

線形文字列変換による機械学習型数式入力インタフェースと編集機能の設計

福井哲夫[†]

武庫川女子大学[†]

1. はじめに

本研究は数式情報をデジタル化するための数式入力ユーザインタフェースに関するものである。入力された線形文字列から機械学習型辞書を使って、ノードが数式候補であるような決定木を算出し、ユーザとの相互作用によって所望する数式を構築する新方式を2011年8月に提案した^[1]。その後、そのアルゴリズムの定式化と効果・効率についても報告した^{[2][3]}。本研究のねらいは、中学・高等学校に導入されようとしている学習者用デジタル教科書の数学科目などにおいて、数式入力を平易にすることによって個別学習を支援することにある。本発表では、提案方式による数式入力ユーザインタフェースのユーザビリティを向上させるための編集機能の設計について報告する。

2. 従来技術

数式入力のための従来技術として次が代表的である。

- 1) 線形文字列コンパイル方式 (例: LaTeX, MathML など),
- 2) GUI テンプレート方式 (例: 数式エディタ^[4]など),
- 3) 手書き入力方式 (例: Windows7 の数式入力パネルなど)

1)は線形文字列を入力し、整形プログラムによって数式を構築するタイプで、文法が複雑である。編集はソーステキストに戻って行うので、文書処理ソフトウェアの編集と全く同一である。2)は数式構造を表すテンプレートをGUIによって選択しながら構築するもので、長い式の編集では煩わしい場合がある。3)の場合、入力は最も自然であるが、誤認識や入力ミスによる編集操作は文書編集に比べ煩わしく感じる。いずれの方式も理数系を専門としない一般ユーザにとっては依然使い易いとは言いがたい。この理由は、システムが完全に数式構築できる情報をユーザの操作開始時に要求する仕組みにある。

3. 提案方式の概要^[1]

そこで本研究では、初期入力を平易にするような新しい数式入力方式を提案する。

3.1 線形文字列表記法 (新提案)

まず初期入力のための数式表記法を次のように定める。

表記法 所望する数式を、ユーザが読む順番にその数式要素に対応するキー文字 (列) によって区切りなく線形に並べる。

本表記法に従った線形文字列形式の例を表1に示す。例えば、変数 a や α などはいずれもキー“a”で表す。この

A design of the editing function of an interactive interface for equation edit using the linear string formatting from the key dictionary by machine learning

[†]Tetsuo FUKUI

Mukogawa Women's University

ように、キー文字 (列) は ASCII コードからなり、数式要素を連想しやすい頭文字や LaTeX などに準じた文字列を採用してもよい。また、演算子の分数記号は“/”, 積分記号は“int”などで表し、その前後にはオペランドの線形文字列が並ぶ。ユーザの負担を少なくするために、キーを並べる順番として、人がその数式を読む順番を採用してもよい。特に他の表記法と大きく異なる点は、暗黙積や冪乗演算のように表記されない記号は、線形文字列に含めないところである。例えば、式 a^2 は“a2”と表記する。

3.2 提案数式入力方式の概要

このように、本方式で使用する線形文字列表記法は、単純・簡潔になっている。その代わりに、所望する数式を構成している数式記号のスタイルや要素間の区切りや各演算子に対するオペランドの範囲などが省略されており、入力された線形文字列形式の情報だけでは2次元の数式表記が一意的に定まらず、完全にフォーマットすることはできない。そこで、本システムはキー辞書データを用いて数式候補を提示し、仮名漢字変換のように、ユーザに不足情報を補うための簡単な指示を要求する。そのような数式の構築過程を図1に示す。ユーザは数式構成要素ごとに判断すればよいので、複雑な式であっても迷うことはない。すべての構成要素が確定すれば2次元の表記の構築が完了する^[1]。

数式は、最小単位記号自身であるか、演算子と作用するオペランド数式の階層構造 (演算子構造) をもち、木構造で表現される。そのため、数式構築アルゴリズムは仮名漢字変換のアルゴリズムと大きく異なる^[3]。

表1. 線形文字列形式の例

| 所望数式 | 本方式の入力 | LaTeX (従来) の入力 |
|-----------------------|----------|--------------------------------|
| $\alpha = 2$ | a=2 | \Y alpha=2 |
| $\frac{1}{a^2+1} = 2$ | 1/a2+1=2 | \Y frac{1}{a ^ 2+1} \Y doteq 2 |
| $\int_a^b c = 2$ | intabc=2 | \Y int_{a} ^ {b}c=2 |

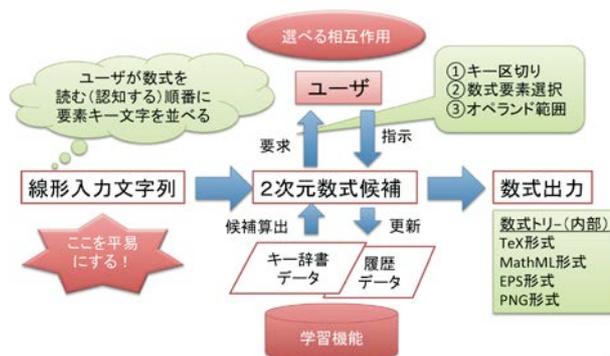


図1. 提案方式の数式構築過程

【進歩点（独創性・優位性）】

このように、本提案方式の独創性は、数式に対する文字の入力の順番をユーザ側の認知行動に対応させた点にある。したがって、従来技術に比べ、

- I. 数式構築のための入力文字列が容易、
- II. 候補を選択すればよいので、数式構築操作が単純、
- III. 数式辞書の機械学習機能により入力効率が向上といった優位性がある。

4. 数式入力インタフェースの実装と評価^[2]

被験者（武庫川女子大学・情報メディア学科 3 年生）10 人に同一タスク問題を、本方式実装システム（MathTouch）と Microsoft（登録商標）Word for Mac2011 ver. 14. 1. 1 に付属する数式エディタ^[4]とで実施してもらい、数式入力時間（タスク達成時間）を測定して比較を行った。タスク問題は次のような分数係数の 2 次多項式である。

$$-\frac{5}{7}x^2 - \frac{5}{8}x + 3, \frac{2}{7}t^2 - 7t + \frac{5}{3}, -7\alpha^2 + \frac{9}{7}\alpha - \frac{5}{4}, -\frac{9}{7}\beta^2 - \frac{7}{4}\beta - 4$$

10 問テストを 3 回実施した後の結果を図 2 に示す。t 検定の結果、有意差 (t(9)=4. 83, p<. 01) があり、このタスクでは、本システムの方が約 1. 75 倍速いことが判る。ただし、両者とも入力ミスのあったデータは除く。

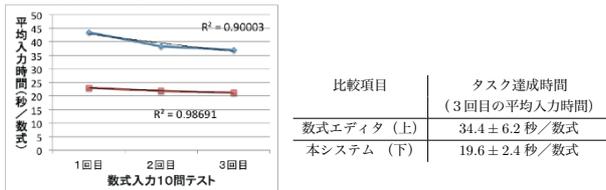


図 2 タスク達成時間の比較（3 回の習熟曲線と結果）

5. 数式編集機能の提案と設計

5.1 編集機能の必要性と方針

前章のように、ユーザは常に正しい入力や操作を行うとは限らずミスを起こす。本システムを数学教育などに応用し実用化するためには、次のような課題解決による数式編集機能が不可欠である。

（課題）入力・操作ミスした場合に、対象箇所のみ修正できること（一から入力し直さなくてよい）

ユーザビリティの向上にはユーザ経験を生かすことが重要とされている。そこで、本システムにおける数式編集機能を次のような方針で開発する。

- 文字編集と同等の機能
- 文字編集と類似の操作性

5.2 文字と数式における編集機能の検討

文字編集機能と対応する数式編集を比較した結果が表 2 である。表 2 ※印の説明は、従来方式 2) の数式編集機能において検討を要する箇所である。

5.3 数式編集機能の設計

前節の検討を踏まえ、次のような編集機能を実現した。

- (1) 表 2-1) ~ 5) の機能
- (2) 演算子構造の削除や挿入も可能にする (図 3 参照)
- (3) 記号や演算子構造の再変換機能 (表 2-6), 図 4 参照)
- (4) 修正内容を数式辞書の機械学習に反映

$$a \underset{\uparrow}{/} b \Leftrightarrow \frac{a}{b} \qquad \frac{1}{\underline{a^2+1}} \Leftrightarrow \frac{1}{\underline{a^2}} + 1$$

図 3 分数挿入・削除 図 4 分数構造（分母）の再変換

表 2 文字と数式における編集機能比較

| 機能 | 文字編集 | 数式編集 |
|-------------------|---------|---|
| 1) 挿入位置 (カーソル) | 文字間 | 数式の前 またはオペランドの挿入位置 (※後者は演算子構造をもつ場合、前者と挿入位置が区別しにくい場合がある) |
| 2) 削除 (バックスペース) | カーソル左文字 | (部分) 数式 (※削除対象を要指示) |
| 3) 挿入内容 | 文字列 | 演算子構造と部分数式 ※演算子構造だけの挿入不可 |
| 4) カーソル移動 | 1 次元的 | 2 次元的 |
| 5) コピー・カット & ペースト | 文字列 | 部分数式 |
| 6) 再変換 | 漢字変換 | 記号・演算子構造 (※不可) |

【解決手段】

機能 (1) のためには、挿入位置をカーソルで表し、削除対象の部分式を囲み枠などの補助表示によって明確にする (図 5)。しかも、この囲み枠表示は機能 (3) の再変換対象の指示にも使い、重要な役目を果たす。カーソルの移動については矢印キーを利用し、本表記法による数式要素入力の順番（読む順番）を採用する (図 5)。

機能 (2) や (3) は従来方式には無い機能であるが、ユーザの部分的な操作ミスを簡単に修正できて大変便利である。本提案方式では数式要素入力後にユーザとの相互作用によって演算子構造を変更・確定できるため、挿入演算子や既存演算子の確定を解除し、既存部分式をオペランドと見なすだけで新規入力と同じアルゴリズムによって容易に実現できる。特に、挿入式の人的エラーにも対処できる設計になっている。

$$ab^2[c \rightarrow ab^1c \rightarrow ab^2c$$

図 5 左矢印キーによりカーソル (I) が移動。

カーソル左の囲みは削除または再変換対象を表す。

5.4 編集機能の動作テストと評価

被験者（本学情報メディア学科 3 年生）17 人に、数式の編集操作に関するタスク実験を実施してもらった。数式特有のカーソル移動や再変換に少し戸惑いはあったが、タスク達成率は 95% であった。アンケート調査した結果、82% の学生から満足度の高い（好感）評価を得た。

6. まとめと今後の課題

以上、本研究で提案した数式編集機能の優位性は次の 3 つである。

- 式挿入編集においても本方式の優位性 I ~ III がある
- 式の記号要素や演算子構造をも再変換編集が可能
- 修正内容を数式辞書の機械学習に反映できる

今後の課題として、中学・高等学校の数学の教科書の数式を扱う練習問題をデジタル化して、自宅の PC や端末で個別学習できるようなシステムに応用したい。

【参考文献】

- [1] 福井哲夫: 数式のインテリジェントな線形入力方式, 数理解析研究所講究録「数学ソフトウェアと教育」, 2012 掲載予定.
- [2] 福井哲夫: 数式のインテリジェントな線形入力方式と評価, 数式処理, 日本数式処理学会, 2012 掲載予定.
- [3] 福井哲夫: 線形文字列変換による対話型数式入力方式の効果, 数理解析研究所講究録「数式処理」, 2012 掲載予定.
- [4] Microsoft (R) co.: マイクロソフト数式エディター 3. 0, <http://office.microsoft.com/>.