

MathML に関するライブラリとその応用について

黒田 拓

KURODA HIRAKU *

北海道大学大学院理学院数学専攻

DEPARTMENT OF MATHEMATICS, HOKKAIDO UNIVERSITY

1 MathML

MathML は数式を記述するための XML 応用言語であり、W3C によって設計された [5]。数式を扱うソフトウェア間でのデータの交換などいくつかの目的のために設計されたものであるが、本稿では MathML を Web ページ中に含めることで数式を含む文書をインターネット上で公開することが出来る点に着目している。

MathML は Web ページのソースである XHTML 文書に含めることができ、これを対応する Web ブラウザーで閲覧すると、きれいに描画された数式を含む文書を見ることができる。MathML に対応する Web ブラウザーとしては、Mozilla Firefox[3]、Microsoft Internet Explorer[2] などがある。Firefox はそのままの状態でも MathML に対応しており、直ちに MathML(およびそれを含む XHTML 文書)を閲覧することができる。ただし、たいいてい場合は数式を表示するためのフォントがシステムに含まれていないので、最良の表示をえるためには Mozilla のサイトの MathML に関するページ [4] などから追加するための数式フォントを入手する必要がある。図 1 は、Firefox による表示の例である。Internet Explorer は本来は MathML に対応していないが、プラグインである MathPlayer[1] をインストールすることで MathML による数式を表示することができるようになる。

2 MathML

筆者が開発した MathML ライブラリ [9] は、オブジェクト指向スクリプト言語 Ruby でのプログラム開発において、 \LaTeX の記法に従って記述された数式を MathML に変換する機能を提供するライブラリである。リスト 1 は、本ライブラリのもっとも簡単な使用例のひとつである。

本ライブラリは、以下のような特徴を持つ。

純 Ruby Ruby でのみ作成されており、C コンパイラ等によるコンパイル作業が不要である。したがって、Ruby がインストールされている環境であればどこでも使用することができる。

\LaTeX ソースから抽出した記号命令の組み込み \LaTeX のソースファイルである `latex.ltx` と `latexsym.sty`、`fontmath.ltx`、さらに \mathcal{AMS} が提供する `amssymb.sty` と `amsfonts.sty` から記号を表す `\alpha`、`\beta` などの命令を抽出して一覧を作成し、対応する記号を表す MathML への変換処理を作成している。

*hiraku@math.sci.hokudai.ac.jp

array 環境 \LaTeX の array 環境を可能な限り再現できる。特に、 \circ 記号による項の挿入や|記号、 \hline 命令による罫線をサポートしている。図 2 ではそのような array 環境の例と、変換した MathML を表示した例を示している。

newcommand, newenvironment によるマクロ 既存の命令や環境を組み合わせる新たに命令・環境を定義する \backslash newcommand 命令と \backslash newenvironment 命令をサポートしている。パラメータやオプションを指定することも \LaTeX と同様に可能である。

Ruby による拡張 既存の命令・環境の組み合わせでは実現できない命令・環境を MathML に変換するために、Ruby による新しい変換処理を作成し、ライブラリに追加することができる。詳細は後述する。

SimpleLaTeX による既存ソフトへの組み込み 既存の Ruby によるソフトウェアにおける文書処理 (特に Wiki 等の記法に従ったプレーンテキストから HTML への変換処理) に、数式部分の MathML への変換を組み込みやすくするためのサブルーチン `MathML::LaTeX::SimpleLaTeX` を提供している。これについても詳細は後述する。

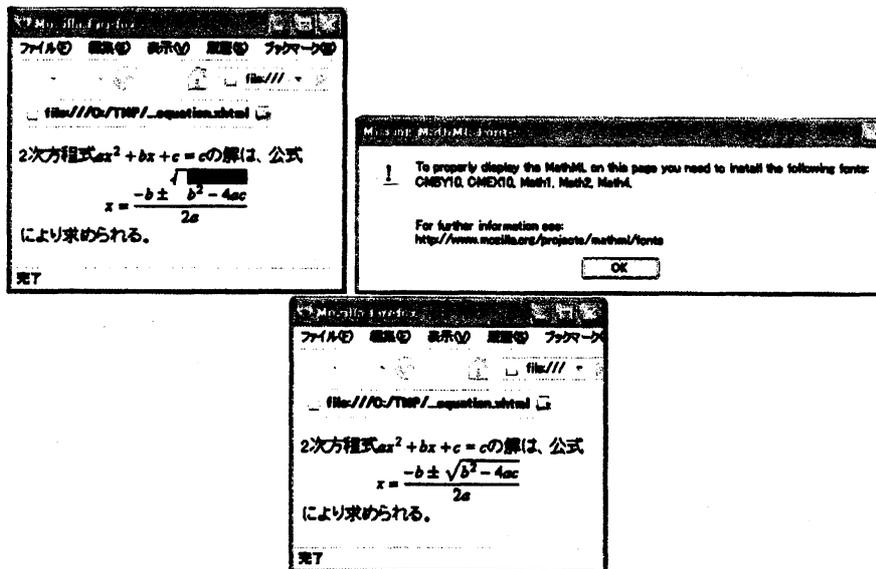


図 1: Firefox での MathML を含む Web ページの表示

上段左: MathML による数式を含んだ Web ページを Firefox を使用して表示している例。システムに MathML のための数式に関するフォントがインストールされていないため、根号の表示が乱れている。上段右: フォントがインストールされていない状態で表示される警告ウィンドウ。Mozilla のサイトからフォントを入手してインストールするようにとの指示が記載されている。下段: 数式フォントをインストールした状態で MathML を含むページを表示している。根号も適切に描画されている。

```
#!/usr/bin/ruby
require "math_ml/string"

puts 'ax^2+bx+c=0'.to_mathml
```

リスト 1: MathML ライブラリの簡単な使用例

MathML ライブラリが提供する、Ruby の文字列クラスを拡張して直接 MathML へと変換する機能を使用した例。L^AT_EX の数式を記述した文字列に対して `to_mathml` メソッドを呼び出し、直接 MathML へと変換している。

```
\newcommand{\cbrt}[1]{\sqrt[3]{#1}}
```

リスト 2: 3 乗根を表す命令の `newcommand` 命令による実装

L^AT_EX の既存の命令を組み合わせて、3 乗根を表す命令を実装している。

3 MathML ライブラリの拡張

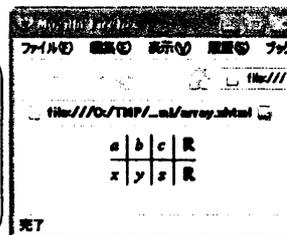
MathML ライブラリでは、ライブラリを利用する開発者が L^AT_EX の新しい命令を Ruby で実装し、追加することができる。

本節では 3 乗根を表す `\cbrt` 命令を具体例として、Ruby による命令の追加の概要を解説する。参考のため、これを L^AT_EX の既存の命令を組み合わせて作成した例をリスト 2 に挙げる。

新しい L^AT_EX の命令は、適当に名前をつけたモジュールの内部で定義されるメソッドの形で実装する。メソッド名は、`cmd_`(作成する命令の名前) である。リスト 3 では、1 行目でモジュール `CubeRoot` の定義が始まり、`cmd_cbrt` メソッドの定義を記述している。

ライブラリが出力する MathML の要素は、各要素に対応するクラスが定義されている。既存の要素の組み合わせで新しい命令に対応する数式を表現する場合は、これらのクラスを利用することで処理の記述が簡単になる。リスト 3 の 3 行目では `mn` 要素に対応するクラスを、6 行目では `mroot` 要素に対応するクラスを使用している。

```
\begin{array}{c|c|c|@{\R}}
a & b & c \\ \hline
x & y & z
\end{array}
```

図 2: MathML ライブラリが変換可能な `array` 環境と、変換結果を表示した例

左: MathML ライブラリが適切に変換することが出来る、項の挿入や罫線を含む `array` 環境の例。右: 変換後の MathML を Firefox で表示した例。挿入された項や罫線が正しく変換されて描画されている。

```

01 : module CubeRoot
02 :   def cmd_cbirt
03 :     cb = MathML::Number.new
04 :     cb << "3"
05 :
06 :     r = MathML::Root.new(cb, parse_any)
07 :   end
08 : end
09 :
10 : ps = MathML::LaTeX::Parser.new
11 : ps.add_plugin(CubeRoot)
12 : ps.add_commands("cbirt")
13 :
14 : puts ps.parse('\cbirt{x}')

```

リスト 3: Ruby による `cbirt` 命令の作成と登録

Ruby による `\cbirt` の実装。各行の行頭に振られた行番号は参考のためのものであり、実際には記述しない。また 10 行目以降は、実際に変換クラスのインスタンスに命令を登録し、使用している様子である。

作成する `\cbirt` 命令は、それに続く以下のいずれかを根号の中の数式として扱う。

- 1 文字の英数字
- バックスラッシュ `\` から始まる命令
- `{...}` で囲まれた 1 つ以上の数式

MathML への変換としては、`\cbirt` 命令に続くこれらの数式を MathML に変換し、それを内容とする `mroot` 要素を出力することになる。Ruby ライブラリではこのような目的のために、

1. `\cbirt` 命令に続く文字列が上に挙げたもののいずれに該当するかを判別し
2. MathML に変換する

という処理を実行する `parse_any` メソッドを提供している。リスト 3 の 6 行目では、`parse_any` メソッドの返り値 (命令の後に続く数式を変換した MathML。根号の中の数式を表す) と、数値 3 を表す MathML 要素 (何乗根であるかを表す) とを指定して `Root` クラスのインスタンスを作成することで、指定された数式の 3 乗根を表す MathML を作成している。

4 MathML::Util::SimpleLaTeX

最後に、MathML ライブラリの応用例として掲示板などの CGI プログラムへ MathML 変換処理を組み込む方法について、いくぶん抽象的な例とともに解説する。具体的な応用例については、ウェブ日記/ログツールである `tDiary`[7]、Wiki エンジンである `Hiki`[6] に MathML 変換の機能を提供するプラグイン [10][8] を公開しているのでこれらを参照されたい。

ここでは以下のような処理を行う CGI プログラムを考える。

1. ユーザからの入力として、特定の記法に従ったプレーンテキストを受け取る
2. 入力されたテキストを HTML として整形する
3. 整形された HTML を出力する

```

input = ....          # ユーザからの入力文字列を代入する変数
html = format_text(input) # HTML への整形を行うメソッドと、変換結果を代入する変数
puts html            # 出力

```

リスト 4: CGI プログラムの例

掲示板などの処理を Ruby で記述した例。各行末尾の # から後はコメントであり、Ruby では無視される。

リスト 4 はそのような CGI プログラムのたまかな処理を記述した極簡単な例である。

このような処理に MathML 変換処理を組み込む事を考えると、最初に思いつくのは HTML 整形の前か後に、扱っている文字列から数式部分を探し出して変換するというものである。しかしこの方法には以下のような問題がある。

HTML 整形の前に数式変換を行う たいていの場合、HTML 整形では入力された文字列中に HTML の記述に用いる <、> などの記号をエスケープする処理を行う。このため、先に MathML への数式変換を行ってから HTML 整形を行うと、出力されるのは MathML のソースを表すような HTML 文書になってしまう。

HTML 整形の後に数式変換を行う Wiki 記法などでは、一部の記号を文の装飾や箇条書きを表す特別なものとして扱う。このため、HTML 整形が施された文字列では、入力されていた数式がそのまま残っているという保証がない。

したがって、HTML 整形を行うような CGI プログラムに数式の MathML への変換処理を組み込む場合は、次のような手順が必要になる。

1. 入力された文字列から数式部分を抽出する
2. 数式を抽出した後の入力文字列には、そこに数式があったことを示す (後の HTML 整形で処理されない) 記号を代わりに挿入する
3. 数式部分を記号に置き換えた入力文字列の HTML 整形と、数式部分の MathML への変換を行う
4. 整形後の HTML 文字列 (数式の位置を示す記号はそのまま残っている) に、数式から変換した MathML を戻す

MathML ライブラリでは、このような処理を簡単に記述するためのサブルーチンクラスとして `MathML::Util::SimpleLaTeX` クラス を提供している。リスト 5 は、`MathML::Util::SimpleLaTeX` を使用してリスト 4 を数式に対応させた例である。

まず 4 行目で、入力文字列から数式の抽出 (`decode` メソッド) を行い、数式部分を特別な記号に置き換えた文字列 (変数 `encoded`) と抽出した数式を MathML に変換したリスト (変数 `data`) を得ている。

6 行目では `format_text` メソッドによって `input` を整形する代わりに `encoded` を整形している。`encoded` では数式があった部分が ASCII コード 0 番の記号に置換されており、この例での `format_text` メソッドは

```

01 : input = ...
02 :
03 : sl = MathML::Util::SimpleLaTeX.new
04 : encoded, data = sl.encode(input)
05 :
06 : html = format_text(encoded)
07 : html_with_mathml = sl.decode(html, data)
08 : puts html_with_mathml

```

リスト 5: MathML::Util::SimpleLaTeX の使用例

この記号を特別扱いしないものとして実装されているので、整形後の文字列である `html` には、この記号がそのまま残る。もし `format_text` がこの記号を特別扱いし何らかの加工を行ってしまうようなものならば、`MathML::Util::SimpleLaTeX` に対して数式があった箇所を表すものとして別の記号を指定することもできる。

7 行目では整形された文字列 `html` に、数式から変換した MathML を戻す作業を行っている。最終的な文字列が `html_with_mathml` に代入されており、これは MathML を適切に含んだ HTML 文書である。

5 まとめ

MathML ライブラリは、 \LaTeX の数式を MathML に変換する処理を提供するライブラリである。豊富な記号命令が組み込まれており、`\newcommand` 命令と `\newenvironment` 命令によるマクロ、Ruby による拡張などを行うことができる。

また、既存の CGI プログラムなどに組み込むためのサブルーチンが提供しており、これを利用したプログラムによって、MathML による数式を含む HTML 文書を容易に作成することができるようになる。

参 考 文 献

- [1] Design Science. MathPlayer.
<http://www.dessci.com/en/products/mathplayer/>.
- [2] Microsoft. Internet Explorer.
http://www.microsoft.com/windows/ie_intl/ja/.
- [3] Mozilla Japan. Firefox.
<http://www.mozilla-japan.org/products/firefox/>.
- [4] Mozilla Japan. MathML が使用可能な Mozilla のためのフォント.
<http://www.mozilla-japan.org/projects/mathml/fonts/>.
- [5] W3C. Mathematical Markup Language (MathML) Version 2.0 (Second Edition).
<http://www.w3.org/TR/2003/REC-MathML2-20031021/>.
- [6] たけうちひとし, Hiki 開発チーム. Hiki.
<http://hikiwiki.org/>.

- [7] ただただし, 他. tdiary.
<http://www.tdiary.org/>.
- [8] 黒田 拓. Hiki xhtml 化キット.
<http://www.hinet.mydns.jp/~hiraku/hiki/?x-hiki>.
- [9] 黒田 拓. Ruby 用 mathml ライブラリ.
<http://www.hinet.mydns.jp/~hiraku/hiki/?mathml.rb>.
- [10] 黒田 拓. tdiary xhtml 化キット.
<http://www.hinet.mydns.jp/~hiraku/20010102.html>.