Top-Down Expression Structure of Mathematical Document for Nonvisual Communication

Ryoji Fukuda
rfukuda@oita-u.ac.jp
Faculty of Science and Technology,
Oita University
Japan

Naoki Tamura
v16e6010@oita-u.ac.jp
Graduate School of Engineering,
Oita University
Japan

Yuki Toyosaka
toyosaka@ip.kyusan-u.ac.jp
Faculty of Infomation Science,
Kyushu Sangyo University
Japan

Abstract:
Mathematical expressions are originally graphical contents, and there are several methods to express them through text contents or voice outputs. However, these are sometimes difficult to understand because of the information structure. In this study, we considered a top-down expression structure for mathematical documents. We prepared four expression patterns, by using which the whole document can be expressed as a nested structure. We also defined an expression method similar to natural language, and its translation from manual explanation is not difficult. This expression can be automatically translated to our top-down expression. By using these expression methods, we could realize easier scientific communication without visual information.

1. Introduction

Nonvisual communication is not only for the visually handicapped. In oral communication using telephone or secondary audio channel of TV programs, we usually do not use visual contents. In these cases, we can use notepads or small information boxes to memorize the contents; they are difficult to understand by memorizing them only in brain. However, these contents must depend on the structure of information elements. Thus, we considered some adequate structures of information for easy understanding of the documents.

Visual contents are divided into several types including text content, math expressions, figures and photos. Some content are easy to convert to nonvisual content, while others are very difficult. Many studies have concentrated on such conversions. Optical character recognition (OCR) and braille translation technologies enable us to automatically translate from a printed text document to a braille code document[1]. Mathematical expression can also be translated to text-based contents (e.g., Tex sources and Math-ML documents.) or braille documents[1]. Several studies have been conducted for understanding figures or pictures[3][4].

However, these studies do not provide a perfect solution to improve the situation. Consider the case in which a printed text document is translated to a braille document or voice outputs. Considerable differences can be observed between the original and final documents. In visual communication, we often use a printed text document as a reminder for when we cannot remember all information contents. In such situations, we need to remember or regain some of the information content according to our needs. Thus, we use some figures in a document for easy understanding of the contents. Sometimes, these figures express mutual relations and sometimes they summarize the contents. The purpose of the current study is to create an expression method for a mathematical document so that by using the expression, individuals can grasp their content without visual information.

In a text book or an article, necessary information tips are constructed using the bottom-up approach. As such, we cannot grasp the article's rough story before we read of the information tips including all details. It becomes very complicated if all the detailed explanations are provided; in contrast, it may be very difficult if many details are skipped. In this study, we attempted to translate these explanations to top-down explanations. The content structures in a mathematical document can be divided into several groups "proposition", "operation", "definition", and "array". In addition, an entire document can be expressed through the recursive expressions of these four groups. An XML document is suitable for expressing a top-down structure. We created an expression rule for mathematical documents and all the documents fall into one of the four types. A document of each type is divided into several elements and every sub-element is categorized into one of the four types. Next, a terminal node was considered a simple sentence. Thus, we were able to express a mathematical document.

Translation to an XML document is not easy. In general, the understanding of natural languages includes many problems because some ambiguities or errors may exist. Thus, we defined some rules to express documents. Some conjunctions express the types of XML elements. We defined some expression tags based on these conjunctions and named the expression a "tagged document." For a tagged document, an XML structure is uniquely determined. In this way, we can create the translation tool to convert a tagged document to an XML document.

Translation from a row document to a tagged document is not difficult compared with its direct translation to an XML document. The present stage includes some human judgments; however, we can implement a half-automatic translation rule for such problems. It may be extremely difficult to create a perfect translation tool but we could use some suitable interactive system as an easy translation system. For translating an XML document, we do not have any convenient output methods. Thus, in the future, we will modify our translation system to include certain output methods.

2. XML Document

We used the XML format to express our top-down structure for mathematical documents and defined four element types, including a whole document. The terminal type is a simple sentence, which is short and can be grasped at once. According to the structure type, every element is divided into several sub-elements if the element is not a simple sentence.

2.1 Four Element Types

We considered four types of elements for a mathematical document: "proposition," "operation," "definition," and "array." Each type has a unique structure and the role of each sub-element.

A proposition element consists of at most three sub-elements: "hypothesis," "conclusion," and "proof." For each sub-element type, there is at most one element. There are many situations, some is very simple "proof" and others with very complicated "proof." In case an element represents a simple property, it may contain no proof but both hypothesis and conclusion would be simple sentences. Such factors may be a part of conclusion in a complex proposition element.

A definition element consists of two sub-elements "def body" and "def sub." Def body is a concept or defines some situations, while def sub expresses its condition. There is only one sub-element for one sub-element type. Therefore, the structure is similar to the proposition element without the proof element. However, their meanings or purposes may differ. Thus, the element type is an important information tip for understanding the document.

An operation element also consists of three sub-elements: "from," "to," and "operator." This expresses a one-to-one correspondence between two elements, that is, the results of functions, mappings, and some simple correspondences can be expressed. The sub-elements "from" and "to" can be very complex based on the situation; however, the sub-element "operator" is supposed to be a simple name, that is, it is a simple sentence.

A simple group of elements is expressed using an array type element. This element has several sub-elements ("arraySub"), which are expressions often found in "hypothesis" or "defSub". In such cases, elements have a logical meaning and two mutual relations "and" and "or" which are expressed using attribution of an element.

2.2 Expression Rules for an XML Document.

Tags of our XML document include "MathDocGraph" and "MdgElement," and all element types are expressed using the attribution "mdgType." Our XML document consists of one "MathDocGraph" element comprising one sub-element ("MdgElement"). which in turn has several sub-elements according to its type. The values of the attribution "mdgType" in each sub element involves a "proposition," "operation," "definition," "array," and "terminal."

Attributions of an "MdgElement" include "mdgType," "mess," and "linkType." Except for mdgType, all elements are sub-element of MdgElement. The attribution "linkType" expresses the element role. The values of this attribution are listed in Table 2.1. Values of the attribution "mess" are string data. If "terminal" is the value of "mdgType," it expresses the document content, and in other cases, these are set to provide a short explanation of the sub-element. In general, the automatic creation of the values of the attribution "mess" is difficult. This may be the title of the theorem or defined concept. For a top-down document, a brief explanation of the root element of a subtree structure is very important to understand the document. In our document, each element is a root element of some subtree structure. Therefore, the attribution "mess" plays a key role in our document. Tables 2.1 and 2.2 list XML tags and element and link types, respectively.

Table 2.1 Tag names and Attributions

XML Tags	Attributions
MathDocGraph	name, id, date
MdgElement	mdgType(element type), linkType(relation type)

Table 2.2 Element and Link Types

Values of MdgType	Values of linkType	purposes
proposition	hypothesis, conclusion, proof	Properties, Theorem, thesis
definition	defMain, defSub	definition,
operation	from, to	function, mapping, correspondence
array	arraySub	group, list
terminal		short sentence

2.3 Sample Document

We consider the following simple explanation of the midpoint consolidated theorem, which is a well known geometric theorem.

Midpoint Theorem
Let ABC be a triangle. Let M and N be the midpoints of AB and AC respectively. Then AB is parallel to MN, and AB=2 MN.
Proof.
Triangle ABC and AMN are similar, because AM/AB =AN/AC=1:2, $\angle$BAC= $\angle$MAN .
Then, $\angle$ABC= $\angle$AMN. This implies that AC is parallel to MN, and AC is parallel to MN because alternate angles are the same. Thus we have AB=2 MN using homothetic ratio.

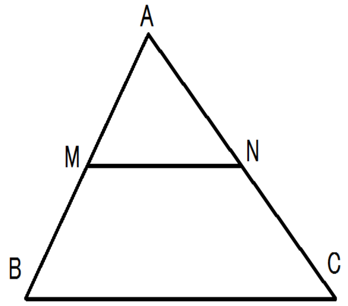

Figure 2.1 Explanation of the midpoint consolidated theorem

According to our proposed rules, this explanation is translated to an XML document (Figure 2.2).

```
<MathDocGraph title="Midpoint theorem">
  <MdgElement mdgType="proposition" title="Midpoint consolidated theorem">
    <MdgElement mdgType="array" linkType="hypothesis">
      <MdgElement mdgType="terminal" linkType="arraySub" mess="Let ABC be a triangle."/>
      <MdgElement mdgType="terminal" linkType="arraySub"
                              mess="Let M and N be mid points of AB and AC."/>
    </MdgElement>
    <MdgElement mdgType="terminal" linkType="conclusion"
                              mess="AB is parallel to MN, and AB=2 MN."/>
    <MdgElement mdgType="array" linkType="proof" mess="proof">
      <MdgElement mdgType="operator" linkType="arraySub">
        <MdgElement mdgType="proposition" linkType="from" name="similar">
          <MdgElement mdgType="terminal" mdgLink="conclusion"
                                    mess="Triangle ABC and AMN are similar"/>
          <MdgElement mdgType="array" linkType="proof">
            <MdgElement mdgType="terminal" linkType="arraySub"
                                      mess="AM/AB=AN/AC=1/2"/>
            <MdgElement mdgType="terminal" linkType="arraySub" mess="∠BAC=∠MAN"/>
          </MdgElement>
        </MdgElement>
       <MdgElement mdgType="terminal" linkType="to" mess="AC parallel to MN "/>
        </MdgElement>
        <MdgElement mdgType="operator" linkType="arraySub"/>
          <MdgElement mdgType="array" linkType="from"/>
            <MdgElement refName="similar" linkType="arraySub"/>
            <MdgElement mdgType="terminal" linkType="arraySub"
                                        mess="the homothetic ratio"/>
          </MdgElement>
          <MdgElement mdgType="terminal" linkType="to" mess="BC=2 MN"/>
        </MdgElement>
      </MdgElement>
    </MdgElement>
</MdgDocument>
```

Figure 2.2 Translation to an XML document

3. Tagged Document

Translation to an XML document is very bothersome, and manual translation of a long mathematical explanation is not reasonable. Furthermore, automatic translation is difficult because some human errors are inevitable. However, this task comprises many automatic mechanisms, which are the main reason for its complexity. Therefore, we can reduce the complexity if we can separate these aspects. We consider a new expression format with the following concept.

1. Near to standard sentences. (Everyone can understand and read it.)
2. Rule is simple. (Easy to remember the rules.)
3. The XML structure is determined uniquely.

3.1 Tags of Documents

Here, we discuss about the role of conjunctions. In may situations we have typical conjunctions to connect with their components. For example, the conjunction "because" is often used to connect a proof and a conclusion of a proposition. In a natural language sentence, the same words can be used for different purposes, therefore, we cannot determine the situation by using a conjunction. Moreover, some tags are required for grouping and referencing to create nested structures. Thus, we define some tag words for determining a structure in a document (Table 3.1).

Table 3.1 Tags

Tag (common)	Role	Tag (each)	Role	XML Element Type
@simplex	Short sentence	@and (@,)	Set "and" connection	array
@begin ({)	Start point of a group	@or	Set "or" connection	array
@end (})	End point of a group	@therefore	"proof" ⇒ "conclusion"	proposition
@ref	reference	@implies	"from" ⇒ "to"	operation
@label	set a label	@because	"conclusion"=>"proof"	proposition
@TITLE	title for the whole	@then	"proof"⇒"conclusion"	proposition
@title	title for an element	@to	"from" ⇒ "to"	operation
@mess	short comment	@givenby	"to"⇒"from"	operation
@not	deny	@with	"defMain"⇒"defSub"	definition
		@defby	"defMain"⇒"defSub"	definition

3.2 Grouping and Reference

We set a title or message using the tags "@TITLE," "@title," and "@mess." "@TITLE" is for setting the title for the whole document, "@title" is for setting the title for a group or an element, and "@mess" is for setting a comment. These tags need string parameters, for example, "@mess" is given by "@mess[this is a sample message]." Although these are not necessary to construct the XML document, they are important when considering some output method.

For the expression of nested structures, we must group some packet of explanations. Thus, we created a group by using the tags "@begin" and "@end ("{" and "}"), and set a label by using "@label." This needs a string parameter, and the expression method is the same as that used with "@mess." The group can be reffered using the tag "@ref."

3.3 Conjunction Tags for Array

The sub-elements of the "array" elements usually are connected through "@and" or "@,." That is, the elements before and after the tag "@and" or "@," are "arraySub" sub-elements of "array." Our expressions of logical explanations are a combination of "@and," "@or," and "@not." Thus, we can construct a nested expression of "array" elements as these tags connect previous and next elements or groups.

3.4 Conjunction Tags for Property

The tag "@therefore" connects the "proof" and "conclusion" sub-elements of some proposition sub element. The previous element of "@therefore" is a "proof" element and the next element is a "conclusion" element. That is, both are sub elements of a "proposition" element. The tag "@then" connect "hypothesis" and "conclusion" sub-elements. Thus, we can construct the structure of the "proposition" element. If one of these tags does not exist a sub element "hypothesis" or "proof" does not exist in general.

3.5 Conjunction Tags for Operator

The structures of these elements are very similar, that is, two sub-elements are connected with a tag and each sub-element has a different role. The tag "@to" connects "from" and "to" sub-elements; the previous element is "from" and the next element is "to". In contrast, for "@givenby," the previous element is "to" and the next element is "from." Thus, these two tags have different functions.

3.6 Conjunction Tags for Definition

The "definition" element has two tags "@with" and "@defBy" that connect "defMain" and "defSub" sub-elements. For both the tags, the previous element is "defMain" and the next element is "defSub". These have the same structures but different meanings.

3.7 General Rules For Tagged Documents.

The smallest unit of the document is a simplex (a small sentence). When we describe it explicitly, we may use the tag "@simplex" (sample expression: "@simplex[Triangle ABC]" or simply "Triangle ABC"). A group is a set of units enclosed in "{ }" or "@begin" - "@end." We call a set of units connected by conjunction tags an element. The unit can be a simplex, a group, or an element. We use a group expression if the boundary of a group is ambiguous.

The tag "@title" is placed in front of a group or a simplex, and a label is at the end. The unit with label can be referred using "@ref." The following expression is an example.

"@title[Assumption]@simplex[AB parallel to CD]@label[A1]........ @ref[A1]

4. Translations

We defined two expression rules for an XML document and a tagged document. We assumed that the tagged element is created manually, and developed a system to translate the tagged document to an XML document.

4.1 Translation to a tagged document

We considered several human judgments for creating a tagged document from a standard explanation. We want to create automatic translation software or auxiliary software in the future.

The following list is a half-automatic procedure to create a tagged document.

1. Find a word corresponding to the tags in Table 3.1.
2. If there is no such a word, create a simplex unit.
3. If a word is chosen, select the most suitable tag.
4. Group some units before and after the word if necessary.
5. Set a title or a comment if necessary.
6. If some part of the document is referenced set an adequate group and its label.
7. If there are some errors, adjust the structures or grouping.

The adjusting tasks in seventh line may be most difficult for automatization as there are various unexpected expressions in natural language. Figure 4.1 illustrates a tagged document created using the above rules.

4.2 Additional Rules for XML Documents

Some additional rules are required to express a tagged document in the XML format. Table 4.1 lists some additional attributions.

Table 4.1 Additional Attributions

XML TAG	Attributions
MathDocGraph	title
MdgElement	title, name, refName, refId,mess

An XML element with "refId" or "refName" attributions, which respectively denote the "id" and "name" of the corresponding element, does not have any content. The attributions "refId" and "refName" denote the attribution "id" and "name" of corresponding element.

4.3 Translation to an XML document

We created rules for the tagged document to determine an XML document uniquely. In our software, an XML document is created according to the following steps.

1. Create an XML element with tag "MathDocGraph"; there is one such an element in a document.
2. When "@TITLE" is defined, the attribution "title" is set for the "MathDocGraph" element.
3. Create an XML element with tag "MdgElement".
4. When a group start is determined, push the present state and extract all the contents of the group. Then, the task repeates from the step 3.
5. When a group end is determined, pop the stored state and restart the task
6. When a conjunction tag is found, the system sets an element type, creates two sub-elements, and sets the sub-element types for these elements if the element type is not defined and there are no sub-elements.
7. If element type is already defined, the system creates one element and sets a sub-element type. In this step the conjunction tag can be incorrect, because of which the system returns error message.
8. There are no remaining documents; thus the task is completed.

```
@TITLE[Midpoint theorem]
@title[Midpoint consolidated theorem]
@begin
        @label[assumption]
        {
                Let ABC be a trianble.
                Let M and N be mid points of AB and AC.
        }
        @then
        AB is parallel to MN, and AB=2 MN.
        @title[proof]
        @because
        {
                Triangle ABC and AMN are similar
                @because
                {
                        AM/AB=AN/AC=1/2 @,
                        ∠BAC=∠MAN
                }
                @implies
                ∠ABC=∠AMN
                @implies
                {
                        AC parallel to MN
                        @because
                        alternate angles are equal to each other
                }
                @,
                {
                        @ref[similar] @,
                        the homothetic ratio
                        @implies
                        AC=2 MN
                }
        }
@end
```

Figure 4.1 An example of tagged document created using the above rules

5. Conclusions

We created a top-down expression method using the XML format. In a communication without visual information, we are unable to grasp many things at a time. In such case, the top-down structure is suitable to grasp the outline of the document and obtain details according to the demands. Documents expressed through natural language often have bottom-up structure, and their translation to our XML document are very difficult. Thus, we defined an expression rule for a tagged documents, and created a system to translate a tagged document to an XML document. We also defined a semi-automatic procedure to translate the standard expression through natural language to our tagged document. In the near future, we plan to a fully or semi-automatic translation system to convert documents from natural language to XML.

References

[1] Ng, C.; Ng, V.; Lau, Y., Regular feature extraction for the recognition of braille,
Third International Conference on Computational Intelligence and Multimedia Applications,p302–306, 1999
[2] M.Suzuki, F. Tamari, R. Fukuda, S. Uchida, T. Kanahori, INFTY: an integrated OCR system for mathematical documents, ACM Symposium on Document Engineering, , p95-104, 2003
[3] K. Komiya, J. Onishi and T. Ono, A Fundamental Study of Pattern Recognition Learning Tool by Auditory labels based on the Vector component for People with Visual Impairment,IEICE technical report Welfare Information technology, p13-18, 2011
[4]Naoki Tamura, Yuki Toyosaka and Ryoji Fukuda, Intention analysis in explainer indications for elementary geometry documents using laser pointer images, Asian Technology Conference in Mathematics, 2016, Poster session