

2 Z - 0 3 汎用入出力機能付き数式エディタ開発

岡村 博文 金堀 利洋 鈴木 昌和

九州大学大学院数理学研究科

1 はじめに

ネットワークやコンピュータの加速的な進歩に伴い、数学の研究や教育の分野において、様々な可能性が生まれて来ている。それに歩調を合わせるかのように、TeX や MathML といった各種数式フォーマット、Mathematica のような数式処理システム、各種ワープロに付属の数式入力機能など、コンピュータ上で数式を扱う環境が開発されて来ている。

しかし、現在、コンピュータ上で数式を扱うのは、まだまだ簡単であるとは言えない。TeX や MathML などは、入力された数式の意味を一目ただけで把握することが難しく、入力の誤りを数式を見ながら訂正することはできない。Mathematica やワープロに付属の数式入力機能などは、メニューボタンなどによる入力操作を繰り返さなければならず、その煩わしさがスムーズな入力を妨げ、ユーザーの思考の流れを中断させる可能性が高い。

また、これらのアプリケーション間の互換性の問題も考えられる。TeX による数式を MathML に変換しネットワーク上で処理したり、MathML による数式を数式処理システムで計算したりすることも必要になるであろう。TeX や MathML の自動点訳、自動読み上げのアプリケーションも目の不自由な人たちにとって必要不可欠なものになっている。しかし、このような数式変換プログラムを様々な数式フォーマットの組み合わせの数ほど開発するには、大変な時間と労力が必要となる。

我々は、この問題を解決するために、汎用性のある数式入出力機能を持った数式エディタを開発した。さらに、手書きによる数式入力をサポートし、簡単なペン操作による数式の入力や編集を実現した。これにより、コンピュータ上の数式を直感的に扱うことが可能になる。

また、このシステムの ActiveX によるコンポーネント化により、様々なアプリケーションに簡単に組み込

むことができるようにした。これは、エディットコントロールやリッチエディットコントロールの数式版を提供しようとするものである。この数式エディタのアプリケーションへの利用として、数式処理システムに接続したシステムを開発した ([3][4])。

以下、この数式エディタのシステムについて述べる。

2 本システムの概要

本システムでは、独自に開発した内部形式で表された数式を画面表示することにより、数式を視覚的に扱うことが可能である。

表示された数式は、簡単なペン操作により一般的な編集（切り取り、コピー、貼り付け）を行うことができる。この機能は、内部形式の構造により簡単に実現することができる。

また、各種数式フォーマットからの入出力機能（インタプリタ、コンバータ）を備えている。インタプリタは数式フォーマットによる数式を内部形式に変換し、コンバータは内部形式を数式フォーマットによる数式に変換する。それらは、汎用性のある構造になっており、文法を記述したリソースと簡単な仮想関数の書き換えによって、様々な数式フォーマットに対応することが可能である。各種数式フォーマットに対応したインタプリタやコンバータを必要に応じてサポートしていくことにより、数式フォーマット間の変換アプリケーションの開発コストの削減につながる（図 1）。

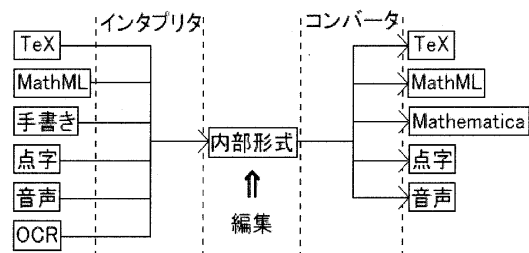


図 1: システムの概要

更に、ペンディスプレイなどを用いた手書きによる数式入力もサポートしている ([3][4])。これにより、TeX、MathML フォーマットによる数式の整形や編集が、手書き入力と簡単なペン操作で行えるといった実用的な

システムが実現した。

3 実現方法

この章では、内部形式、コンバータ、インタプリタの実現方法について述べる。手書きによる数式入力に関しては、[3]の論文を見られたい。

3.1 数式の内部形式

数式の内部形式は、数式の再帰構造を実現する独自のクラス構造で表された形式である。

このクラス設計には、オブジェクト指向ソフトウェアを設計する上で頻繁に再利用されるデザインパターンの1つである Composite パターンを適用させている([2])。Composite パターンは、オブジェクトを再帰的な木構造に組み立てることを可能にするデザインパターンであるため、数式の内部形式の実現に適していると考えられる。この木構造の実現により数式全体とその部分数式とを一樣に扱うことが可能になるので、編集、数式フォーマットへの変換などがより簡単に行えるようになる。

また、新しい数式の構造を簡単に追加できる設計になっているため、今回未対応である行列の構造を追加することも容易に行うことができる。

3.2 コンバータ

コンバータは、内部形式により表される数式の木構造から各種数式フォーマットによる数式文字列への変換を行う。

このクラス設計には、デザインパターンの1つである Builder パターンを適用させている([2])。Builder パターンは、変換の過程を各種数式フォーマットに依存せずに行うことができるため、汎用性のあるコンバータを実現するために採用した。これにより、新しい出力フォーマットへの対応が、各文字、記号の変換を記述したリソースといくつかの数式の構造を変換する仮想関数の書き換えによって実現することができる。

3.3 インタプリタ

インタプリタは、数式フォーマットによる数式の構文を解釈し、その結果生成された構文木から内部形式への変換を行う。

数式の文法を文脈自由文法で表し、その文脈自由文法を解釈するプッシュダウンオートマトンをインタプリタの実現に適用させている([1])。文脈自由文法は、一般的に言語モデルとして使われ、数式のような再帰的な構造を表すことができるので、構文を記述する文法として採用した。

インタプリタは、ある数式フォーマットの文法を記述

したリソースを読み込むことにより、その数式フォーマットを解釈するオートマトンに変化する。これは、文脈自由文法からプッシュダウンオートマトンを生成する方法に従って、数式構文解釈オートマトンが自動生成される仕組みになっているからである。そのため、新しい入力フォーマットへの対応は、そのフォーマットの文法を記述したリソースと各フォーマットに依存するいくつかの仮想関数の書き換えによって実現することができる。現時点では、TeX、MathML、Mathematica との数式のデータ通信を可能にするインタプリタ、コンバータが実現しており、目下、点字インタプリタ、コンバータも作成中である。

4 最後に

我々は、様々な数式フォーマットからの入出力機能を備えた数式エディタを開発し、コンピュータ上の数式を視覚的に編集する機能、各種数式フォーマット間の変換機能を実現した。エディタは、手書きによる数式入力をサポートし、ユーザーにやさしいインターフェイスが実現している。インタプリタ、コンバータは、必要な情報を記述したリソースと簡単な仮想関数の書き換えによって、様々な数式フォーマットへの対応を可能にする汎用性のあるクラス設計になっている。

また、このシステムは、ActiveX によるコンポーネント化により、多様なアプリケーションに容易に組み込むことが可能である。

参考文献

- [1] 都倉 信樹 著, “オートマトンと形式言語”, 昭晃堂 (1995)
- [2] Erich Gamma, Richard Helm, Ralph Johnson 著, 本位田 真一, 吉田 和樹 監訳, “オブジェクト指向における再利用のためのデザインパターン”, ソフトバンク株式会社 (1995)
- [3] H. Okamura, T. Kanahori, W. Cong, R. Fukuda, F. Tamari and M. Suzuki, “Handwriting Interface for Computer Algebra Systems”, *Proc. 4th. Asian Technology Conference in Mathematics*, December (1999) 291-300
- [4] 岡村 博文, 金堀 利洋, 叢 偉, 福田 亮治, 玉利 文和, 鈴木 昌和, “数式処理システムへの手書き数式入力インターフェイス”, 情報処理学会 第60回全国大会 (2000)