A Mathematical Formula Matching Algorithm Based on MathML

Yuping Qin [1]
College of Engineering
Bohai University
Jinzhou, China
qlq888888@sina.com

Aihua Zhang [3]
College of Engineering
Bohai University
Jinzhou, China
Jsxinxi_zah@163.com

Junnan Guo [2]
College of Mathematics and Physics
Bohai University
Jinzhou, China
1042406104@qq.com

Abstract—A mathematical formula matching algorithm is proposed. Firstly, this paper creates the tree presentation of the mathematical formula by its MathML presentation markup, normalizes the tree structure by rule base, and then horizontally traverses the tree to normalize the variable names to get the structure code. To match two mathematical formulas, this paper compares the structure codes, preorders traversal sequences and postorder traversal sequences in turn, if they all are equal respectively, then the two mathematical formulas are matching. The experimental results show that the algorithm is not only suitable for the matching that the mathematical formulas are same structure, but also for the matching that the mathematical formulas are same semantic, and the accuracy is high. Therefore, it is a more practical algorithm.

Keywords-Mathematical Formula; Matching; MathML; tree; structure code

I. INTRODUCTION

In order to prevent the academic plagiarism, online detection technology has become a hot issue in the field of information retrieval and some results[1]-[5] have been achieved. However, these results are only suitable for text detection. In an academic paper, especially a science and engineering academic paper, the main ideas are often described by mathematical formulas. Therefore, how to effectively detect the mathematical formulas has attracted increasing attention. Many organizations or scholars have carried out related researches and have achieved some results on the recognition of mathematical expressions [6]-[10], but the study on matching of mathematical expressions is still in the stage of exploration[11]-[13].

MathML (Mathematical Markup Language), based on the XML standard, is a markup language of describing mathematical formula, not only realizes the establishment and transmission of mathematical formula on the internet, but also realizes reuse and conversion in other application programs. Therefore, more and more mathematical formulas described by MathML appear on the internet.

MathML provides two kinds of description markup, one is presentation markup, and the other is content markup. Both markups can completely describe any a mathematical formula. The presentation markup of a mathematical formula is encoded according to the symbols written order in the mathematical formula, the code are simple and intuitive. In addition, the content markup code can be converted into presentation markup code[14]. Therefore, a mathematical formula matching algorithm based on MathML is proposed. The algorithm realizes accuracy matching of mathematical formulas, and the effectiveness of the algorithm is verified by the experiments.

The rest of paper is organized as follows. Section 2 gives the tree description of mathematical formula. Section 3 describes the mathematical formula matching algorithm in detail. Experimental results for different modification way are presented in section 4. Conclusion is outlined in section 5.

II. THE TREE REPRESENTATION OF MATHEMATICAL FORMULA

A. Construction of tree representation

The presentation markup code has noticeable structural feature. A mathematical formula is divided into multiple sub-expressions, each sub-expression is divided into many smaller ones, repeating the operation until the sub-expression is indivisible. The indivisible sub-expressions are called formula elements.

In presentation markup code, the mathematical formula, each sub-expressions and each format control all are marked by beginning markup and ending markup. Therefore, the markup code of a mathematical formula is clear nested layout, and the order is corresponding to the order that formula elements appear in the mathematical formula. The tree presentation of a mathematical formula can be created according to the nested structure feather. The data structure of the tree is given in Tab 1.

TABLE I. DATA STRUCTURE OF TREE

Member	Data type	Meaning
Text	string	formula element, markup
Attribute	string	MRK (markup), OPD (operand), VAR (variable), CON (constant), CHR(boundary or delimiter)

Reading the presentation markup code in written order, , and using recursion approach to create the tree representation. Root is created firstly, and then creating the sub-trees of root from left to right in turn. In the process of creating tree, if the processed code is ending markup, then reading next code, if the processed code is the beginning markup of operand, then creating a node, the text is the marked content, the attribute is OPR, if the processed code is the beginning markup of boundary or delimiter, then creating a node, the text is the marked content, the attribute is CHR, if the processed code is the beginning markup of variable, then creating a node, the text is the marked content, the attribute is VAR, if the processed code is the beginning markup of constant, then creating a node, the text is the marked content, the attribute is CON, if the processed code is other beginning markup, but does not belong to above, then creating a node, the text is the beginning markup, the attribute is MRK.

By the process of creating tree representation, the constant and variable is leaf node, function and operand are non leaf node. The lower the priority of operand and function is, the closer the corresponding node to the root node is.

For example, for $f = \frac{a + \sqrt{(b+c)(b-c)}}{2b}$,

Fig. 1 shows its presentation markup code, Fig. 2 shows the tree presentation that created according to Fig. 1.

```
<math>
  <mi>f</mi>
  <mo>=</mo>
  <mfrac>
    <mrow>
      <mi>a</mi> <mo>+</mo>
      <msqrt>
        <mrow>
          <mo>(</mo> <mi>b</mi>
          <mo>+</mo> <mi>c</mi>
          <mo>)</mo> <mo>(</mo>
          <mi>b</mi> <mo>-</mo>
          <mi>c</mi> <mo>)</mo>
        </mrow>
      </msqrt>
    </mrow>
    <mrow>
      <mn>2</mn> <mi>b</mi>
    </mrow>
  </mfrac>
```

FIGURE 1 Presentation markup code

Figure 1. Presentation on markup code

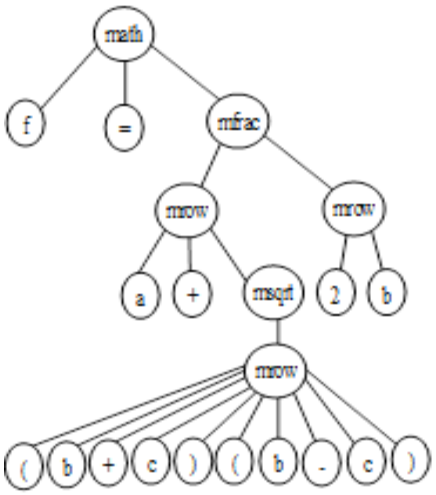

Figure 2. Tree representation

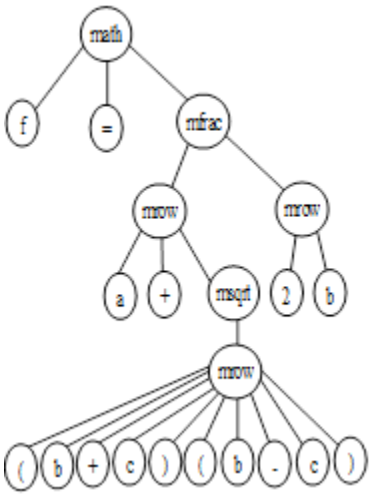

Figure 3. Normalized Figure 2

B. *Normalization processing*

MathML provides presentation markup and content markup. If the mathematical formula is described by content markup, then the content markup must be converted into the presentation markup. The method of conversion is from structure to structure, from element to element, from attribute to attribute and from element of the text[14].

The structures of the same meaning mathematical formula may be different, and the corresponding tree structures are different also. Therefore, the tree structures must be normalized. The method of normalizing tree structures is that establish a rule base, and every rule is tree pair $R(LT, RT)$. For the sub-tree T_i of in the tree T, matching T_i with LT in turn, if the matching is successful, then replacing T_i with RT, and modifying the parameters of sub-tree RT according to T_i at the same time. The transformation for T_i is completed. When transformations for all sub-trees are completed, the tree structure is normalized.

Variable name has nothing to do with the meaning of mathematical formula. To get the unique traversal

sequence according to given traversal way, the variable names must be normalized. The method of normalizing variable names is level traversal tree, if the attribute of the current node text field is VAR, replace the current node text with the identifier in the given variable name sequence.

For example, if there is rule $R(LT,RT)$ in the rule base, where LT is the tree corresponds to $(x+y)(x-y)$, RT is the tree corresponds to x^2-y^2, then the normalizing result for Fig. 2 is given in Fig. 3.

III. MATHEMATICAL FORMULA MATCHING ALGORITHM

For two mathematical formulas described by presentation markup, creating the tree presentation, normalizing tree structures and variable names. If the two trees have the same structure and the corresponding node values are equal respectively also, then the two mathematical formulas are matching. Therefore, the tree structure is used to determine if two mathematical formulas are matching firstly. The tree structure can be presented approximately by structure code.

If a mathematical formula is presented by a tree, and the depth of the tree is H, the number of nodes in ith layer is N_i, then the structure code of the mathematical formula is defined as follows:

$$SC = str(N_2) + str(N_3) + \cdots + str(N_H) \quad (1)$$

Where, $str(N_i)$ is a numerical string corresponds to N_i.

If the structure codes of two mathematical formulas are equal, then comparing the value of the corresponding node respectively. The method is that compare the preorder traversal sequences and postorder traversal sequences of the two trees respectively. If the preorder traversal sequence and postorder traversal sequence are equal respectively, then the two mathematical formulas are matching, or otherwise they are mismatching. This is because that a tree can be converted into a binary tree, and preorder traversing a tree is equal to preorder traversing the corresponding binary tree, postorder traversing a tree is equal to inorder traversing the corresponding binary tree. In addition, only one binary tree can be determined by the preorder traversal sequence and the inorder traversal sequence. Therefore, only one tree can be determined by the preorder traversal sequence and the postorder traversal sequence. Therefore, if the preorder traversal sequences and the postorder traversal sequences of two tree are equal respectively, then they structures and the corresponding node values are equal respectively, the corresponding two mathematical formulas are matching.

Assume E_1 and E_2 are two mathematical formulas, they are described by MathML, where E_1 is the source formula, E_2 is the target formula. The match algorithm is described in detail as follow.

Step 1: If E_1 is described with content markup, then converting it into presentation markup. If E_2 is described by content markup, then converting it into presentation markup.

Step 2: Creating the tree presentation of E_1 according to its presentation markup code, normalizing the tree structure and variable names to get tree T_1. Create the tree presentation of E_2 according to its presentation markup code. Normalizing the tree structure and variable names to get tree T_2.

Step 3: Level traversing tree T_1 to get the structure code C_1. Level traversing tree T_2 to get the structure code C_2. If $C_1 \neq C_2$, then E_1 and E_2 are mismatching, goto step 6, else goto step 4.

Step 4: Preorder traversing tree T_1 to get the traversal sequence P_1. Preorder traversing tree T_2 to get the traversal sequence P_2. If $P_1 \neq P_2$, then E_1 and E_2 are mismatching, goto step 6, else goto step 5.

Step 5: Postorder traversing tree T_1 to get the traversal sequence L_1. Postorder traversing tree T_2 to get the traversal sequence L_2. If $L_1 = L_2$, then E_1 and E_2 are matching, else mismatching.

Step 6: End.

IV. THE EXPERIMENTAL RESULTS AND ANALYSIS

Researchers select 216 different mathematical formulas from 200 pressed research papers. Every mathematical formula is representative. The mathematical formulas include both basic and recombined. Such as radical expression, fraction expression, summation expression, function expression, vector expression, superscript expression and matrix etc. The mathematical formulas are source formulas. Describe them with MathML presentation markup, create the tree presentation for every mathematical formula according to its presentation markup, and normalize the structure and variable names.

To verify the performance of the proposed method, every mathematical formula is modified according to Tab .2. The modified mathematical formulas are target formulas.

Experiment is Pentium 2.0G, memory 2G and windows Xp operating system. The programming language is VC++.

In experiments, 346 times matching are done. The average accuracy is 94.28%, the average matching time is 3.24ms. The matching results of various modification way are given in Tab .3.

TABLE II. THE WAY OF MODIFYING MATHEMATICAL FORMULA

Modifying number	1	2	3	4	5
way of modifying	No modifying	Modifying variable name	Modifying structure	Modifying operand	Modifying constant

TABLE III. EXPERIMENTAL RESULTS

Modifying number	1	2	3	4	5
accuracy (%)	100	100	86.31	100	100

The experimental results show that the accuracy of the algorithm is high. For no modifying, the accuracy is 100%. The reason is that the presentation markup code of the source formula and the source formula are equal, therefore, the tree structure and the corresponding node value must be equal. For modifying variable names, the accuracy is 100%. After the variable names of the source formula and the target formula are normalized, the tree structures and the corresponding node values must be identical. For modifying operand and modifying constant, the accuracy is 100. If the operands or constants of a mathematical formula are modified, then its meaning must be modified, the original formula and the modified formula must be mismatching. For modifying structure, the accuracy is 86.31%. The key reason is that the patterns are incomplete in the rule base. The matching speed of the algorithm is fast. The reason is that the algorithm compares with the structure codes firstly; if the structure code is equal, then compares with the preorder traversal sequences; if the preorder traversal sequences are identical, then compares with the postorder traversal sequences.

V. CONCLUSION

Based on the tree representation of mathematical formula, a mathematical formula matching algorithm for MathML is proposed in this paper. The algorithm realize accurately matching of mathematical formulas, and matching speed is fast. The algorithm is not only suitable for the matching that both the structure and the meaning are identical respectively ,but alTherefore, for the matching that the structures are different but the meanings are same. Perfecting the rule base and improving the matching accuracy of the same semantic mathematical formulas would be our research work in future.

ACKNOWLEDGMENT

This study is partly supported by the National Natural Science Foundation of China (No. 61304149, No. 11171042) and 2014 education committee project of Liaoning province in China (No.L2014444).

REFERENCES

[1] T Hoad and J.Zobel, "Methods for identifying versioned and plagiarism documents," Journal of the American Society for Information Science and Technology, vol.54, no.3, pp.203-215, 2002.

[2] A. Chowdhury and O. Frieder, "Collection statistics for fast duplicate document detection," ACM Transactions on Information System, vol.20, no.2, pp.171-191, 2002.

[3] B. Jin, Y.-J. Shi and H.-F. Teng, "Document-structure-based copy detection algorithm," Journal of Dalian University of Technology, vol.47, no.1,pp.125-130, 2007.

[4] J.-J. Zhao, and X.-G. Hu, "A way to judge plagiarism in academic papers based on word-frequency statistics of paragraphs," Computer Technology and Development, vol.19, no.4, pp.231-233, 2009.

[5] X.-G. Qin, "Research on the copy detection based on the similarity of sentences," New Technology of Library and Information Service, vol.28, no.11, pp.63-66, 2007.

[6] J.B.Baker, A.P. Sexton,and V. Sorge. "Faithful mathematical formula recognition from PDF documents," Proceedings of the International Workshop on Document Analysis Systems, Boston, USA ,pp.485–492, 2010.

[7] R. Zanibbi and L.Yu. "Math spotting: Retrieving math in technical documents using handwritten query images," proceedings of the International Conference on Document Analysis and Recognition, Beijing, CN, pp.446–451, 2011.

[8] T.H Rhee and J.H. Kim. "Efficient search strategy in structural analysis for handwritten mathematical expression recognition," Pattern Recognit. Vol.42, no.12, pp.3192–3201, 2009.

[9] U.Garain and B.B. Chaudhuri. "Recognition of online handwritten mathematical expressions," IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics, vol. 34, no. 6, pp. 2366–2376, 2004.

[10] X.-Y Lin, L.-C Gao and Z. Tang. "A Text Line Detection Method for Mathematical Formula Recognition," Proceedings of International Conference on Document Analysis and Recognition, Washington, USA, pp. 339-343, 2013.

[11] R. Zanibbi, Blostein D. "Recognition and retrieval of mathematical expressions,"International Journal on Document Analysis and Recognition, vol.15, no.4, pp.331−357, 2012.

[12] K. Sain, A. Dasgupta and U.Garain. "A tree matching-based performance evaluation of mathematical expression recognition systems,"International Journal on Document Analysis and Recognition, vol.14, no.1, pp.75–85,2011.

[13] X.-Y Lin, L.-C Gao and Z. Tang. "Mathematical formula identification and performance evaluation in PDF documents," International Journal on Document Analysis and Recognition, vol.17. no.3, pp.239-255,2014.

[14] T. Zhang, L. Li and W. Su. "A Mathematical Formulae Converter Based on Mathedit," Computer Applications and Software, vol.17, no.3, pp.239-255, 2014.