

MathMLにおける一意な表現形式記述への自動変換

理学専攻・情報科学コース 渡辺 裕美（指導教員：浅本紀子）

1 はじめに

近年、インターネット利用の多様化や教材のデジタル化に伴い、Web ページにおける数式の表現は重要性を増すとともにその表現方法も変化してきている。以前は数式が記述されている画像を Web ページ貼り込む方法が多く利用されていたが、現在では数式を Web ページ上に直接記述する方法が利用されている。例えば、MathJax を使用して $\text{T}_{\text{E}}\text{X}$ で記述する方法、MathML[1] で記述する方法、Microsoft Office や Mathematica などの専用ソフトウェアを利用して記述する方法などがある。本研究では、コンピュータによる数式の意味認識において有利であり、また柔軟に数式を記述することのできる MathML に注目した。

MathML とは、W3C が勧告している XML アプリケーションの 1 つで、数式を記述するためのマークアップ言語である。HTML の文書内に埋め込み HTML 文書として記述する。MathML には、表現に特化した表現形式記述 (W3C: MathML では Presentation Markup)、意味に特化した意味形式記述 (W3C: MathML では Content Markup) の 2 種類の記述の仕方がある。本研究では、Web 上で表示することのできる表現形式記述に着目し、表現形式記述の問題点を改善するシステムの提案・開発した。

2 表現形式記述

2.1 表現形式記述とは

表現形式記述とは、数式を表現することに特化した記述方式である。数式を表現する際に文字の大きさ・位置関係を指定してレイアウト構造を定めるなどして記述される。概念は $\text{T}_{\text{E}}\text{X}$ と同様だが、数式に意味情報を付与する見えない演算子があるなど、 $\text{T}_{\text{E}}\text{X}$ より数式の意味情報に関して詳細に記述できる。

数式: $ax + b$
<code><mi>a</mi></code>
<code><mo>&InvisibleTimes;</mo></code>
<code><mi>x</mi></code>
<code><mo>+</mo></code>
<code><mi>b</mi></code>

図 1: 表現形式記述の記述例

2.2 表現形式記述の問題点

表現形式記述は、W3C の規格に基づいたある程度正規的な書き方が存在する。しかし、表現形式記述は同一の数式を記述する際に複数の記述の仕方が存在する数式がある。そして、複数の記述の仕方が存在する場合、数式の見た目に現れない意味表現に関する箇所は省略した形で記述されることがある。その省略される箇所が、見えない演算子 (図 2) という数式の見た目に記号として現れないが数式に意味情報を持たせる演算子を記述する箇所と、複数の変数がまとめて記述される箇所である。例を図 3 に示す。図は同じ数式 $ax + 1$ を記述している。この例における ax は $a \times x$ である。

図の左が正規的な書き方である。 a と x の間に乗算を意味する見えない演算子 `⁢` が入ることで、 $a \times x$ であるということを正確に表している。図 3 の中央が `⁢` を省略した形である。図 3 の右が変数がまとめて記述された形である。本来分割して記述される a と x が、変数を記述する際に使用されるタグである `<mi>` にまとめて記述されている。

Unicode 名	コード番号	文字実体参照	同(短縮形)	使用例
FUNCTION APPLICATION	U+2061	<code>&ApplyFuntion;</code>	<code>&af;</code>	$f(x)$
INVISIBLE TIMES	U+2062	<code>&InvisibleTimes;</code>	<code>&it;</code>	ax
INVISIBLE SEPARATOR	U+2063	<code>&InvisibleComma;</code>	<code>&ic;</code>	a_{ij}
INVISIBLE PLUS	U+2064			$3\frac{2}{5}$

図 2: 見えない演算子一覧

数式: $ax + 1$		
<code><mi>a</mi></code>	<code><mi>a</mi></code>	<code><mi>ax</mi></code>
<code><mo>&InvisibleTimes;</mo></code>	<code><mi>x</mi></code>	<code><mo>+</mo></code>
<code><mi>x</mi></code>	<code><mo>+</mo></code>	<code><mi>b</mi></code>
<code><mo>+</mo></code>	<code><mi>b</mi></code>	
<code><mi>b</mi></code>		

図 3: 表現形式記述の問題点

3 変換対象

変換の対象とする数式の範囲は、高等学校までの算数及び数学の学習指導要領 [2] に記載されている範囲に含まれる数式とし、指導要領は平成 20 年及び平成 21 年に改訂されたものを参照する。また、見えない演算子については、多く用いられる FUNCTION APPLICATION と INVISIBLE TIMES を扱う。

4 概要

4.1 システムの概要

本研究は、表現形式記述において、見えない演算子が省略されて書かれた部分に適切な見えない演算子を加え、適当に書かれた変数に対して分割するなどの適切な処理を行い、正規的な書き方に変換するシステムである。

システムの一連の流れは以下の通りである。

1. 変換したい入力ファイルを読み込んで MathML が記述された箇所を MathML ファイルとして抽出する。
2. MathML ファイルから表現形式記述で用いられるタグを利用しリストを作成する。
3. 作成したリストから正規的な表現形式記述に変換し変換後の MathML ファイルとして出力する。
4. 変換後の MathML ファイルを入力ファイルの MathML 箇所に置き換え、入力ファイルと同様の形式のファイルとして出力する。

4.2 リストの作成

リストは図 6 のように作成する。リストの starttag,endtag,content から変換をどう行うかの処理を施す。

starttag : <math>	starttag : <mi>	starttag : <mo>	starttag : <mi>	
content : NULL	content : f	content : (content : x	
endtag : NULL	endtag : </mi>	endtag : </mo>	endtag : </mi>	
comment : NULL	comment : NULL	comment : NULL	comment : NULL	...
num : 1	num : 2	num : 2	num : 2	
prev : NULL	prev : NULL	prev : NULL	prev : NULL	
next	next	next	next	
リスト1	リスト2	リスト3	リスト4	

図 4: リストの作成

4.3 見えない演算子の処理

見えない演算子を使用した記述の例を図 4 に示す。 $f(x)$ や $x(x+1)$ のような変数の後に括弧が来る場合、関数・乗算・座標の場合と複数の意味で取ることができるので、ユーザが判断して見えない演算子の処理を行う。

FUNCTION APPLICATION	INVISIBLE TIMES
$\langle \text{mi} \rangle \text{sin} \langle \text{mi} \rangle$ $\langle \text{mo} \rangle \&\text{af}; \langle \text{mo} \rangle$ $\langle \text{mi} \rangle \theta \langle \text{mi} \rangle$	$\langle \text{mn} \rangle 2 \langle \text{mn} \rangle$ $\langle \text{mo} \rangle \&\text{it}; \langle \text{mo} \rangle$ $\langle \text{mi} \rangle a \langle \text{mi} \rangle$
$\langle \text{mi} \rangle f \langle \text{mi} \rangle$ $\langle \text{mo} \rangle \&\text{af}; \langle \text{mo} \rangle$ $\langle \text{mo} \rangle (\langle \text{mo} \rangle$ $\langle \text{mi} \rangle x \langle \text{mi} \rangle$ $\langle \text{mo} \rangle) \langle \text{mo} \rangle$	$\langle \text{mi} \rangle x \langle \text{mi} \rangle$ $\langle \text{mo} \rangle \&\text{it}; \langle \text{mo} \rangle$ $\langle \text{mo} \rangle (\langle \text{mo} \rangle$ $\langle \text{mi} \rangle x \langle \text{mi} \rangle$ $\langle \text{mo} \rangle + \langle \text{mo} \rangle$ $\langle \text{mn} \rangle 1 \langle \text{mn} \rangle$ $\langle \text{mo} \rangle) \langle \text{mo} \rangle$

図 5: 見えない演算子を使用した例

4.4 変数の処理

本来であれば別々に記述される変数が、1つのタグでまとめて記述された場合に分割処理を施す。図 5 に例を示す。複数の変数がまとめられて記述された場合は、別々の変数の場合と1つの変数の場合があるため、こちらもユーザが判断して分割処理を行う。

まとめて記述された例	
$\langle \text{mi} \rangle xyz \langle \text{mi} \rangle$	$\langle \text{mi} \rangle 2x \langle \text{mi} \rangle$
$\langle \text{mi} \rangle \text{sin} x \langle \text{mi} \rangle$	$\langle \text{mi} \rangle \text{log} x \langle \text{mi} \rangle$
$\langle \text{mi} \rangle \text{exp} x \langle \text{mi} \rangle$	

図 6: 複数の変数がまとめて記述された例

4.5 実装

数式 $f(x) = 2(x+1)(x-1)x^2 + ab$ を記述した example.html というファイル为例として実行する。本システムは CUI プログラムとして動作するもので、「./henkan」というコマンドにより処理が開始され引数に変換したいファイル名を入力する。実際の実行画面は図 7 のようになっている。example.html においてユーザの判断が必要な変換箇所は図 8 の箇所であり、今回は $f(x)$ の箇所は関数、 ab の箇所は ab という1つの変数と判断し変換処理を行った。変換前と変換後で変換した箇所の一部を図 9 に、変換前と変換後のファイルを Firefox でそれぞれ表示したものを図 10 に示す。

```
watanabehirominoMacBook-puro:henkan watanabehiromino$ ./henkan example.html
1つ目のブロック内のε(ε)がかけ算だったら1、関数だったら2、座標やこれら以外だったら3を入力してください
1つ目のブロック内にあるab内の変数:abが、別々の変数の掛け算であれば1、abという名前の変数であれば2を入力してください
```

図 7: 実行画面

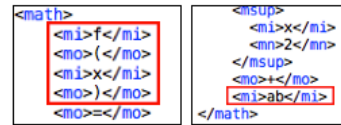


図 8: ユーザの判断が必要な変換箇所

5 考察

本システムで表現形式記述で記述されたファイルの変換を行ったところ、変換が必要だと認識した箇所においては適切な処理を行い、変換を行う必要ない箇所については変更を加えずに出力するという結果を得られた。Web 上でも元の数式と同じ数式を表示することができた。また、変換を行ったことで Web 上での見た目もより綺麗に表示されている。

6 まとめと今後の課題

本研究では、高校数学までの範囲で、表現形式記述で記述された MathML について定期的に一意な表現形式記述に変換するシステムを開発した。

本研究により、数式の意味情報における曖昧さを解消できるので、表現形式記述から意味形式記述への変換を補助することができる。また、本研究で変換することでより綺麗に数式を表現することができる。

しかし、本研究では変換する数式の範囲を絞って変換を行ったため、範囲外の数式において正しい記述に変換を行えない可能性がある。よって、表現形式記述のあらゆる記述の仕方やすべての数式で変換を行えるようにする必要がある。また、ユーザインタフェースが見づらいため使いやすい GUI を開発する必要がある。

参考文献

- [1] W3C 「MathML」 <<https://www.w3.org/TR/MathML3/>> (2020/1/8/閲覧)
- [2] 文部科学省 (2012) 「学習指導要領」 <https://www.mext.go.jp/a_menu/shotou/new-cs/youryou/1356249.htm> (2020/1/8/閲覧)



図 9: 変換前と変換後で変換した箇所の比較

MathMLによる数式の記述	MathMLによる数式の記述
MathMLで数式を記述します	MathMLで数式を記述します
1つ目	1つ目
$f(x) = 2(x+1)(x-1)x^2 + ab$	$f(x) = 2(x+1)(x-1)x^2 + ab$
変換前	変換後

図 10: Web 上での比較