数理解析研究所講究録 1456 巻 2005 年 108-111

式変形支援システムの作成

元吉文男
(独) 産業技術総合研究所*
F.MOTOYOSHI
AIST

1 はじめに

人間が式の変形を行なうときには紙の上で行なうことが多いが、これを、ディスプレイ上で行なうための支援プログラムを作成しているので報告する。

いわゆる「数式」の変形には多くの数式処理システムがあるが、それにもかかわらず、紙を使用する場合も多い。これは、それらのシステムの多くが単に演算機能の提供に留まり、式変形の過程を紙の上で実行するような機能が不足しているためと考えられる。また、論理式などの数式以外の式の変形に関しては、定理証明や項書換えシステムなどがあるものの、それらは機械的に自動で実行できる部分を行なうもので、逐次的に人間が判断をしながら式を変形するようなものではない。

そこで、逐次的に人間が変形規則を指定しながら変形を進めるシステムを開発することにし、現在、プロトタイプといえるものを作成したので、以下にその詳細の述べる。

2 式変形支援システム

設計方針

前節で述べたように、式の変形を、人間が変形する対象や変形規則を指定して1ステップずつ実現することを基本に置く。操作はグラフィックインタフェースを活用し、自動的に決定できる事項に関しては、システムの方で処理を行なうこととする。

システムは図1に示すような画面をメインとし、画面上部に変形規則の一覧を表示し、下部に変形された式を過去の履歴も含めて表示する形態となっている。システムは Java で記述し、ユーザインタフェース部分は SWING と呼ばれるツールキットを使用し、Windows などの画面操作と同様の操作感になるようなプルダウンメニューも使用する。メニューバーの機能は現状では不十分ではあるが、システムの停止や途中結果のファイルへの保存、印刷などもここから実行させることにする予定である。

式変形は最下行の式への変形操作という形で実現する。マウスで部分式を指定するとその部分式に適用可能な規則の色が変化し、そのうちの1つを選択することによって変形操作が1ステップ進む。ここで重要な役割りを果たしている変形規則に関して以下に述べる。

*f.motoyoshi@aist.go.jp

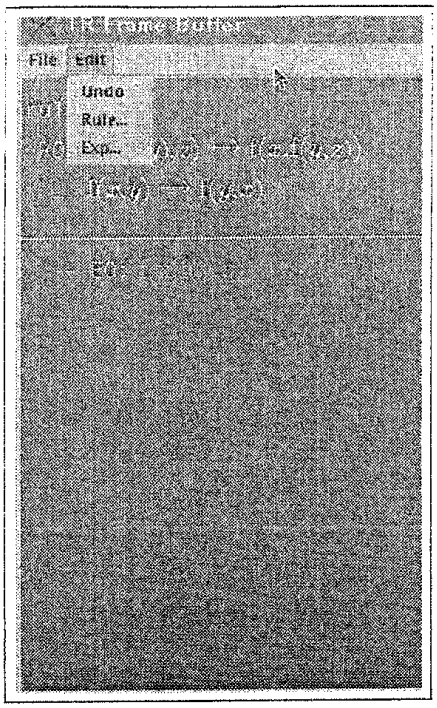

図 1: 式変換ツール (画面ハードコピー)

変形規則

変形は変形規則の集合を元に行ない、現在のところは、それ以外のシステム内部の規則はなにも無いとしており、画面上に表示されている規則が全てである。

変形規則は左辺と右辺からなっている。左辺は照合すべきパターンであり、右辺は照合した式を変形した後の式となっている。パターンにはパターン変数と呼ぶ変数を含むことが可能であり、パターン変数はどのような式とも照合する。ただし、同じパターン変数には同じ式しか照合しない。パターン変数以外の記号は同じもの同士しか照合しない。システム上では表示に際にイタリック体の文字はパターン変数を表し、立体の文字はそれ以外の定数記号を表わして両者を区別している。

式は全て、変数か $f(a, g(x))$ という関数適用の形で表現するものとし、等号などの関係式やラムダ式、限量された論理式なども $\exists(set(x, y), \wedge(p(x), q(y)))$ のように、みかけ上関数適用の形で表現することにする。

現在は未実装であるが、中置演算子やラムダ式などの自然な構文は、標準的なものはシステムで用意し、加えてユーザが構文を定義することも可能にする予定であり、また、そうすることが使い易いシステムになるためには欠かせないと考えられる。

フォント

表示に使用するフォントは TeX 用の computer modern である cmr10、cmmi10、cmex10、cmsy10 を truetype にしたものをダウンロードして使用している。Java には truetype フォントを直接読み込む機能が備わっており、Windows や Linux という OS に依存しない形でフォントを利用している。これらのフォントを Java を使用して表示した画面を図 2 に示す。この図のうち、cmr10、cmmi10 のボールドのフォントは Java の派生フォントの機能を使用して表示しており、cmb10,cmmib10 を (truetype フォントの構造をよ

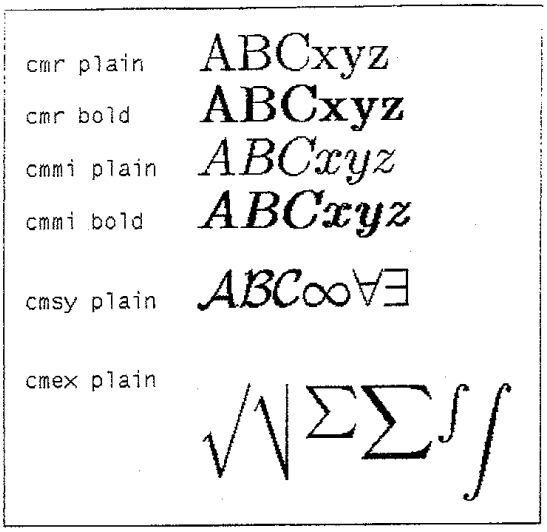

図 2: TeX 用 truetype フォント

く理解していないので、実際はそれぞれの内部にボールド体のフォントを持っているのかもしれないが、見かけ上は) 使用しているわけではない。とりあえず、この 4 種類のフォントを拡大率を変化させることでサイズを変えて数式の表示に使用している。TeX では添字の入れ子を表現するために、同じ字形でサイズが $1/1.2$、$1/1.2^2$ の計 3 つを使い分けて自然な数式を表示しており、ボールド体と合せて $18(=3\cdot 6)$ 個のフォントを使い分けることになる。

truetype による CM フォントも何種類か入手可能であるがここで使用しているのは BaKoMa フォントと呼ばれる無償のものである。これらのフォントは種類によって、元の metafont で 32 未満のコードに割り当てられている記号のコードと字形との対応が異なっており、互換性がないのが現状である。

ユーザインタフェース

使用感が良くなければ紙と鉛筆の替わりに使用する気にはなれないので、そのためにグラフィックインタフェースを多用している。

変形する部分式を指定するためには、それが変数であるときにはそのものを、関数適用であるときにはその関数名をマウスでクリックすることによって行なうことにした。四角い枠をマウスでドラッグし、その内部に含まれる最大の部分式を指定したものとする方法も考えられるが、枠と部分式との対応がユーザにとって明確でないことがあると考え、採用していない。

部分式が指定されると次にそれに変形操作を行なうことになるが、システムが自動的に適用可能な規則を検索して、可能なものに色を付けてユーザに表示する。その中からユーザが適用したい規則を選択することで変形を行ない、その下の行に新たに生成された式を表示する。その際に式に番号を付け、適用した規則の番号も表示するようにして、履歴が分かるようになっている。また古い式で選択された部分式に付いた色はそのまま残しておき、履歴の一部として表示されている (図 3)。

変換された式をユーザが見て、間違いと判断したとき、あるいは、別の変形を試したくなったときのために undo 機能を用意し、任意の時点まで戻れるようになっており、最初からやり直す必要がないようにしている。

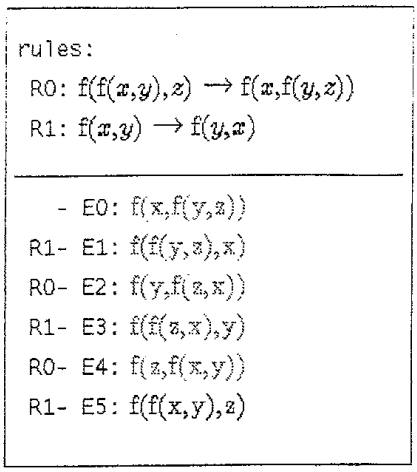

図 3: 変換操作 (白黒反転)

3 おわりに

以上紹介してきたように、現在のシステムは骨格だけができたという段階であり、不足している機能が以下のように考えられる。

- 変数名の生成
  変形規則の右辺にしか出現しない変数、あるいはラムダ変数などの束縛変数は任意の変数であるが、これが既に存在している他の変数と名前の衝突を起した場合には新たな名前にしなければならない。このときに $x001$ のような機械的に生成した名前ではなく、適切に名前を生成する。

- ファイル入出力
  変形規則や変形対象の式をファイルで作成してそこから入力する機能と変形の中途状態をファイルに保存しておき、後でその時点から処理を続ける。

- 構文規則のユーザ定義
  現在では関数適用形だけしか扱えないが、中置演算子やさらには表示法をユーザが定義することを可能にし、一般的に使用されている形で表示する。

- 標準的な変形規則
  交換則や結合則のようによく使用される規則はユーザがそのつど変形規則を定義することなく適用する。

- 数式処理機能との連携
  数式処理機能を 1 つの変形規則として考えることによって、数式の変形への利用を可能にする。

現在のシステムは以上の機能の追加が容易に行なえるように設計してあり、逐次追加していく予定である。また Java のツールキットである SWING 機能を十分に理解していると言い難いところがあるため、ユーザインタフェースにおいての機能の追加、変更も行なうことになると思われる。