Structural Analysis and Display of Handwritten Mathematical Formula Based on Ternary Tree

PENG HAN[1] LONG CAN HU[2] AI JUN FAN[1]

1Chongqing Academy of Science and Technology, Chongqing 401123, P.R. China,
han.peng@ciat.cq.cn , fan.aijun@ciat.cq.cn

2Chongqing University of Posts and Telecommunications,Chongqing 400065, P.R.China,
hu.longcan@ciat.cq.cn

Abstract

Due to arbitrariness of handwritten mathematical formula, it is difficult to do the structural analysis and displaying automatically. In this paper, an improved algorithm of generating ternary tree is proposed to solve the limitation. Firstly, a ternary tree of character sequence in handwritten mathematical formula is generated according to rule of first-in and first-out. Then the ternary tree is translated into MathML by preorder traversal algorithm, and finally handwritten mathematical formula is displayed on the web browser. Compared to the traditional binary tree, the ternary tree can show the hierarchical structure of mathematical formula better, and can also analyze structure of complicated formula well. Meanwhile, the ternary tree contains high sense of portability, which can be applied to chemical and physical formula etc.

Keywords: Handwritten mathematical formula, Structural analysis, Ternary tree, MathML.

1. Introduction

Owing to the arbitrariness of handwriting and the uncertainty of the spatial position of characters, the recognition and displaying of handwritten mathematical formula need to be solved. The recognition of handwritten mathematical formula consists of the recognition of independent character and structural analysis of formula. At present, the technology on recognition of independent character includes neural network [1], SVM (support vector machine) [2] and other methods, their recognition rates are all more than 90%.

Though many methods have been investigated to analyze and store handwritten mathematical formula [3-7], most of them are not effective or accurate. For instance, LL(1) grammar [3] and minimum spanning tree [4-6] are the common methods of structural analysis. LL(1) grammar is a top-down algorithm using stack structure (first-in and last-out), which deduces with production from start symbol until find the methods of

This work was supported by National Key Technology R&D Program of China (No. 2012BAH55F01, 2012BAH77F02, 2012BAH55F03).

inputting symbol string. It works well for the structural analysis of printed mathematical formula, but it cannot achieve high accuracy for the non-standard of handwritten mathematical formula's structure. Minimum spanning tree is processing the priority to operator to establish binary tree, whose idea is to establish binary tree after processing priority to the character set of mathematical formula. Though it saves internal storage, it demands higher priority, which affects greatly on the efficiency and accuracy of analysis. The method of normalization based on the operator priority of expression and tree structure put forward in [7]. According to the tree structure of mathematical expression with operator priority, operational data and operators in mathematical expression with communicative law are rearranged into normalized mathematical expression. The meaning of operators in expression needs to be clear and some implicit symbols need to be added artificially. For instance, in expression "a+bc", multiplication sign "*" needs to be added. A method on the base of block tree put forward by Lu and Lin [8]. The expressions are broken into several sub-modules according to their internal structural features, and the structural relation between characters in every sub-module are indicated by tree structure, and the final the expressions of tree are formed. It obtains the risk that one character might join two block trees. Methods above all need to analyze the character meaning of mathematical formula. According to the arbitrariness of handwriting and the uncertainty of the spatial position of characters, Jiang and Liu etc. [9] put forward a kind of analyzing method based on ternary tree. They graduate mathematical formula with symbol and store the expression with ternary tree. However, it can only be applied to several input symbols including 10 Arabic numbers, decimal point and four arithmetic operators. It can't recognize other characters or their superscripts and subscripts, like "a_2^3".

Here we improve the algorithm of literature [9]. The structure of handwritten mathematical formula is translated into a ternary tree. Experiments have shown that this method works well to analyze mathematical formula in terms of structure, enhances the efficiency and accuracy of the recognition of handwritten formula, and can be applied to the formula structural analysis of other subjects, such as chemistry and physics. Compared to grammar LL(1) and minimum spanning tree, it reduces the requirement of priority. Finally, MathML is generated by preorder algorithm of ternary tree, and the mathematical formula can be displayed on the web browser.

The rest of the paper is organized as follows. In Section 2, the improved algorithm of generating ternary tree is developed to analyze and store the mathematical formula. Then, the generation and display of MathML are conducted and are presented in Section 3. Finally, a conclusion is given in Section 4.

2. Improved algorithm of generating ternary tree

2.1. Concept of ternary tree

2.1.1 Ternary node

Ternary node is a data structure which contains stored information and three sub-nodes. Literatures [10-12] have studied and demonstrated the structure of ternary tree, and this structure has been applied in many fields recently [13-15].

JAVA language of ternary node is as follows:

```
public class Ternary Node {
```

```
    public String info;
        // Node's information;
    public Ternary Node llink;
        // Node's left-child;
    public Ternary Node mlink;
        // Node's mid-child;
    public Ternary Node rlink;
        // Node's right-child;
}
```

The link structure of ternary node shows as Fig. 1. "info" is the information of node. "llink", "mlink", "rlink" are three pointers, respectively point the left, middle, right sub-node.

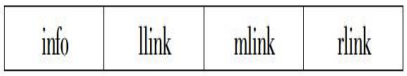

Fig. 1: Link structure of ternary node

Three-pointer method is intuitive as storage structure of ternary tree, and its time complexity of basic operation is low, but the storage efficiency is low as well. Literature [10] shows the algorithm and analysis in detail. For example, if the time complexity of the traversal time of n node tree is O(n), find the child node, the time complexity of operation inserting and deleting is only O(1).

Ternary tree has two main properties [12]:

Property 1: on i layer of ternary tree exist at most 3i-1 nodes ($i \geq 1$).

Property 2: ternary tree whose depth is k has at most (3k-1- 1) / 2 nodes ($k \geq 1$).

2.1.2. Generation of ternary tree

Ternary tree is generated by the link of one or multiple ternary nodes. The link structure can express ternary tree intuitively. Ternary tree can be generated in order (that is established by root node linking to sub-nodes in turn). Generating many ternary trees by many ternary nodes and then linking these ternary trees to form one ternary tree. Of course, this method can work as well. Specific illustration is shown in Fig.2 and Fig.3:

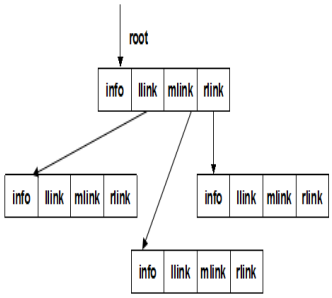

Fig. 2: Link structure of ternary tree.

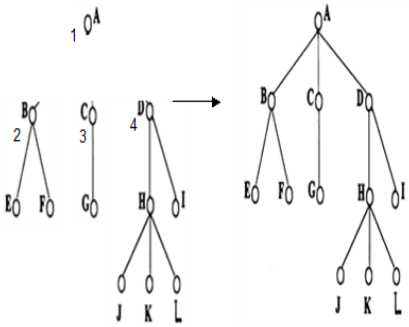

Fig. 3: Many ternary trees linking as one ternary tree.

2.1.3. Traversal of ternary tree

Similar to binary tree, ternary tree has traversal, too. Take N as the root node, L, M, R are respectively indicate the left sub-tree, middle sub-tree, right sub-tree of N. Ternary tree has four kind of main traversals, including pre-order traversal (NLMR), in-order traversal (LNMR and LMNR), post-order traversal (LMRN) and hierarchy traversal. The recursive pre-order traversal algorithm describing in java of the pre-order traversal (NLMR) in this paper is as follows:

```
Public String NLMR (TernaryNode
                        node)
{
  String Info=node.info;
  if(node.llink!=null)
   {
     Info+=NLMR(node.llink);
```

```
    }
  if(node.mlink!=null)
    {
      Info+=NLMR(node.mlink);
    }
  if(node.rlink!=null)
    {
      Info+=NLMR(node.rlink);
    }
  }
  return Info;
  }
```

2.2. Structural Analysis of the character of handwritten mathematical formula

2.2.1. Spatial position attribute of character

Spatial position attributes include the following boundary and central point of character:

Left boundary of the rectangular area containing character: $x_{left} = x_{min}$,

Right boundary of the rectangular area containing character: $x_{right} = x_{max}$,

Upper boundary of the rectangular area containing character: $y_{up} = y_{min}$,

Lower boundary of the rectangular area containing character: $y_{down} = y_{max}$,

Width of the rectangle: $W = x_{max} - x_{min}$,

Height of the rectangle: $H = y_{max} - y_{min}$,

Center of character:

$X_{center} = x_{left} + (x_{right} - x_{left})/2$,

$Y_{center} = y_{up} + (y_{down} - y_{up})/2$.

2.2.2. Relation between characters

There are two methods to judge the spatial position relation of characters: datum line and "#" font. It is easy to judge the position relation of characters in formal mathematical formula with datum line. However, it works little to recognize handwritten mathematical formula. Here we adopt "#" font method to judge the spatial position relation of characters.

"#" font can judge the spatial position relation of most characters. But the method can't be applied to some special mathematical characters, such as radical sign "$\sqrt{\ }$", bracket ("( )", "[ ]", "{ }") and so on.

To overcome the shortages, besides "#" font, the other two position relations are added. As a result, there are ten kinds of position relation, including up, down, left, right, up-left, upright, down-left, down-right, include, included. They are symmetrical structure between any two parts, which make the character spatial position more accurate. The relation of position and inclusion in "#" font is shown as Fig. 4 and Fig. 5.

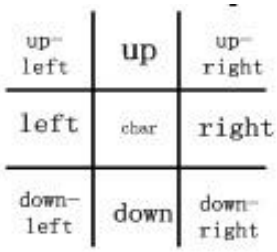

Fig. 4: Position relation of "#" font.

$$\sqrt{a+b}$$

Fig. 5: Inclusion relation.

2.2.3. Dominative structure of character

Compared to the operator priority, we weaken the character priority, and only take dominative processing on some special operators. The common structural analysis of character is shown as follows.

A. Inclusion shaped structure. For instance, the rectangular boundary of $\sqrt[5]{3}$ contains "5" and "3". So the llink of ternary node whose info is radical sign is pointed to the ternary node that stores info as "5", and the mlink is pointed to the ternary node that stores info as "3".

B. Up-down shaped structure. Such as fraction "—", summation "∑" and so on. So the llink of ternary node whose info is "—" or "∑" is pointed to the ternary node where the far up-left character places, and

the mlink is pointed to the ternary node where the far bottom-left character places.

C. Superscript and subscript structure (which only contains superscript or subscript). Take "a_2^3" as an example. So the llink of ternary node whose info is "*a*" is pointed to the ternary node where the superscript "3" places, and the mlink is pointed to the ternary node where the subscript "2" places.

D. Left-right shaped structure. Take "*ab*" as an example. So the rlink of ternary node whose info is "*a*" is pointed to the ternary node where info "*b*" places.

2.2.4. Finding root node

Finding root node plays an important role in the establishment of ternary tree. There are two main types to find root node.

(1) On the occasion that there isn't relation of up-down or inclusion in the spatial position relation of the far left character in a mathematical formula, the character is root node.

(2) On the occasion that there is relation of up-down (dominative structure B) or inclusion (dominative structure A) in the spatial position of the far left character in a mathematical formula. Firstly the node is marked as root node. Secondly, the character is checked whether is in the dominative structure (such as fraction, radical sign, summation symbol etc.) or not. If it is, the root node should be specific character instead. Otherwise, the ternary node is true.

2.3. Statement and Examples of Algorithm

Firstly N ternary nodes are generated if number of characters is N in a mathematical formula, and the information of every ternary node is initialized, in which "info" is used to store characters, and three pointers of llink, mlink, rlink all point to empty. Secondly, according to the in-order linking method of ternary tree, we start from the root node, and link sub-nodes in order of left-middle-right. Finally, a completing ternary tree is formed.

To be specific, the improved algorithm of generating ternary tree is as follows:

Step 1: Initialize an empty queue Q, and put the found root node at the end of queue L.

Step 2: Take a ternary node from queue Q and mark it as N. Judge the position relation of characters in stored information (info) of node N whether there is three dominative structures A, B and C. If so, turn to step 3. Otherwise, judge whether there is structure D. If so, turn to step 4. Otherwise, turn to step 5.

Step 3: According to the dominative structure of each kind of characters, make the left pointer (llink) and right pointer (rlink) point to its sub- ternary node, and meanwhile put its sub-nodes from the end into queue Q successively. Judge the position relation of characters in stored information (info) of node N, whether there is the dominative structures D. If so, turn to step 4. Otherwise, turn to step 5.

Step 4: According to dominative structure D of characters, make the right pointer of ternary node N point to its sub-ternary node, and meanwhile put its sub-node from the rear into queue Q. Turn to step 2.

Step 5: Judge whether queue Q is empty. If it is not empty, turn to step 2. Otherwise, destroy the queue, and algorithm is over.

The method above establishes a ternary tree in order by in and out of queue after repeated loops. For instance, Figure 5 is the handwritten mathematic formula. The setup procedure of ternary tree is shown in Figure 6, and its hierarchical structure in Figure 7.

$$\frac{\sqrt[5]{a_2^3+b}}{c}+(a+b)^2$$

Fig. 6: Handwritten formula.

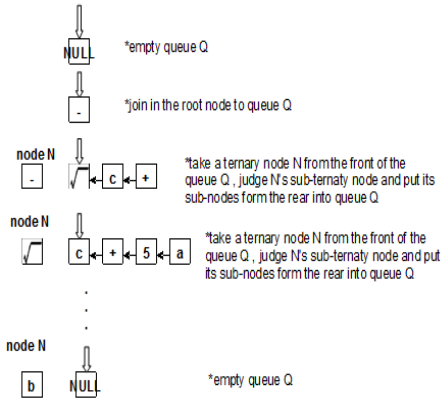

Fig. 7: Establishment order of ternary tree.

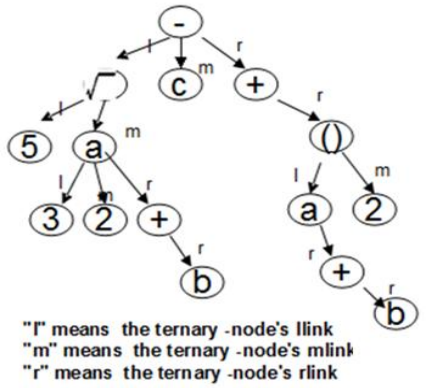

Fig. 8: Structure of ternary tree.

3. Generation and result display of Math ML

W3C Mathematics Workgroup first put forward MathML and released its final edition in October, 2003. MathML describes the structure and content of mathematical expression with XML in order to make Web to serve, receive and process mathematical formula. Literatures [16-18] showed the application of MathML in mathematical formula.

Storing mathematical expression in ternary tree has an obvious advantage that it helps to generate MathML. The adoption of preorder of ternary tree can generate MathML, that is if some node contains left child (llink is not empty) or middle child (mlink is not empty), first check the type of the character stored by the node (such as fraction, radical sign, summation symbol etc.), generate different MathML markup languages in line with different types (MathML of fraction is <mfrac></mfrac>, MathML of radical sign is marked as <mroot></mroot>), put the left child and middle child into the markup language, the right child out. As for the ternary tree shown in Fig 7, MathML in preorder is shown as follows:

```
<math
xmlns="http://www.w3.org/1998/Math/MathML">
  <mstyle displaystyle="true">
   <mfrac>
    <mrow>
     <mroot>
       <mrow>
            <msubsup>
             <mrow>
              <mi> a </mi>
             </mrow>
             <mrow>
              <mn> 2 </mn>
</mrow>
             <mrow>
              <mn> 3 </mn>
             </mrow>
            </msubsup>
            <mo> + </mo>
            <mi> b </mi>
       </mrow>
       <mrow>
            <mn> 5 </mn>
       </mrow>
     </mroot>
    </mrow>
    <mrow>
     <mi> c </mi>
    </mrow>
   </mfrac>
   <mo> + </mo>
   <msup>
    <mrow>
```

```
        <mfenced>
          <mrow>
            <mi> a </mi>
            <mo> + </mo>
            <mi> b </mi>
          </mrow>
        </mfenced>
      </mrow>
      <mrow>
        <mn> 2 </mn>
      </mrow>
    </msup>
  </mstyle>
</math>
```

The view on the web browser is shown as Fig 9.

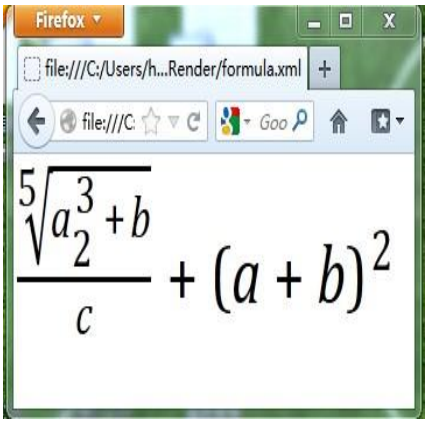

Fig .9: View on the Web Browser.

4. Conclusion

In this paper, we propose a method that adopts ternary tree to store mathematical formula and translates mathematical formula into MathML in preorder of ternary tree for network display. Experiments have shown that it can enhance the efficiency and accuracy of the recognition of handwritten mathematical formula effectively. This method can contribute to the cloud service construction of education. It can rapidly join applications such as teaching, learning, management etc. with different terminal carriers (smart phone, computer, laptop, compound learning terminal etc.), and help to build a lifelong education system, form a learning society and boost the continuous study of members of society all their life.

Merely the handwriting recognition and display of mathematical formula are analyzed in the paper. Due to the multidisciplinary comprehensiveness of educational cloud service, for example, the handwriting recognition and display of inorganic chemical formula have not been analyzed. However, the application of its improved algorithm generating ternary tree will be explored further in the future work.

5. References

[1] A. Choudhary and R. Rishi, “Improving the character recognition recognition efficiency of feed forward BP neural network,” International Journal of Computer Science & Information Technology, Vol. 3, No. 1, pp. 85-96, 2011.

[2] T. M. Tri Do and T. Artières, “Learning mixture models with support vector machines for sequence classification and segmentation,” Pattern Recognition, Vol. 42, No. 12, pp. 3224-3230, 2009.

[3] W. WU and L.C. HOU, “Structural analysis of printed mathematical expressions using LL(1) grammar,” Journal of Dalian University of Technology, Vol. 46, No. 3, pp. 454-459, 2006.

[4] K. F. Chan and D. Y. Yeung,“An efficient syntactic approach to structural analysis of on-line handwritten mathematical expressions,” Pattern Recognition, Vol. 33, No. 3, pp. 375-384, 2000.

[5] R. Zanibbi and D. Blostein,“Recognizing mathematical expressions using tree transformation,” IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 24, No. 11, pp. 1455-1467, 2002.

[6] B. Q. Vuong, S. C. Hui, and Y. L. He, "Progressive structural analysis for dynamic recognition of on-line handwritten mathematical expressions," Pattern Recognition Letters, Vol. 29, No. 5, pp. 647-655, 2008.
[7] L. J. Chen and C. Jing, "Research on normalization method of mathematical expression," Journal of Zhejiang university of Technology, Vol. 40, No. 2, pp. 229-232, 2012.
[8] X. W. Lu and J. Y. Lin, "A structural analysis approach to online handwritten mathematical expression," Computer Engineering and Science, Vol. 32, No. 10, pp. 69-72, 2010.
[9] H. Jiang, C.S. Liu, S.Y. Li, and Y. M. Zou, "An on-line handwritten arithmetic expression recognition system," Journal of Electronics & Information Technology, Vol. 32, No. 5, pp. 1126-1130, 2010.
[10] N. X. Zhang, "Trinary tree structure and its implementation," Journal of Computer Research and Development, Vol. 1, pp. 50-54, 1993.
[11] G. J. Mao, and D. F. Yang, "A Storage Structure for Trinary Tree and the Implementation of Its Basic Operations," Computer Research and Development, Vol. 31, No.5, pp.62-65, 1994.
[12] Y.Q. Yin, "The Analysis and Comparison of the Storage Structures for Trinary Tree," JOURNAL OF GUANG XI UNIVERSITY FOR NATIONALITIES, Vol. 9, No. 2, pp. 50-53,2003.
[13] J.F. Tian, J.C. Huang, R.Z. Du, and J.Q. Zhai, "Research of a high performance pattern matching algorithm ,"JOURNAL OF CHINA INSTITUTE OF COMMUNICATIONS, Vol. 25, No.1, pp. 61-69,2004.
[14] X.P Liang, and Y Zhu, "Research of Merging Algorithm of Implement based on Balanced Trifurcate Tree," Microcomputer Information, Vol.23, No.3, pp. 235-237, 2007.
[15] Y.Y. He, "A trinomial tree methods for pricing American options," JOURNAL OF NATURAL SCIENCE OF HEI LONG JIANG UNIVERSITY, Vol.25, No.1, pp. 81-84, 2008.
[16] J. Francis, and Wright, "Interactive Mathematics via the Web using MathML," SIGSAM Bulletin, Vol. 34, No. 2, 2000.
[17] R. Ross, "Education forum: trying again with MathML," SIGACT News, Vol.36, No.3, pp.82-84, 2005.
[18] T. W. Cole, "MathML in practice: issues and promise," Data Science Journal, Vol. 5, pp. 209-218, 2006.