

数学学習支援システムにおける解答の正誤判定のための数式表現正規化手法 Normalization of Math Expression for Discriminate between Correct and Incorrect Answers in Mathematics Learning System

金光 隆広† 山田 剛一† 絹川 博之†
Takahiro Kanamitsu Koichi Yamada Hiroshi Kinukawa

1. はじめに

Web を用いた数学学習支援の方法として様々なものが開発されているが、紙のテストと同じく記述式の解答を用いた学習支援システムは少ない。その理由としてオンラインテスト等対話型の処理を行う際、組版された数式を Web 上のテキストを用いた情報として扱うことが難しいことが挙げられる。現在ではマークアップ言語である MathML [1]が開発され、解釈することが可能なブラウザも存在している。MathML を用いて記述式解答による対話型の処理が可能になった。しかし、記述式解答では、数式に関して複数の表現方法ができるため、模範解答の数式と解答の数式との単純な比較照合で判定することは難しい。本研究では高校数学の範囲で数式を含めた記述式解答が可能なインタフェースを用い、学習者に入力させた記述式解答と教師が作成した模範解答の意味が同じであれば正答と判定する方法として、両者の数式の表現を正規化する手法を提案する。合わせて、テキスト形式ではなく構造を持った数式として解答を扱うことで具体的に誤っている部分を検出することを可能とする。

2. 数学学習支援システムの数式の扱い

2.1 記述式解答を可能とする数式入力方法

現在 Web と数式の親和性は高くなく、表現方法が限定されている。例えば、数列の和の公式として

$$\sum_{k=1}^n k = \frac{1}{2}n(n+1)$$

が用いられるがこれを Web 上で人間が入力する場合、

$$\Sigma [k=1 \rightarrow n]k=(1/2)n(n+1)$$

等、数式を文字列で表現するための知識が必要となる。また、文字列で表す際に本来必要なかった部分に括弧を入れる必要があるため新たに必要な括弧を入れ忘れ、計算機で異なる解釈がされる可能性がある。これらの原因は Web で文字を記入する際に最も汎用的に使われているテキスト入力では数学の記号が文字として存在せず、上付き下付きといった記法も使えないため数式独特の表現方法を入力することができないことにある。そのため、Web を用いた学習支援システムの対話型処理に関して、TeX などの専門知識を持っていない人が使うためには WYSIWYG 形式のインタフェースによる支援が必要である。本稿では Web 上での数式の入力に BriEdima[2] を用いる。入力された数式をデータとして処理を行うために MathML に変換する。MathML を選択したのは Web ブラウザ上でマークアップ言語を用いて数式の形に解釈することが可能であることと数式の構造を木構造で表現できるためである。

2.2 記述式解答を用いた正誤判定

計算機を使った自動正誤判定は基本的に文字列や番号等で模範解答の設定がされており、学習者が入力した文字列や選択した番号と比較を行うことで正誤判定を行っている。そのため、入力する文字列や選択する番号等の解答は一意である必要がある。しかし、数学において解答としては一意であるがその表現方法が複数の場合が存在する。例えば模範解答の例として

$$\frac{1}{2}n(n+1)$$

が指定されている時、ユーザの解答として

$$\frac{n(n+1)}{2}$$

とすることも可能である。双方の式とも同じ意味の解答であるが、構造が異なっているため正誤判定を行うと不正解と判定される。そのため、記述式解答を取り入れた学習支援システムで自動正誤判定を行う場合、数式の正規化が必要である。正規化とは同じ意味の数式群を、代表する 1 つの表現に変換することである。

3. 数式表現正規化手法

3.1 正規化の流れ

本研究では学習支援システムを使うユーザによって作成された数式を用いて計算機内で正規化を行うために、図 1 の流れを用いた処理を行う。数式を MathML へ変換し、その構造から後述する特徴を抽出し、構造を走査し、構造の変換処理を行う。最後に構造変換を行ったかを確認し、構造変換を行っていれば再度現在の構造から特徴の抽出を行い、処理を繰り返す。

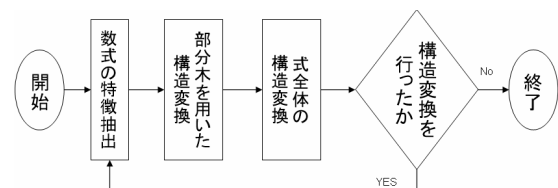


図 1. システムを用いた学習者の学習の流れ

3.2 実現すべき機能

3.2.1 数式の特徴抽出

インタフェースにより入力された数式を MathML に変換し、構造の特徴抽出を行う。最初に MathML に変換した数式の一部を図 2.として示す。また、式全体を木構造として表現したものを図 3.として示す。MathML で分数を表す “<mfraction>” のように数式独自の表現に用いられる要素を “関数ノード” と定義する。また、数や算術演算子などの識

†東京電機大学大学院 未来科学研究科
Graduate School of Science and Technology for Future Life,
Tokyo Denki University

別子の表現に用いられる要素を“変数ノード”と定義する。図 3. では根ノードを楕円形,関数ノードを四角形,変数ノードを菱形,変数ノードに格納されている内容を丸形で表現する。最後に後述する構造変換を行う際,木の一部分をひとくくりにとめる処理を行うがその関数ノードと変数ノードが混在するノードを“式ノード”と定義し,図では変換前の式ノードを点線,変換後の式ノードを六角形で表現する。図 3.における式ノードは最下層の関数ノードを基点とした場合の式ノードの範囲である。この構造を用いて幅優先探索にて走査を行い,根ノードと関数ノードをリスト化する。なお,関数ノードの子が式ノードのみであった場合はリストに登録しない。このリストに登録されたノードについてそれぞれを基点として以下の特徴の抽出を行う。

- 基点より下層のノード全体の数
- 基点から派生しているノードの数
- 基点から最下層のノードまでの深さ
- 基点から下層の要素リスト

```

<math>
  <mfrac>
    <mrow>
      <mn> 1 </mn>
    </mrow>
    <mrow>
      <mn> 2 </mn>
    </mrow>
  </mfrac>
</math>

```

図 2. $\frac{1}{2}$ を MathML で表現

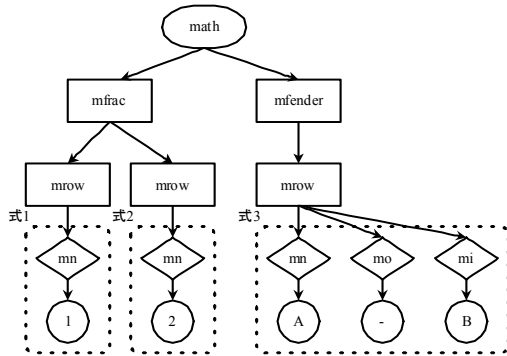


図 3. $\frac{1}{2}(A-B)$ を木構造で表現

3.2.2 部分木を用いた構造変換

幅優先探索によって 1 番最後に走査した関数ノードから順番にその関数ノードを根ノードとした部分木を生成し,上記で得た特徴を用いて構造の変換を行う。この段階では以下の処理を行う。

- (1) 多項式における順序の整理
- (2) 公式を用いた変換
- (3) 部分木を 1 つの変数ノードとして置き換える処理

多項式における順序の整理では,単項式のリストを作成し指数と係数が高いものから整理し,順序の決定を行う。次に公式を用いた変換については特徴と変換が必要な数式の特徴のマッチングを行い,一致したものに登録されている構造を入れ替える処理を行う。最後に部分木を 1 つのノードとして置き換える処理では部分木の根ノードより下位のものをまとめ,式ノードに変換し,元の木の該当部分に置き換える。

3.2.3 式全体の構造変換

公式を用いた変換を用いる際,ある関数ノードが違う場所に移動を行うことや親子関係が逆転する場合が存在する。その変換を行うために根ノードを基点として全体の構造を走査し,それらの特徴と一致した場合,この段階で処理を行う。変換方法は 3.2.1 で示したものと同様である。

図 4-1.に変換前,図 4-2.に関数ノードが移動した後の例を挙げる。これらの処理を行った際,最初に変換処理を行った関数ノードの特徴が使えなくなるため,構造をそのままにして再度数式の解析を行う。この流れを繰り返し,構造変換を行い,全てのノードが式として式ノードに変わった時に正規化を終了する。

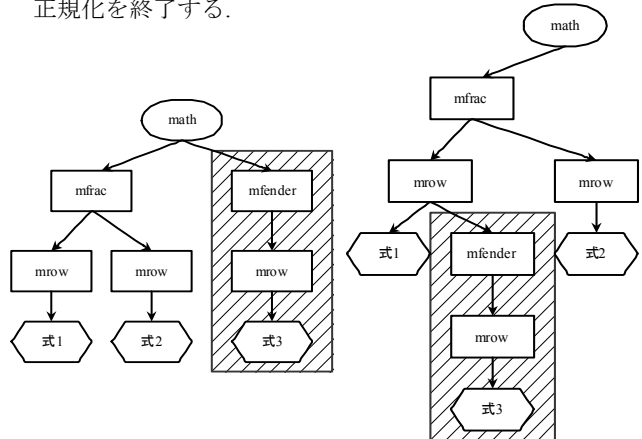


図 4-1.全体の構造

図 4-2.一部を移動した結果

4. 考察

MathML を用いて自動正誤判定を行う際に,正規化手法を導入することができたが,構造の変換に際して変換可能な形が複数ある場合がある。その場合は変換した後にさらなる構造変換が出来る場合が考えられるため,複数の出力結果から正誤判定を行うことが必要である。また,導出の過程で解答を行なっている場合,数式処理をして計算機内で導出する機能を備えていないため,部分点などの評価を行うことができないことが問題である。

5. おわりに

5.1 得られた成果

本稿では数学学習支援システムにおける記述式解答を行う際,自動正誤判定を行うための正規化手法について論じた。また,数式入力インタフェースを用いることにより専門的な言語を用いないで学生でも Web 上に数式を入力することが可能になった。

5.2 今後の課題

構想を実現するために様々な段階の機能が未実装であり,多くの課題が残されている。特に構造を変換する際に必要とされる数式公式データベースの設計を行うことが必要であり今後の課題として挙げる。

参考文献

[1] MathML : <http://www.w3.org/TR/MathML2/>
 [2] 中尾,森光,村尾”Wiki や blog でも数式を:BriEdima を使って”数理解析研究所講義録,vol.1624,pp.67- 83,2009