Expanded Grammar for Detecting Equivalence in Math Expressions

Mohammed Q. Shatnawi
Department of Computer Information Systems
Jordan University of Science and Technology
Jordan

Marwan T. Alquran
Department of Mathematics & Statistics
Jordan University of Science and Technology
Jordan

Fatima M. Quiam
Department of Computer Science
Jordan University of Science and Technology
Jordan

ABSTRACT
huge amount of different types of information are being posted on the web on a daily basis; therefore, searching capabilities should be provided to help users in finding their requested information. Locating a specific type of information within large repositories of disparate data becomes difficult, if not impossible, without specialized information retrieval systems.

Traditional or text-based search engines do not achieve the level of success that users seek in retrieving structured information (e.g. mathematical information). For example, when a user searches for x(y+z) using Google, Google retrieves documents that contain xyz, x+y=z, (x+y+z) =xyz or any other document that contains x, y, and/or z, but not x(y+z) as a standalone expression. The reason behind this is that Google uses the text-based search capabilities/ Algorithms that depend, mostly, on techniques for matching and probabilities of occurrences of x, y, and z.

The major obstacle of math search in current text search systems is that those systems do not differentiate between a user query that contains a mathematical expression, and any other query that contains text terms. Therefore, those text-based search systems process mathematical expressions as other texts, regardless of its nature whether being well-structured or not.

Here in this context, the text search process will be refined to be applicable in searching for a mathematical expression by implementing a system that is responsible for detecting equivalent math expressions. In fact, more algorithms will be added to the Information Retrieval System in order to make it suitable to do search for a mathematical expression as well as other forms of text.

General Terms
Information retrieval, Math search.

Keywords
Math search, expression's equivalent forms, mathematical expression, detecting equivalency, grammar, text-based search systems.

1. INTRODUCTION
Web Information consists of two main types [1]:

- Structured Web Information, which is defined as information ordered in a particular way. Such as mathematical expression, database tables, transactions, math documents, etc.
- Unstructured Web Information; which is defined as information in random pieces. This type includes bitmap objects such as images, video or audio files, and textual objects such as text, the body of the e-mail message, Web pages, or word processor document.

This research will focus on processing mathematical content, which is an example of structured information. Math content is structured in a way that the meaning of certain math expression depends on the structure of that expression.

2. EQUIVALENCE IN MATHEMATICAL EXPRESSION
Mathematical expression can be expressed in many and sometimes, infinite number of equivalent forms. For example, 0.5 is the same as ½ mathematically, and x*y is the same as y*x. Searching for x*y using traditional search engines does not retrieve documents that contain the expression y*x, because those search engines use techniques that are not suitable to accurately locate math contents. Therefore; there rises a need for tools that help the users locate the requested math expression and all of its equivalent forms. For example, when a user searches for the expression tan(x), the search engine must retrieve the documents that contain the expression itself, and the documents that contain the expression sin(x)/cos(x), because both expressions are mathematically equivalent.

3. MATHEMATICAL EXPRESSION AS SEARCH TERMS
Mathematical expressions are a distinct type of information. Searching the Web for a mathematical expression is not a well-defined process; the result of the search is unexpected most of the time. The inaccurate result is due to the nature of the mathematical expression search process, which is not based on clear and structured rules. In addition, the available techniques are not applicable to search for such expressions but they are designed and tailored to work with normal text along with different kinds of documents (e.g. multimedia documents).

4. SEARCHING FOR MATHEMATICAL EXPRESSION USING TRADITIONAL SEARCH ENGINES
Information retrieval systems have been developed since several decades [18]. Mathematical materials such as formulas and equations are symbolic and highly structured. Current search systems do not provide the means of searching such entities or understanding math queries that contain non-alphabetic symbols. A text that is retrieved by current search systems is an unstructured text with no data type definition, and no conceptual definitions as well. Mathematical expressions are well-structured, and the structure conveys their correct interpretation. This is an important reason why current search engines fail in retrieving items that contain mathematical expressions [2].

The same mathematical expression can be represented in many equivalent ways. For this reason, it is not effective to use a thesaurus (i.e. a finite set of concrete per-term definitions) structure in searching for all equivalent expressions. If the current search engines are enhanced to retrieve a specific type of a mathematical expression, they will still fail in retrieving the documents containing equivalent forms of that expression. In the text-based search systems, equivalence is determined to synonyms supported by thesaurus. Mathematical equivalence is infinite for each term in traditional thesauri. This is another important reason that makes current search engines fail in retrieving mathematical expressions. Thus, this confirms the highly demand of retrieving the documents that contain not only the expression itself, but also the expression's equivalent forms [2].

Mathematics has developed many of non-alphanumerical symbols. Large number of these symbols is undefined or misinterpreted by text search systems. Mathematical expressions have complex structures (numerator/ denominator, subscripts/superscripts, summation, integration limit…etc). When mathematical expressions or equations are retrieved by text-based search systems, they are handled as a linear contiguity structure without separation into meaningful parts. This adds another obstacle for the failure of the current search engines in retrieving math documents.

Capabilities of indexing and searching in text-based search engines do not work properly for math search system; therefore searching and indexing for mathematical equations require further researches.

5. MOTIVATION FOR MATH SEARCH

Nowadays, mathematical expressions are used in different areas (e.g. science, engineering, and education). Therefore, it is expected that many people will be interested in math search. Also, while such a huge amount of information is posted on the Web, there should be a reliable search system that is capable of searching accurately among this information for a specific user request.

Usually, the user's query does not match the database content. Therefore, it is difficult to retrieve documents that contain terms different from the user's query, even though they are semantically the same. The online-reasoning systems can, in theory, be used to check for equivalence between query expressions and the content expressions. Those systems would take prohibitively long time to check if a query expression is equivalent (or not) to the expressions in the searchable contents.

6. ACCESSING MATH EXPRESSIONS ON THE WEB

Virtually all searches are text-based [3, 5], thus, there are problems associated with accessing math expressions on the Web. Those problems can be summarized as follows:

- Unless there is an agreed upon technique that understood by both, users and search engines, the user still needs to know the best search terms and the best way to write a query to be used in searching for any mathematical expression.
- When a user searches for a mathematical expression, there would be non-alphabetical symbols that are not understood by current search engines (e.g. Log10x+y^2).
- The same expression can be rewritten in many different but equivalent ways (e.g. 1/x and x^-1).
- Text-based search engines do not consider the syntax of a mathematical expression as one of its main features.
- The way that is used to search for equivalent text terms (i.e. thesaurus to search for synonyms) is not feasible for searching for an equivalent mathematical expression.

7. EQUIVALENCE AND INCONSISTENCY

One major problem in being able to retrieve relevant items is the inconsistency between the author's vocabulary and the user's vocabulary. Therefore, the user may search for a term the author does not provide. This problem has been studied in text search and there are some proposed solutions, such as searching for the synonyms during the search process using thesaurus lookup. A similar problem arises when you search for a mathematical expression because the term y+x is the same as x+y mathematically. Even if the current search engines are equipped with tools to enhance their ability in retrieving items that contain a certain type of a mathematical expression, they will still fail in retrieving the documents that contain variants of that mathematical expression. Therefore, there is a need for a way to retrieve the documents that contain the expression itself and all of its equivalent forms.

7.1 Syntax Interpretation

A word in an unstructured text is simply a word with no data type definition and no conceptual definition. This explains why current search engines fail in retrieving mathematical expressions as they do not understand mathematical structures, but the current search engines well-understand text.

Mathematical expressions are well structured and the structure itself holds their correct interpretations. For example, math holds a difference between 2*(x2-x3) and 2*x2-x3. However, if we were doing text retrieval there would be no difference between both expressions [19].

8. RESEARCH OBJECTIVES

There are some techniques that have been applied in searching for a mathematical expression. Unfortunately, those techniques do not take into consideration the properties of the mathematical expressions, which leaves math contents apart from all other types of information.

The objective of this research is to design and implement an effective and reliable technique that transforms a user input expression into a unique normalized form. This form will be used in searching for a mathematical expression in a way that takes into consideration its unique properties. The way that expressions are stored in the searchable database must be compliant with the way normalized expressions are interpreted. Eventually, the expression, and all of its equivalent forms are transformed into one a unique form that is called the normalized form. The system results in a performance enhancement of the precision when searching for math expressions.

9. DEFINITION OF MAPPING

Mapping is a process of transforming a mathematical expression from an original form into an equivalent one [5]. Mapping can be divided into two categories; algebraic Mapping and structural Mapping. In algebraic Mapping, the work is on the expression in its algebraic form. Therefore, there will be a difference in the form of algebraic mathematical expression before and after performing the Mapping process. For example, the expression c*b*a will be normalized into a*b*c, the labels of the expression parse tree nodes will be changed, maintaining the parse tree's structure unchanged [5].

In structural Mapping, the expression's parse tree structure [20] will be changed after Mapping. For example, the parse tree for the expression (x+y) +z before the Mapping is illustrated in Figure 1.

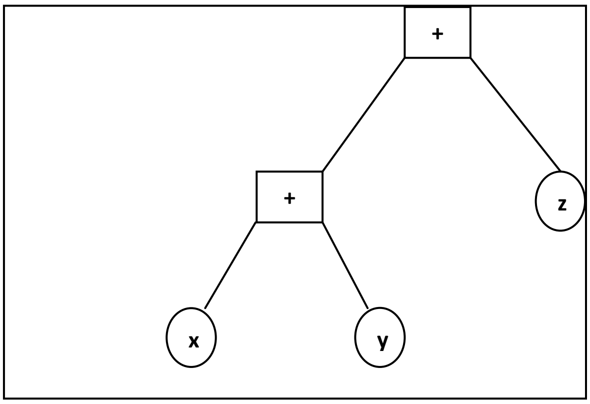

Fig 1: Parse tree for expression (x+y)+z (before mapping)

The parse tree for the expression x+(y+z) after mapping has changed as shown in Figure 2.

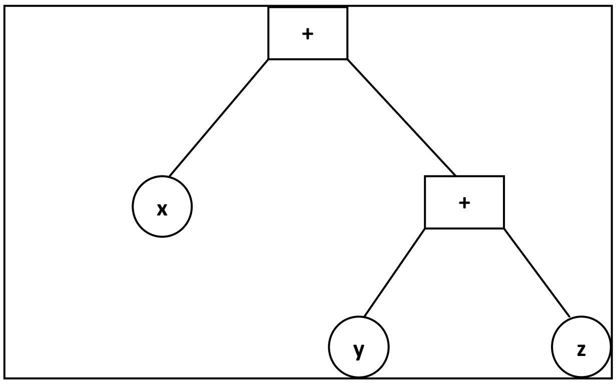

Fig 2: Parse tree for expression x+(y+z) (after mapping)

The Mapping process ends up with a unique normalized form, which is the kind where all equivalent expressions have the same representation. Thus, the Mapping process maps different math expressions' representation into a unique form.

10. PROBLEM STATEMENT

One of the major obstacles in math search systems is that a mathematical expression can be represented in different yet equivalent forms. Mapping different representation into a unique form is a way to solve this problem [5]. This common form will be compared against the searched database, which contains the normalized form of that expression as well.

In math search systems, the normalized form will be used in searching for a mathematical expression in a way that takes into account the properties of the mathematical expressions. Recently, work has begun to develop techniques and algorithm for detecting equivalency of math expressions [5]. This work has been developed only for detecting equivalency in math expressions that contain the basic mathematical operations (e.g. addition, subtraction, multiplication, division, and the exponential operation). Therefore, it is needed to add more mathematical operations, in which detecting equivalency is generalized to be valid for most of mathematical operations. For example, the logarithmic operation, limit operation, polynomial operation, and trigonometric operation were added and the necessary algorithms for detecting equivalency have been developed in this research.

The main problem in measuring the performance of math search systems is the lack of any math query benchmark. Consequently, measuring the enhancement achieved by the Mapping process is mandatory.

11. SIGNIFICANT CONTRIBUTIONS

This research makes three significant contributions in the field of math search.

- Introduction of a new approach for addressing the math equivalence and detection techniques.
- Development of a completely new method to discover different equivalent math expressions and map all of them into one normalized form.
- Quantification of the performance improvement in terms of precision and/or recall of each added equivalence rule.

12. RELATED WORK

There are different available math systems. In this research paper, only a few of them will be mentioned.

12.1 Digital Library of Mathematical Function (DLMF)

Digital Library of Mathematical Functions (DLMF) was developed by the National Institute for Standards and Technology (NIST). The general approach in DLMF is an evolutionary one; as it increases the existing text search engines with all mathematical searching capabilities. DLMF is an online project, which is intended as an update of Abramowitz and Stegun's Handbook of Mathematical Functions.

Two phases are needed to build up the DLMF. The first one is to find a text search engine, which has collections of characteristics and capabilities as mentioned in [6], in which they are:

1- Boolean and proximity operators.
2- Fields and fields-based search.
3- Thesaurus support "dictionary of meaning".
4- Allowance for surrogate files; view document as a pair of files.

The second phase is to implement the actual increase that needs the following tasks as mentioned in [6]:

1- Textualization to all mathematical symbols in a user query and in content.
2- Flattening to mathematical structures and equations in contents and in queries.
3- Mapping to convert the linear form of each expression/ formula into a normal form.
4- Development of a query language to express math queries as well as text queries.
5- Math query language must be flexible, easy-to-use and more expressive.
6- Math query parser/translator must be a frontend layer to do the parsing and translate formulas queries to textual and numeric queries using Boolean and Proximity operators.
7- Contents transformation module is to transform the mathematical content in DLMF database (equations, formulas and mathematical symbols) into a textual-numerical normalized form.
8- Surrogate-files generator, the surrogate file contains the transformed math content as well as metadata.

9- Thesaurus contains the entries for generic math, markup related and DLMF-specific entries.

DLMF did some kind of Mapping by converting the linear form of each expression/ formula to a normal one. For example, a Mapping algorithm that sorts the parse tree by defining a new data model called the sorted parse tree normal form. This Mapping is achieved by sorting the parse tree that is not similar to our approach, which is more general and dynamic as well.

12.2 System for Encoding Math Expression on the Web

There are several systems for encoding math expression on the Web, such as, MathML, OpenMath, and the OMDOC systems.

Mathematical Markup Languages (MathML) is an XML-Based Languages. It is found to describe and represent mathematical notations, in addition to capturing the content and structure of mathematical notations. MathML works by enabling mathematical data to be received, served, and processed on the Web.

MathML plays an important role in enhancing the math search process. In MathML, the user types a math equation bringing to retrieve a list of documents in which it occurs. By this way, MathML may have a role of enhancing existing search systems scoffed toward bibliographic metadata [7].

OpenMath is a Language that is used to represent mathematical notations and their semantic. It makes the mathematical objects to be commutative between the computer programs stored in database or published on the web. It is constrained on the representation of information and is not accountable to be used directly for presentation.

The original motivation for OpenMath came from the Computer Algebra community. Computer Algebra packages were getting bigger and more unwieldy and they seemed reasonable to adopt a generic "plug and play" architecture to allow specialized programs to be used for general purpose environments [8].

OMDOC is a markup formats that are found to be used as semantics-oriented representation format and ontology language for mathematical knowledge. OMDOC is an extension of the Content MathML and OpenMath formats. It has been found to extend these formats by markup for the document and theory level of mathematical documents, so it can be specified by the author for that document, and the reader for OMDOC or a mathematical software system works to be exploited by that mathematical document [9].

OMDOC can be used as a communication standard between mechanized reasoning systems, and it can also be used as input, and without forgetting its use as documentation preparation language for MBase.

None of the above mentioned systems deal with the Mapping in the way this research is pursuing are concerned with. But in more canonical form of content MathML; content MathML uses different equivalent expression encoding for the same expression. For example, Inverse hyperbolic and trigonometric function have two equivalent representations arcsin(x) and sin−1(x). The two representations are permitted by Content MathML, because some documents will find sin − 1(x) as a description of arcsin(x). Consequently, this equivalence issue cannot be solved by converting instances of one encoding into the other without changing the meaning of thebauthor. In addition, the user should be given the ability to use both formats when searching [10].

There are different systems that can be organized into different categories. For example:

- Computer algebra systems (e.g. Mathematica, MAPLE).
- Systems to add interactive mathematical tools on the Web (e.g. MathType, MathEdit, WebMathematica).
- Systems that support mathematical knowledge base (MKB) (e.g. MBase System).
- Math Web learning systems (e.g. ActiveMath) and Mathematical library (e.g. HELM, MIZAR).

Computer Algebra Systems do not deal with mapping in the way this research is doing. In Maple, finding a normal form for a symbolic expression is very important to decide whether two expressions are the same or not [11]. The Mapping of rational expressions clears all common factors from the numerator and the denominator. For example the expression (x+1/x) is normalized to ((x^2+1)/x). Systems to Add Interactive Mathematical Tools on the Web do not consider the problem of mapping which is the core of this research. MIZAR also does not deal with Mapping in the way this research is doing. The MIZAR's normalization is used only in structuring the documents not in structuring the mathematical expressions as this research is doing in the Mapping process.

13. THE RESEARCH'S ASSUMPTIONS

In this research, the following assumptions are taken into consideration: First, we did not deal with wildcard-like queries; therefore, a+b matching x+y is outside the scope of our research. Actually, it is the burden of the user to use wildcards for non-literal matching.

Second, standard math equivalences are assumed to be wanted by general users to be detected. Otherwise, the users may be provided by a GUI so they can choose whether to accept equivalence detection or not. Based on that, the equivalence detection may be applied by default or the users can have a choice over that.

The third assumption is that the math expressions are assumed to be recognizable from broader text, since math contents are now being converted to XML-formatted documents (e.g. MathML), where expressions are clearly marked up.

The last assumption is that after doing the required Mapping, the search ultimately is on serialized normalized trees.

14. CURRENT SEARCH ENGINES AND MATH SEARCH ISSUES

Text-based search engines cannot search efficiently for different types of mathematical constructs (e.g. axioms, formulas… etc.). Mathematical expressions have some distinct properties that make current search engines inadequate to search for such expressions. There are issues the current search environment has never had to face. Three of them will be mentioned according to [6, 12].

- Searching for a mathematical expression is usually combined with non alphabetical symbols (e.g. x^3 dy/dx, x**2, etc).
- Different types of mathematical expressions are structured and the structure itself conveys the meaning of these expressions.
- The more challenging issue is that the same expression can be represented in many different ways. For example, 1/3 mathematically is the same as 3^-1.

This research will try to solve the third issue by mapping different ways of representing a specific expression into a unique normalized form.

15. EQUIVALENCE DETECTION AND MAPPING

The equivalence detection and Mapping (EDM) is the most important part of this work. As a matter of fact, it is the core of our research. The EDN aims to transform the expression tree into a normalized tree. The normalized tree is equivalent to the original tree, yet it is an agreed upon representation based on some rules to facilitate the search process. Therefore, the normalized tree should be the common form between the searchable database and the math query term. This normalized unique form will be used during the search process thereafter.

16. GRAMMAR OF EQUIVALENCE RULES

Grammar of Equivalence Rules (GER) is a part of the developed system that does the Mapping based on a context free grammar (CFG). With GER, the Mapping system is not based on fixed rules of Mapping, but is based on a well-defined grammar. By enabling the system administrator to add a very large set of equivalence rules that comply with the predefined grammar within the GER subsystem, the GER will turn into a more general and reliable system of Mapping. Table 1 shows some of those equivalence rules for which the system administrator can add to the system.

In this context grammar is used to generate an infinite set of valid mathematical equivalence rules (*e.g.* x^-2 mathematically equivalent to 1/x^2). The grammar will impose a desired structure on the equivalence rules. The system administrator should follow this structure in order to add a valid mathematical equivalence rule to the GER component.

16.1 Syntax of GER Rules

The colon ":" has been chosen as the equivalent operator between the two equivalent expressions. The basic syntax for the rules in GER would be in the form of:

E: E (E is a non- terminal symbol, which represents a mathematical expression)

The left hand side of the ":" operator is the expression before applying rules of equivalence, while the right hand side of the ":" operator is the expression after applying rules of equivalence. Examples of those rules are shown in Table 1.

Table 1: An example set of equivalent rules

Math Expression	Equivalent Expression	Rule's Format in GER
x^-(y/g)	1/x^(y/g)	$1^-(#2):1/$1^#2
X*y^(-n)	x/y^n	$1*$2^(-$3):$1/$2^$3
tan(x)	sin(x)/cos(x)	tan($1):sin($1) /cos($1)
$\log_e$ (a)	ln(a)	$\log_e$ ($1) : ln($1)

17. EXPANDED GER'S GRAMMAR (EGER): FORMAL DEFINITION

The grammar that is used in this research is a Context Free Grammar [13]. Every production rule in the CFG is in the form V → w, where V is non-terminal symbol and w is a string consisting of terminals and/or non-terminals.

The grammar G in this research is a quadruple (T, N, S, R), where:

T is a finite set of terminal symbols,
N is a finite set of non-terminal symbols, S is a unique starting symbol.
R is a finite set of productions of the form α → β, where α and β are strings of non-terminals and terminals.

The following grammar is built to form the base of the Mapping system.. The '|' notational shorthand, which can be read as "or", is used to represent multiple production rules within a single line:

This grammar in Figure 3 is an expansion for the grammar that the authors present in [5].

G= {T, N, S, R,}
T = {0, 1, 2, -----, 9,-1, $, -
$, #, π} N ={S, A, T,C, P,
E, F, B , D, Z}
R ={ S→ A: A
A → A+T | A-T | T
T → T*C | T/C | T^C | T@C | C
C → ln P | abs P | fac P | sum P | log P| $\log_2$ P|
$\log_e$ P |
dif P| I(A,A):P | lim P | P
P → cos E | $\cos^2$ E | acos E | sin E | $\sin^2$ E |
asin E |
tan E | $\tan^2$ E | atan E |
sec E | $\sec^2$ E | csc E | $\csc^2$ E | cot E |
$\cot^2$ E | E E → e | F
F → (A) | B
B→ $D | -$D | # D| 0| 1 | -1 (i.e. D stands
for digit) D→ 0, 1, 2, ..., 9, π | DD
Z→ +ZR | -ZR | *ZR | /ZR | ^ ZR | @ ZR|
(Z) | R | RR
R→ B }

Fig 3: The Grammar

There are few abbreviations that are used in the previous grammar:

T@C mathematically means $\sqrt[C]{T}$
abs P mathematically means |P|
fac P mathematically means P!
sum P mathematically means $\sum^{\infty}$
dif P mathematically means $P^{\backslash}$
I(A,A):P mathematically means $_A\int^A P$
log P mathematically means $\log_{10}P$
asin mathematically means $\sin^{-1}$
acos mathematically means $\cos^{-1}$
atan mathematically means $\tan^{-1}$
P lim P mathematically means $\lim_{i\to\infty} P$

18. GENERIC MAPPING

The Mapping system that is built based on GER is termed Generic Mapping (GM). Based on GER, the system administrator should be able to add any valid mathematical equivalence rules. The Mapping system should be able to detect equivalence for those added rules thereafter. Besides, the areas of math that our system has provided equivalence detection for must be determined. In addition, the groups of users whom the system has targeted must be identified. In this research, algorithms that detect equivalence for any added rule that conforms to the grammar have been developed. Besides, any added rule to the generic Mapping system is derived from a general principle in which a rule is admissible, if and only if. There is a corresponding transformation on the parse-tree [14].

The GM processes a massive amount of math content. Thus, there are difficulties associated with searching such content using current search engines as mentioned before. Consequently, this research adopts the concept and properties of digital ecosystems [21, 22] trying to enhance the ability of GM system in increasing the precision and/or recall when searching math content. Accordingly, the GN system has been developed to be:

- Able to be incorporated in different environments, (i.e. web-based systems, math-search systems, etc.)
- Designed as separate component that can cooperate efficiently with other ecosystems.
- Flexible in which a user can chose to whether apply the GM or not.
- Scalable in which the GN can be easily expanded to include all related math content.

18.1 Rule Validation

A Validator is a component of the Mapping system. The Validator is responsible for validating the correctness of any equivalence rules that is added by a system administrator.
The Validator verifies if the added rule is compliant with the GER grammar format. This validation process is done using one of compiler compiler packages such as javaCC [15].

19. PERFORMANCE ANALYSIS

Measuring the performance of any newly developed system is required to evaluate its effectiveness and to compare it with other systems.

The major problem in measuring the performance of math search systems is the lack of any math query benchmark; because this area is relatively new. In the absence of an agreed upon query benchmark, the performance of the Mapping system is based on the searchable database content. Therefore, the result of a certain search using the same set of Mapping rules on two different database contents results in two different outcomes.

The main goal of the Mapping system is to increase the number of true hits when a user searches for a math expression. Therefore, after applying a set of Mapping rules on both; database content and a user math query, the process will result in new math expressions, which will not be founded without applying that set of Mapping rules. The following examples will clarify the above concepts.

Suppose the database contains (x * sec(y) + ln (z)) expression, and the user searches for (x * (1/ cos(y)) + $\log_e$ (z)) expression, or the user searches for a part of this mathematical expression (i.e. (1/ cos(y)) or ($\log_e$ (z))). In this case, there are two approaches that can be followed to retrieve the content:

1- Without Mapping : the search does not retrieve the mathematical expression (x * sec(y) + ln (z)), because this expression does not match the user request expression (x * (1/ cos(y)) + $\log_e$ (z)).
2- With Mapping: there are two Mapping rules that are accepted and added to a list of Mapping rules. These Mapping rules are:
   - sec (x) : 1/ cos (x)
   - ln (x) : $\log_e$ (x)

After applying the above Mapping rules, the database content and the user query are normalized accordingly. Therefore, the database content (x * sec(y) + ln (z)) will be normalized to (x * (1/ cos(y)) + $\log_e$ (z)). So the search retrieves the user's request as the database contains the expression after it has been normalized, which was not possible without normalizing the database content.
The above example is used to clarify the goal of Mapping. In a case of a complete database and enough Mapping rules, the number of relevant retrieved items will be increased (i.e. precision is increased). With Mapping, some of database items will be retrieved, and then the relevant item will be increased. Some of the items that would not be retrieved without Mapping, Mapping increases the chance for such those items to be retrieved, therefore, the recall will be effected positively as well. By increasing the number of retrieved relevant items and decreasing the retrieved items, the number of user query hits increases.

20. FURTHER GRAMMAR EXPANSION

According to the above, this research is good in terms of enhancing the mathematical expression web search process. This way of enhancing is done by using the proposed Mapping system that is based on GER. The GN system can search massive amount of math content and cooperate, efficiently, with other related ecosystems trying to achieve the user's goal, when he/she searches math-based content.
Math content is represented in different formats, which make it more difficult to design a specialized system to search this content. Therefore, adopting the concept of digital ecosystems to standardize the way a math expression is represented minimizes the difficulties of searching such content.
A simple experiment has been performed using four datasets. Tthe first one contains 10 expressions, the second one contains 20 expressions, the third one contains 30 expressions and the fourth one contains 4o expressions. The query is tan(x)+ ln(a)+ln(b). The following equivalence rules can be applied on the above expression:

- loge($1):ln($1)
- ln($1)+ln($2):ln($1*$2)
- sec($1)/csc($1):tan($1)
- tan($1):sin($1)/cos($1)

The enhancement of the precision can be measured after applying a rule, two rules, three rules, and eventually, four rules. Figure 4 shows the precision enhancement that has been achieved by applying the mapping process.
Figure 4 shows precision values of searching for query tan(x)+ln(a)+ln(b), precision without Mapping is 0.5. After applying the first rule loge($1):ln($1) the items containing the equivalent mathematical expressions (tan(x)+loge(a)+loge(b),

tan(x)+loge(a)+ln(b) and tan(x)+ln(a)+loge(b)) are retrieved, and the items that are retrieved relevant are also increased. Consequently, the precision has increased to 0.66. After applying the second rule ln($1)+ln($2):ln($1*$2) the items containing the equivalent mathematical expressions (tan(x)+ln(a*b)) are retrieved, and the number of items that retrieved relevant are increase. Consequently the precision has increased to 0.77 and so on so forth. /

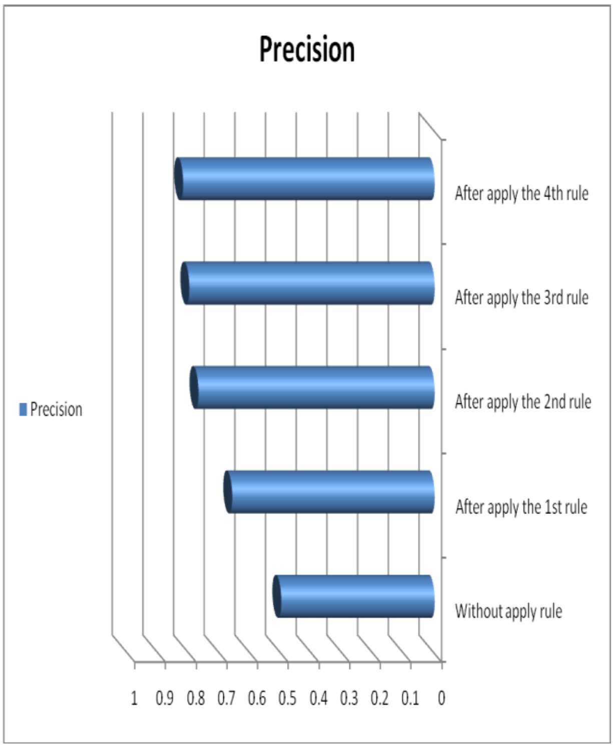

Fig 4: Quantification of each added equivalence rule for (tan(x)+ln(a)+ln(b))

Figure 5 and Figure 6 show the result of the precision for each added equivalence rule. As has been illustrated above, the more equivalence rules are applied, the more items containing equivalent expressions are retrieved. The numbers of relevant items retrieved also increases. Consequently the precision value increases as well.

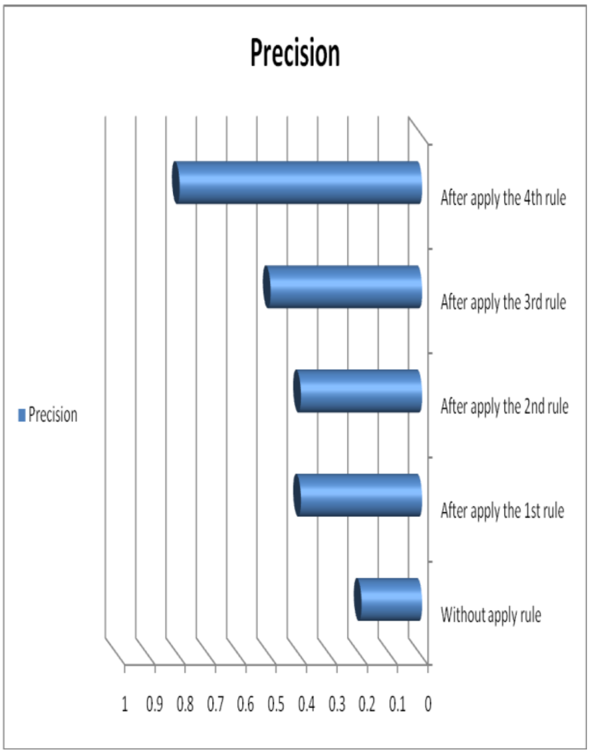

Fig 5: Quantification of each added equivalence rule for (tan(r)+loge(b))

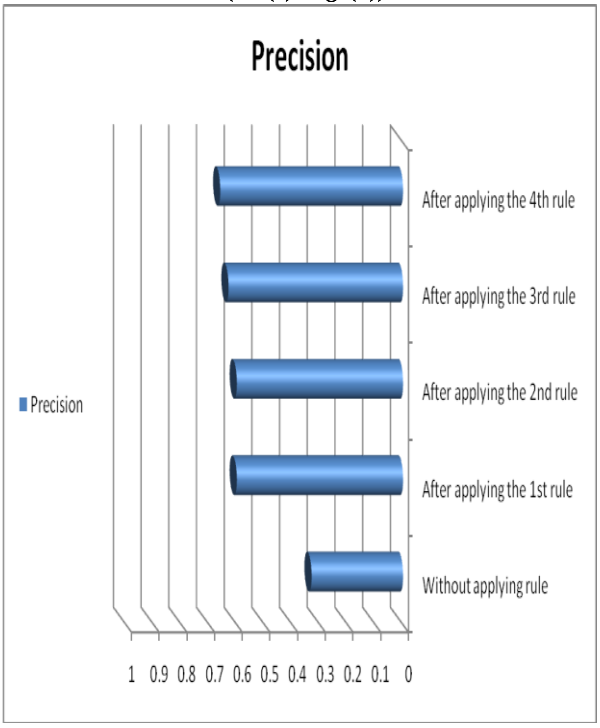

Fig 6: Quantification of each added equivalence rule for (tan(s)-ln(e))

Based on the previous experiment, the enhancement is achieved by increasing the number of relevant equivalence rules. Therefore, the more added relevant equivalence rule, the more precision occurs. Consequently, more and more equivalence rules are needed to enhance the performance of math search system.

21. FUTURE DIRECTION

Nevertheless, much work remains to be done. The following is a list that contains possible directions of future work:

- Measuring the improvement in recall (and precision) which Mapping brings to math. This can be done once digital libraries of mathematics (e.g. the DLMF of NIST [6,17]) become available and "standard" benchmark mathematical queries have been developed and accepted.
- Testing on human subjects in various science/ math communities and at various professional levels which equivalence rules are helpful and which would be confusing.
- Measuring the relation between Mapping and relevance ranking.
- Relevance ranking can be adjusted to reflect:
    - How widely recognized is the equivalence rule that caused the matching and
    - The profile of the users.

The Grammar of Equivalence Rule (GER) that presented in this research can be used to represent most of the valid equivalent mathematical expressions; it does not handle data types (e.g. complex numbers). Adding more equivalence rules for handling data types would be a useful expansion of the work.

22. REFERENCES

[1] Weglarz G. Two Worlds of Data – Unstructured and Structured, DM REVIEW MAGAZINE. http://www.dmreview.com/article. [accessed, March 2012].

[2] Kowalski, Gerald J., Maybury, Mark T. "Information Storage and Retrieval Systems: Theory and Implementation", Springer, 2nd edition, 2000.

[3] Adeel M , Cheung H S and Khiyal S. "MATH GO! PROTOTYPE OF ACONTENT BASED MATHEMATICAL FORMULA SEARCH ENGINE". Journal of Theoretical and Applied Information Technology, 2008.

[4] Sergey Brin and Lawrence Page, "The Anatomy of a Large- Scale Hypertextual Web Search Engine", Proceedings of the 7th international conference on World Wide Web, Brisbane,Australia, 1998

[5] Mohammed Shatanwi, Abdou Youssef, "Equivalence Detection Using Parse-tree Normalization for Math Search ", ICDIM 2007 Lyon-France Oct 28-31-2007.

[6] Earl W. Swokowski, Michael Olinick, Dennis Pence "Calculus", Pws Pub. Co., Sixth edition, 1994.

[7] Derivations and Parse Trees, http://www.cs.nuim.ie/~jpower/Courses/parsing/node24.html , [accessed February 2012]

[8] Miller B and Youssef A. Technical aspects of the Digital Library of Mathematical Functions. In: Annals of Mathematics and Artificial Intelligence; 2003; 38,121-136.

[9] World Wide Web Consortium. Mathematical Markup Language (MathML), http://www.w3.org/TR/MathML2.html Version2.0. [accessed, January 2012]

[10] The OpenMath Society. OpenMath, http://www.openmath.org/society/index.html [accessed March 2012]

[11] Kohlhase M. OMDoc: An Open Markup Format for Mathematical Documents, http://www.mathweb.org/omdoc/pubs/omdoc1.2.pdf [accessed December 2011]

[12] Altamimi M and Youssef A. A more canonical form of content MathML to facilitate math search. The 2007 Extreme Markup Languages conference; 2007 Aug 7-10; Montréal, Canada.

[13] Heck A. "Introduction to MAPLE", 3rd Ed. New York: Springer-Verlag; 1996.

[14] Youssef, A. "Information Search And Retrieval of Mathematics Contents: Issues and Methods", The proceeding of the ISCA 14th International Conference on Intelligent and Adaptive Sys- tems and Software Engineering (IASSE-2005), July 20-22,2005, Toronto, Canada.

[15] Thomas W. Parsons, "Introduction to Compiler Construction", Computer Science Press, W.H. Freeman and Company, 1992.

[16] Christopher W. Fraser, Robert R. Henry, Todd A. Proebsting, "BURG -- Fast Optimal Instruction Selection and Tree Parsing", December 1991.

[17] H. Boley and E. Chang, "Digital ecosystems: Principles and semantics," in Proceedings of the Inaugural IEEE International Conference on Digital Ecosystems and Technologies, 2007, pp. 398–403.

[18] P. Dini, N. Rathbone, M. Vidal, P. Hernandez, P. Ferronato, G.Briscoe, and S. Hendryx, "The digital ecosystems research vision:2010 and beyond," European Commission, Tech. Rep., 2005.

[19] Java Compiler Compiler, https://javacc.dev.java.net/ [accessed, September 2011]

[20] Lozier, D. W., Miller, B.R., and Saunders, B.V., " Design of a Digital Mathematical Library for Science, Technology and Education". Proceeding of the IEEE Forum on Research and Technology Advances in Digital Libraries; IEEE ADL '99, Bal- timore, Maryland, May 1999.