. The evaluation revealed a few usability glitches, most importantly the incomprehensibility of the word *metadata* to qualify queries for items' attributes.

Three German high school classes have used the same tools and content for several weeks. First observations from log files indicate that 95% of the spontaneous usage of the search tool are simple text queries.

Precision and Recall Evaluation This sample of the queries enriched with expected matches was tested as it is commonly done for a precision-and-recall evaluation [Mah06]. Some queries with their precision and recall measures are displayed in Table 1.

type	**query**	**evaluation**	**precision**	**recall**
plain-text	durchschnittliche	14 matches, 4 correct, no miss	0.286	1.0
plain-text	tangant	two results correct no miss	1.0	1.0
plain-text	sin	more than 20 matches, all wrong	0.0	0.0
formula	x^2	1 result correct, no miss	1.0	1.0
plain-text	theorem about quotient	1 correct match, no misses	1.0	1.0

Table 1. A few example queries, their precision and recall.

query	ActiveMath amount matches	Google amount matches
whenever function differential quotient	1	1
tangent convex concave	15	8
parabola maximum	6	17
water maxmimum	39	0
tangant maximum	31	0
sin maximum	48	0
inflection point	243	27
inflection point (no fuzzy)	20	27
sketch	69	1
cauchy sequence	20	65

Table 2. A few example queries with the number of matches in ActiveMath and Google search tools.

The mean recall value is 0.93 (very high). The mean precision is 0.63, which is low since the fuzzy matching uses both the phonetic and edit-distance[3] matching approaches.

Since it requires a priori knowledge of the expected matches this sample of 40 queries in a book of 30 pages is small. However, for mathematical material, there seems to be no classical sample collection available such as the ones gathered for the TREC competitions.[4]

Google Comparison Another approach to evaluate a search engine is to compare the engine with another one which can also retrieve the content. Since ActiveMath is on the Web, it can be visited by Web-robots. We used the ability of the Google search engine to restrict its search on a given Web-server to compare Google results to ours.

A Web-server needs to be linked online in order to be accessible to Web-robots. It also requires to respond appropriately to robots' requests, that is, avoid HtML-frames and cookies which are both basic ingredients of rich browser-based Web-applications. In the versions of ActiveMath that was evaluated a special access was arranged for Web-robots, listing the content items. As a result, the Google search engine could retrieve the presentation of individual content items.

We gathered another set of queries, expected to be matched in the complete content of the collection realized in LeActiveMath [LG06] and compared the number of matches. Selected results can be found in Table 2. The Student-T-test

[3] The Lucene library offers a form of fuzzy queries which applies elementary modifications to the query words and recompute the matches. They are returned with a lower score based on the *edit-distance*, the amount of elementary modifications applied. Fuzzy textual matches in the ActiveMath search tool also use these queries.

[4] The TREC competition is a yearly competition organized along the TREC conferences by NIST where large collections of texts are given to participants, followed by queries. The result is evaluated, among others for precision and recall, by NIST. See `http://trec.nist.gov/` and the description in [Mah06].

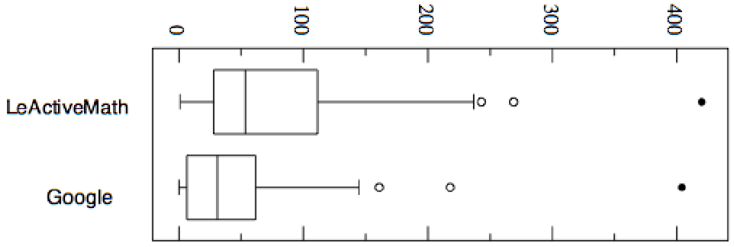


Fig. 4. The distribution graph of both ActiveMath and Google matches

comparison between the two columns indicated a t-score of 5.41 which indicates a significant difference. Distribution graphs of the number of matches are depicted in Figure 4 which indicate a broader variation of the ActiveMath search tool. One of the main differences of ActiveMath search tool compared to Google is the matches (good and bad) introduced by the fuzzy matches which the Google search engine does not provide. Two other factors led to the differences: the dates of the index construction differ; and the fact that Google searches the text of the HtML item views which means, for example, that the text of the relations from the items, which are shown along the presented item, are considered to be part of the item or that the words of a presented formula, such as the word *sin* in $\sin x$, are matched as well.

6 Related Work

A few search tools allow for the query of mathematical terms, e.g. [AGC+04] or [Urb04]. Because of the lack of a formal library, the ActiveMath search tool cannot manipulate the formulæ applying formal knowledge, for example term-ordering normalizations or symbol generalization. This results in a relatively low tolerance in formulæ search.

The search tool of the Digital Library of Mathematical Functions explained in [You05] is dedicated to functions. As a result it makes several normalizations of mathematical terms such as the conversion of $ab^{-1}cd^{-1}$ in $\frac{ac}{bd}$. Such normalizations introduce tolerance within the search tool. The usage of a simple type system for OpenMath objects could enable ActiveMath search tool to perform normalizations which would, otherwise, be abusive with mathematical objects such as group elements or matrices. The ActiveMath search tool also differentiates itself from the DLMF search tool by the user interface: while the DLMF search tool defines a plain-text input syntax, queries in the ActiveMath search tool are realized by input of concrete words, attribute-value, or formulæ.

Another avenue has been explored by Paul Cairns in [Cai04] where Latent Semantic Analysis can be used to provide a *semantic* distance between token-vectors, including mathematical terms. We started to explore LSA usage.

MathWebSearch [KS06] is a Web-crawler for documents with MathML content. This tool uses the term indexing techniques of [Gra96] to index formulæ collected on the Web. Compared to this search engine, the ActiveMath search tool is more learner-oriented but still lacks the ability to perform queries for formulæ with variables that occur several times. Since it uses the Lucene library, the ActiveMath search tool appears better scalable compared to the term indexing technique which needs an in-memory representation.

From the overall access point-of-view, the ActiveMath learning environment seems to be one of the rare mathematical content items presentation servers which manages items with a fine granularity. A few similar projects are the Thesaurus at `http://thesaurus.maths.org/`, whose goal is an international dictionary of mathematical concepts, or the encyclopedia projects. ActiveMath is the only one among them which offers search by attributes and formulæ and a few other features which are consequences of the semantic nature of the content encoding.

7 Conclusion

In this article, we described the access to mathematical content items in ActiveMath in particular, through its search tool and presented evaluations of this search tool. The main contribution of this research is the development of an analysis process for mathematical formulæ which converts them to streams of tokens on which information retrieval techniques can be applied. Using the Lucene library for the indexing matching makes the search tool efficient We also developed a dedicated ranking scheme for search results which allowed us to order most fuzzy matches below exact matches. The search interface has been designed for learners and it has been evaluated for usability. Two main conclusions can be drawn from the evaluations:
Not surprisingly, fuzziness introduces noise into the search results. Conversely, the fuzziness introduces tolerance to the queries.
Future Work The comparison with other search engines will be refined, on the one hand by enlarging the size of the sample and on the other hand, by using automated methods to construct the sample, perform the test, and to invoke queries on the Web-robot.

We are investigating the benefits of alternative indexing and matching mechanisms [Gra96] that may be more suitable for mathematical expressions.

We started working on access by Web-robots that can understand the mathematical semantic documents. They will be crawling `OMDoc` documents, or semantically rich xHtML documents in which mathematical formulæ are given in MathML whose semantic is given in parallel markup.

Acknowledgements

This publication is partly a result of work in the context of the LeActiveMath project, funded under the 6th Framework Program of the European Community – (Contract IST-2003-507826). The authors are solely responsible for its content.

References

AGC$^+$04. A. Asperti, F. Guidi, C. Sacerdoti Coen, E. Tassi, and S. Zacchiroli. A content based mathematical search engine: Whelp. In J.-C. Filliatre, C. Paulin-Mohring, and B. Werner, editors, *Proceedings of the TYPES 2004*, LNCS 3839, pages 17–32. Springer-Verlag, 2004. See also `http://www.cs.UniBo.it/helm/`.

BCC$^+$04. S. Buswell, O. Caprotti, D. Carlisle, M. Dewar, M. Gaëtano, and M. Kohlhase. The OpenMath standard, version 2.0. Technical report, The OpenMath Society, June 2004. Available at `http://www.openmath.org/`.

Cai04. P. Cairns. Informalising formal mathematics. In A. Asperti, G. Bancerek, and A. Trybulec, editors, *Proceedings of MKM 2004*, volume 3119 of *LNCS*, pages 58–72. Springer-Verlag, 2004. Available at `http://www.uclic.ucl.ac.uk/paul/research/MizarLSI.pdf`.

Gra96. P. Graf. *Term Indexing*, volume 1053 of *LNCS*. Springer-Verlag, 1996.

Koh00. M. Kohlhase. `OMDoc`: Towards an OPENMATH representation of mathematical documents. Seki Report SR-00-02, Fachbereich Informatik, Universität des Saarlandes, 2000. See also `http://www.mathweb.org/omdoc`.

KS06. M. Kohlhase. and I. Sucan, A Search Engine for Mathematical Formulæ. Proceedings of AISC 06, LNAI 4120, See also `http://kwarc.eecs.iu-bremen.de/software/mmlsearch/`.

LG06. P. Libbrecht and C. Gross. Experience report writing LeActiveMath Calculus. In Proceedings of MKM 2006, LNAI 4108, Springer Verlag,

MAB$^+$01. E. Melis, E. Andrès, J. Büdenbender, A. Frischauf, G. Goguadze, P. Libbrecht, M. Pollet, and C. Ullrich. ActiveMath: A Generic and Adaptive Web-Based Learning Environment. *International Journal of Artificial Intelligence in Education*, 12(4):385–407, 2001.

MAF$^+$03. E. Melis, J. Büdenbender E. Andrès, A. Frischauf, G. Goguadze, P. Libbrecht, M. Pollet, and C. Ullrich. Knowledge Representation and Management in ACTIVEMATH. *Annals of Mathematics and Artificial Intelligence*, 38(1-3):47–64, 2003. Volume is accessible from `http://monet.nag.co.uk/mkm/amai/index.html`.

Mah06. K. Mahesh. Text retrieval quality: A primer. Technical report, Oracle Coproration, 2006. See `http://www.oracle.com/technology/products/text/htdocs/imt_quality.htm`.

MLUM06. S. Manzoor, P. Libbrecht, C. Ullrich, and E. Melis. Authoring presentation for OpenMath. In Michael Kohlhase, editor, Proceedings of MKM 2005, volume 3863 of *LNCS*, pages 33–48, Heidelberg, 2006. Springer.

Por05. Martin Porter. The Porter stemming algorithm, 2005. See `http://www.tartarus.org/~martin/PorterStemmer/`.

Urb04. J. Urban. Momm - fast interreduction and retrieval in large libraries of formalized mathematics. In *Proceedings of the ESFOR 2004 workshop at IJCAR'04, 2004*. More information on the project at `http://wiki.mizar.org/cgi-bin/twiki/view/Mizar/MoMM`.

vR79. C.J. van Rijsbergen. *Information Retrieval*. Butterworths, 1979. Available at `http://www.dcs.gla.ac.uk/~iain/keith/`.

You05. A. Youssef. Information search and retrieval of mathematical contents: Issues and methods. In *proceedings of IASSE-2005*, Toronto, Canada, 2005.