MathML における表現形式から意味形式へのコンバータツールの開発

理学専攻・情報科学コース
1840644
荒川　玲佳

1 はじめに

近年，電子形式での数式データの利活用の需要が高まっている．ウェブ上で数式を表現するために MathML というマークアップ言語があり，二種類の形式が存在する．一つは表現形式という，数式の位置情報のみで数式自体の意味を持たない形式で，ウェブページの多くがこの形式で記述されている．しかし，表現形式では数式の計算処理が困難である．そこで，もう一つの意味形式へとソースの形式変換を行い，変換結果のコードを出力するツールの開発を試みる．これは表示のみに留まっていたデータを，数式として再利用性の高いデータにすることを目的とする．表現形式の多様な記述パターンに対応したデータ構造，さらに独自に定義した構文規則の組み合わせで変換を実現した．この研究により，数式検索の実用化などに繋がる．

1.1 MathML について

W3C[1] により数式のウェブ表記用 XML 言語として MathML（Mathematical Markup Language）[2] が開発された．これまでの固定レイアウト型に限られた数式表示からリフロー型[1]な表示を可能にした．MathML には表現形式と意味形式がある．前者は，数式を TEX のように視覚的な情報によって表示する形式である．後者は，MathDTD により演算子や引数を厳密に個々に型づけして記述し計算機が解釈可能な形式である．

1.2 関連研究

先行研究に意味形式への変換を試みる事例がある [3, 4]．表現形式記述の自由度の高さにより生じる意味構造の曖昧さについて説明し，一意に解析木を作成できないことを分析している．そして，sed を用いて構造として考えられるパターンを表示し，人の手による修正を加えて解析木を構築する方法を提案している．しかし，変換処理の詳細や物理構成が明記されていなかった．

2 研究方法

2.1 提案方法

表現形式は数式要素をラベル付けする型の制約が弱く，表示には影響しない演算子の記述を省略する場合が多い．これは数式の意味構造に曖昧性が生じる問題となる．また，演算子の種類によって異なる数式記法が混在し，構文木の作成を難しくする問題がある．これらに対し以下の方法を提案する．自動変換の各処理の詳細については次節以降で説明する．

方法 1 中間表現として階層構造リストと構文木を作成

方法 2 演算子の欠落が生じている場合は，欠落箇所を特定して自動的に欠落情報を付加

提案した方法で演算子の省略に関しては，データ構造とデータの型の並びの処理で欠落箇所の特定と付加が容易となる．また，演算子に異なる数式記法が混在することに関しては，記法に応じてリストの構成の仕方を線型と階層型に変える．これにより演算子の適用範囲を物理的に表現することが可能となる．コンバータツールの処理の流れを図 1 に示す．数式を一式ずつ変換処理を行う点で，インタプリタに近い処理系となる．

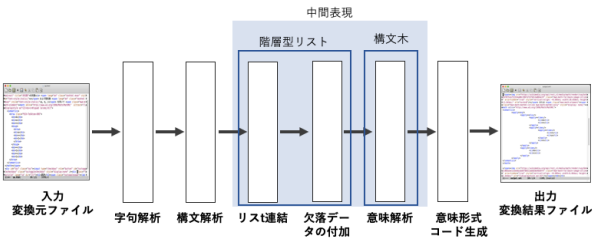

図 1: コンバータツールの全体処理

2.2 字句構文解析

数式は MathML インターフェース要素を対にして，その内部に記述される．字句解析はこの要素を認識したら，構文解析に移り数式コンテンツの解析処理を行う．数式情報を保持する最初の中間表現は階層型リスト構造で，その最小単位となるオブジェクトデータとする．演算子や引数のデータは，タグまたはそれに挟まれた要素を個々に識別し，生成されたオブジェクトデータに認識された型と要素などの情報を格納する．

2.3 階層リスト構造の作成

字句構文解析で生成されるオブジェクトデータをリスト状に連結する．本研究では MathML の表現形式用の拡張 BNF を定義した．さらに異なる情報を保持する三つのスタックを用いて連結を行う．中置記法の演算子はデータを線型に連結し，前置記法の演算子は自身のデータを分岐元として，引数データを分岐先に連結する．さらに，前置記法演算子は引数が一つと二つのものがある．分岐数もそれに応じる．

2.4 型チェックと欠落情報の自動付加

連続したデータ間で演算子が欠落している場合

相互のデータの型を参照したときに引数型が連続する場合は，引数データの間に掛け算情報が欠落していると認識する．このとき，図 2 上のように掛け算情報を保持するオブジェクトデータを生成して挿入する．

複数変数が複数格納されている場合

正しい表現形式では一変数の掛け算，例えば数式 xy は x と y は個別にラベルづけして，その間に見えない掛け算を次のように記述する．
`<mi>x</mi><mo>&InvisibleTimes;</mo><mi>y</mi>`，
しかし，実際は`<mi>xy</mi>`のようにまとめて記述されることが多い．この場合は図 2 下のように格納され，一つに格納された変数要素を分解して連結し直す．

[1] リフロー型：表示する端末の画面サイズの変更に合わせて，テキスト及びレイアウトが自動で変わる方式

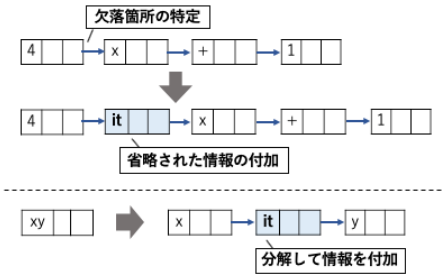

図 2: 欠落箇所の特定と演算子情報の挿入

2.5 意味解析・構文木の作成

数式の意味構造を持たせるために構文木を作成する．まず二種類のスタックを用いて，リストのデータを演算子の結合優先度に基づいて後置記法の並びに変換する．このとき，ネストレベルの深いリストから並び替えを行い，一番浅いリストのデータの並び替えが完了すると最終状態となる．スタックの最上位を根としてデータを木構造状に連結し直すと，構文木が構築できる．表 1 で優先度因子の数字が小さい方が優先度が高い．

表 1: 演算子の結合優先度因子

優先度因子	演算子の種類
PR1	乗算
PR2	加算，減算
PR3	その他の定義した演算子

2.6 意味形式への書き出し

構文木で保持された表現形式の情報から，意味形式コードの生成を行う．木を行きがけ順に辿り，各ノードに保持された型情報と対応する意味形式タグに逐次置換する（図 3 左）．葉である引数要素はそのまま書き出す．意味形式では演算子の適用範囲を明示的に表現するための `apply` タグが定義されており，各部分木のトラバースの最初と最後に `apply` タグを書き出す．

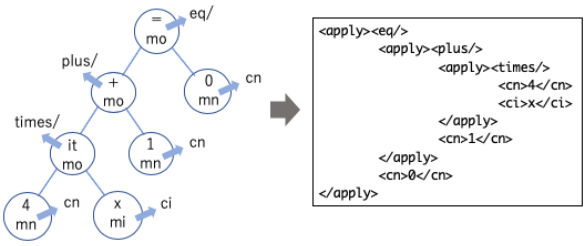

図 3: 数式 $4x + 1 = 0$ の構文木から意味形式への変換

3 結果

コンバータの構成

本コンバータは C 言語でマルチモジュール構成で実装し，数式を一つずつ逐次的に変換する．未定義の演算子がある場合はエラーメッセージを表示する．ブラウザの開発ソースを入力で受け取り，以下のコマンドで実行される．結果は新規ファイルに `out.html` で取得が可能.（実行コマンド：`Converter` 変換元ファイル）

変換結果の一例

変換した数式は判別式 $b^2 - 4ac > 0$ である（図 4）．この数式の変換元ソースでは，定数 4 と引数 ac の間，変数 ac の間で乗算の欠落が発生している例である．

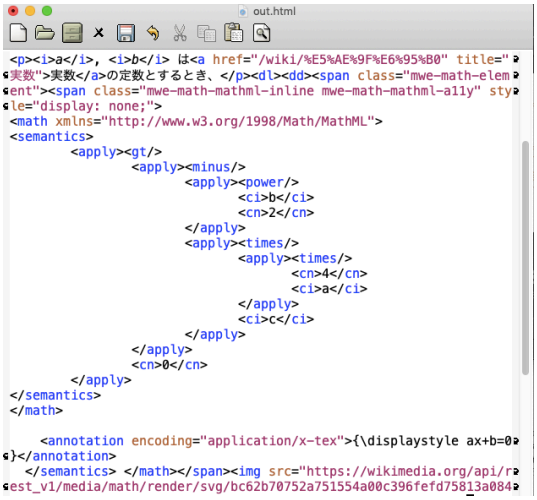

図 4: 意味形式への変換結果：数式 $b^2 - 4ac > 0$

4 考察

図 4 の変換結果は，欠落した演算子情報を `<times/>` で置換できており，変数も分割され同様に変換できていることが確認できる．

他にも数式例を 40 個準備して，実行結果の検証を行った．数式には中間表現において単一の線型リストや複雑な階層型リストを構成するもの，単項演算子である．また，想定できる乗算の欠落した記述状態を含む．これらの数式も変換が正しくできたことを確認した．

変換可能な演算子は，加減乗，分数，指数，平方根，根号，等号，不等号である．

5 まとめ

今後の課題

本ツールは変数が一文字であること，引数が二つまでの演算子を前提とする．この前提条件に対応しない演算子に関しては，リストの分岐数と，構文木の構造を増やすなどの拡張が必要となる．

技術展望

意味形式コードは `Mathematica` にインポートすることが可能で，複雑な数式のグラフ描画することに役立つ．また，本コンバータを改良してブラウザにプラグインすることができれば，関連研究にある数式検索の実用化や視覚障がい者の数学学習支援などに繋がる．

参考文献

[1] W3C，http://www.w3.org/(2020/1/1 閲覧)

[2] MathML Wolfram Language Documentation, https://reference.wolfram.com/language/XML/tutorial/MathML.html(2020/1/2 閲覧)

[3] 石山寿子，高野文子，原俊介，大武信之：XML における数式の表示形式から意味形式への変換，電子情報通信学会技術研究報告．ET，教育工学，Vol. 101，No. 506，pp. 23-30(20011208).

[4] 大武信之，秋山富貴子：Web 上における数式表現の意味形式記述への半自動変換システム開発，国際経営・文化研究．Vol. 7，No. 1，pp. 29-41(200301).