

MathMLにおける一意な表現形式記述への変換ツールの開発

理学専攻 情報科学コース 2040658 八巻 澄奈 (指導教員: 浅本 紀子)

1 はじめに

Web ページ上に数式を記述する方法として、これまででは、数式の画像を貼り付けるという方法がよく利用されていた。しかし、現在では、Web ページ上に直接数式を記述するという方法が多く利用されるようになった。これにより、数式処理ソフトに入力して再利用するなどのことが可能になり、利便性が向上した。この方法のうちの一つが数式を記述するためのマークアップ言語、MathML である。本研究では MathML の問題点を解決する Web アプリケーションを提案し、開発を行なった。

2 MathML

2.1 表現形式記述と意味形式記述

MathML には、表現形式記述と意味形式記述の 2 種類の記述方法が存在する。表現形式記述は表現に特化した記述方法であり、数式を表示することのみが重要である状況、例えば閲覧のみを目的とした Web ページに数式を掲載する場合等に便利である。一方で、意味形式記述は意味に特化した記述方法であり、数学的意味をコード化することが重要な状況で便利である。例えば、読者が数式をコピーして数式処理のソフトウェアにペーストして処理を行うといったことが可能になる。表現形式記述は多くのブラウザでサポートされているが、意味形式記述をサポートするブラウザは少ない。そのため、多くの Web ページでは表現形式記述で数式が記述されている。

2.2 表現形式記述の問題点

表現形式記述では同一の見た目に複数の記述方法が存在するため一意でない表現で記述できてしまい、数式の意味が正しく伝わらないという問題点が存在する。例えば、「 $ax+1$ 」という見た目の数式には表 1 に示したように 3 通りの記述方法が存在する。A の記述方法では、乗算を表す演算子 `⁢` を記述することで「 ax 」が「 $a \times x$ 」を意味することが明確に表現されている。B の記述方法では、`⁢` が省略されており、「a」と「x」が乗算されているのかがどうか明確で無い。C の記述方法では、「 ax 」が 1 つの変数として扱われている。このように Web ページ上では同じ式に見えても実際の意味は異なるため、数式の意味が正しく伝わらないという問題が発生する。本研究ではこの問題点に着目し、MathML の文書を正規的な記述に変換し出力するアプリケーションの開発を行った。

3 関連研究・先行研究

3.1 関連研究 1

『MathML の表現形式から意味形式への変換およびその応用』(内橋夏実, 2021) [1] 表現形式を意味形式に変換するツールの開発を行なっている。さらに、コンバータツールの応用の例として、数式部分が表現形式で記述されている数学の問題から、

表 1: 数式 $ax + 1$ の記述方法

A	B	C
<code><mi>a</mi></code>	<code><mi>a</mi></code>	<code><mi>ax</mi></code>
<code><mo>&it;</mo></code>	<code><mi>x</mi></code>	<code><mo>+</mo></code>
<code><mi>x</mi></code>	<code><mo>+</mo></code>	<code><mn>1</mn></code>
<code><mo>+</mo></code>	<code><mn>1</mn></code>	
<code><mn>1</mn></code>		

LMS である Moodle 上の STACK を利用した数学オンラインテストの問題を自動生成するツールを開発している。本ツールと併用することで、一意でない表現形式の文書の正しい意味形式の文書への変換や Web ページに記載された数式を活用した数学教育ツールの開発などが可能になる。

3.2 関連研究 2

『視覚障害者学習支援のための MathML 変換』(渡辺千晶, 2018) [2] 視覚障害者の数学学習支援のために、数式を含む文書を既存のスクリーンリーダーを用いて学習可能にするための文書変換アプリケーションの提案を行っている。

3.3 先行研究

「MathML における一意な表現形式記述への自動変換」(渡辺裕美, 2020) [3] この研究では、一意な表現形式記述への変換を行う CUI のツールを開発している。一方で本研究では、これを GUI の Web アプリケーションに発展させることで、誰でも利用できる操作性の良いツールを開発した。また、先行研究では同じ変換ルールを適用できる複数の箇所があった場合に、それぞれに対して毎回ルールの選択を行なう仕様となっていたが、これを一括でルールを選択できる仕様に変更することで、変換効率を向上させた。

4 研究概要

4.1 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 [4] 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。具体的には、変換が必要になるケースには大きく分けて以下の 3 つのケースがある。

4.1.1 変数

複数の文字が 1 つのタグにまとめて記述された場合には、それらを一つの変数として扱うか、複数の変数として扱うか、という問題が発生する。例えば「ab」という変数があった場合には、これを 1 つの変数とするか、「 $a \times b$ 」という 2 つの変数の乗算の数式とするかという 2 つの解釈ができてしまう。後者の場合にはまず「a」と「b」という 2 つの変数に分けるという処理

が必要になる。

4.1.2 見えない演算子

MathMLには表2で示した「見えない演算子」が存在する。これらの演算子が記述されていない場合には、関数・乗算・座標など複数の意味で取ることができてしまうので、必要に応じて見えない演算子の挿入を行う必要がある。

表 2: 見えない演算子一覧

Unicode名	コード番号	演算子	短縮形	数式の例
FUNCTION APPLICATION	U+2061	<code>&ApplyFunction;</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			$5\frac{7}{2}$

4.1.3 複数の解釈が可能な数式

MathMLには同一の見た目の数式で複数の解釈が可能な数式が存在する。高等学校学習指導要領に記載されている数式の中では表3に示した6つの数式がこれに当たる。例えば「 e 」は「文字 e 」と「ネイピア数」の2通りの解釈が考えられる。なお、「 y' 」及び「 f^{-1} 」は二つの解釈を区別する記述方法が存在しないため、今回は変換対象外とした。また、「!」も MathML タグの内側にエクスクラメーションマークが記述されるというケースは発生しづらいため、対象外とした。

表 3: 複数の解釈が可能な数式

数式	解釈 1	解釈 2
y'	微分式	記号 $'$
$\frac{dy}{dx}$	微分式	分数式
A^T	転置行列	指数
e	ネイピア数	文字 e
!	階乗記号	エクスクラメーションマーク
f^{-1}	逆関数・逆元	指数式

4.2 変換方法

変換には、PythonのWEBスクレイピング (HTMLやXMLファイルからデータを取得し、解析する) 用のライブラリである Beautiful Soup を用いた。変換の主な手順は以下の通りである。

1. 変換対象の入力ファイルまたは URL をユーザーから受け取り、MathML が記述された箇所を MathML ファイルとして抽出する。
2. タグとそのテキスト部分を解析し、変換対象となる箇所を抽出する。
3. ユーザーに変換ルールの選択を一括で求め、そのデータを渡す。
4. 受け取った変換ルールに基づき変換し、変換後の

MathML ファイルとして出力する。

5. 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。

(1) と (2) を実行するプログラムと (4) と (5) を実行するプログラムの2つを Python で作成した。(3) の処理は1つ目のプログラムから受け取ったデータをもとに ruby で実行し、ユーザーから受け取ったデータを2つ目のプログラムに引き渡している。また、出力ファイルの他に、変換箇所とその内容を記述したログファイルもダウンロードできるようにした。

4.3 同一の変換箇所が複数あるケースの GUI

本ツールをより利用しやすいものにするため、ルール選択時、同一のルールを適用できる箇所が複数ある場合には、一括で同一ルールを選択したり、そこから特定の箇所だけ別のルールを選択したりできるようにした。

5 まとめと今後の課題

本研究では、表現形式記述の MathML を定期的で一意的な記述に変換する Web アプリケーションの開発を行なった。

今後の課題としては、今回変換対象外とした数式の変換を可能にすることやルール選択時の UI を該当箇所がコードのどの部分に当たるのかを視覚的に認識しやすいようなものにするなどが挙げられる。

参考文献

- [1] MathML の表現形式から意味形式への変換およびその応用, 内橋夏実, お茶の水女子大学大学院 人間文化創成科学研究科 理学専攻 情報科学コース 修士論文, 2021
- [2] 視覚障害者学習支援のための MathML 変換, 渡辺千晶, お茶の水女子大学大学院 人間文化創成科学研究科 理学専攻 情報科学コース 修士論文, 2018
- [3] MathML における一意的な表現形式記述への自動変換, 渡辺裕美, お茶の水女子大学大学院 人間文化創成科学研究科 理学専攻 情報科学コース 修士論文, 2020
- [4] 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> (2021/9/2/ 閲覧)