

オンライン手書き数式認識システム「*METAH*」の実現

4H-5

村瀬敦史, 佐藤俊, 中川正樹  
(東京農工大学)

## 1. はじめに

現在我々は、文書作成における多くのフェーズを計算機を用いて行っている。確かに、文章の作成に関する環境は非常に快適なものとなっている。しかし、図や数式の入力では依然として多大な労力と時間を要するのが現状である。これは、その入力デバイスであるキーボードとマウスが、これらの入力に適していないためと考えられる。

ところで、最近入力デバイスとして表示一体型タブレットを用いたペン入力が注目されている。筆者は、このデバイスを文書創造時に用いることが有効と考え、特に、二次元的配置情報を有する数式の入力に着目し、その研究開発を試みた。

筆者の開発したシステムは、ユーザにより手書きで入力された数式のパターンをもとに、その構造を認識するものである。本稿では、主に本システムにおける数式の構造解析手法について述べる。

## 2. 手書き入力方式と数式の適合性

数式は、二次元的配置情報を有している。また、その入力のために、余計な操作をしたくない、極端な場合考えながらも入力したい、という要求がある。そこで、ペンを用いた手書き入力方式が、本当にこれらに適しているかということについて考えてみる。

数式は、記号の大きさや配置によりその構造を示しているが、手書きであれば、これらを自由に入力することができる。また、キーボードやマウスと異なり、座標間の移動を容易に高速に行うことが可能である。更に、従来のデバイスを用いる場合、入力時に様々なモード変換などの操作が必要となる。その結果、ユーザに思考の中断が生じ、数式を創造しながら入力することはほぼ不可能となる。これに対し、手書き入力は、人間が慣れ親しんだ方式で、特別な学習なしで入力することが可能である。加えて、誤認識などの修正を入力後に一括して行う UI を実現することにより、入力時にほとんど思考の中断が起らない。このことにより、計算機上で数式を解くことが可能となる。

逆に、手書き入力の欠点としては、レイアウトに必要な微調整が困難であったり、パターンまたは構造の認識が必要となる点が挙げられる。しかし、これらの点を自動化できれば理想的な数式入力(作成)が可能であると思われ、数式の入力方式という観点としては、手書き入力方式が適していると考えることができる。

## 3. オンライン手書き数式認識の処理構成

ここでは、手書きで入力されたパターンをもとに整形された数式を画面や紙面に出力するためには、どのような処理が必要となるかということについて述べる。

図1に示したように、オンライン手書き数式認識では、大きく分けて次に挙げる四つの処理が必要となる。

- (1) 記号認識処理
- (2) 記号分割処理
- (3) 構造化処理
- (4) フォーマット処理

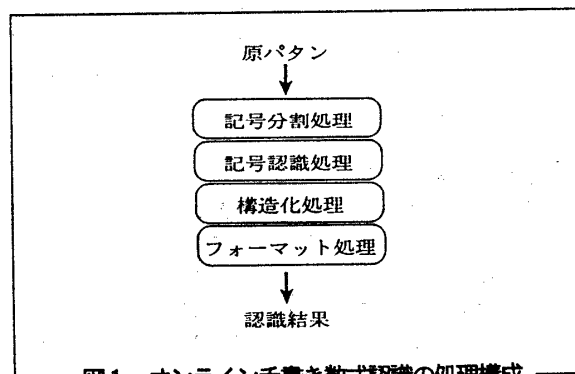


図1 オンライン手書き数式認識の処理構成

## (1) 記号認識処理

数式は、様々な記号(変数、数字を含む)から構成されている。そこで、まず手書きパターンをもとに、どんな記号が入力されたのかということを知る必要がある。このように、手書きのパターンをもとに入力された記号を符号化する処理が記号認識処理である。この処理には、我々の研究室で開発したオンライン手書き文字認識システム *JOLIS-2E* を用いる。

## (2) 記号分割処理

第2章で挙げたように、手書き入力方式であれば、記号の大きさや配置の制限なく入力することができる。しかし、このためシステムがあらかじめ一つの記号を入力するための枠などを用意し、そこへ入力するといった方法はとれない(手書き入力の利点を活かさないため)。したがって、入力されたパターンには、記号の区切りが在しない。そこで、記号認識処理を行う前に、入力されたパターンを各記号ごとに分ける処理が必要となる。これが記号分割処理である。

### (3) 構造化処理

入力された数式を整形して出力するためには、その構造を知る必要がある。上述したように、数式の構造は、記号の大きさや配置によって示されている。そこで、符号化された記号の幾何学的な情報をもとに、数式としての構造を認識する処理が必要となる。この処理を、構造化処理と呼ぶ。

### (4) フォーマット処理

流れの最後として、認識した構造をもとに画面や紙面に出力を行う処理が必要となる。数式には、その意味(構造)を示す表記規則が存在するので、これに沿って入力された数式を出力する。これがフォーマット処理である。

## 4. 数式の構造化処理の概要

ここでは、上述した構造化処理において、どのようにして入力された数式の構造を抽出しているかということについて述べる。

構造化処理は、次に挙げる二つの処理によって構成されている。

#### (1) 記号連結処理

#### (2) 項作成処理

これらは、自然言語の解析における形態素解析と構文解析という分類に似ている。

#### (1) 記号連結処理

数式の構造を示す幾何学的情報とは、ある記号の近傍に他の記号がどのような大きさ、位置関係で配置されているかということの組合せで表現されると考えられる。例えば、「x」の右上に小さく「2」と配置されてあれば、「x」の自乗を示すこととなる。そこで、数式の構造解析を行うために、まず記号どうしがどのような配置関係となっているかということ抽出する処理が必要となる。この処理は、それぞれの記号間に配置関係を示す連結(右上など)を付けるので、記号連結処理と呼ぶ。ただし、このような記号どうしの幾何学的な関係は、あらかじめ決められた配置にあるときのみ意味を持つ。

記号間に連結を付けるためには、まずあらかじめそれぞれの記号がどのような連結の種類を持ち得るかということ定義しておく必要がある(現在七種類の連結種を想定している)。そして、定義した連結種に対応する場所に条件を満たす他の記号が存在した場合、その記号を連結する。更に、この定義では、具体的に各連結種ごとにどの領域について探索すればよいのかということ定義する必要がある。この領域は、記号に対するある位置、ある大きさの矩形で表現する。つまり、この矩形内に他の記号が存在すれば連結を付けるわけである。

記号連結処理では、以上のような定義を参照しつつ、入力されたすべての記号間に意味のある連結を付ける。

#### (2) 項作成処理

数式は、項の入れ子構造となっていると考えられる。項とは、定数、変数または項とオペレータから構成される関数のことである。このように、項の構成要素として更に項が入るとい入れ子構造のため、入れ子をたどっていけば最終的には、必ず定数もしくは変数に行き着く。逆に、数式を成している定数、変数、オペレータを項という単位でまとめてい

ば、最終的にはただ一つの項ができあがる。したがって、一つの数式は、それ自体が一つの項であるということがいえる。そして、この構造は、オペレータをノードとし、変数または定数をターミナルとする木構造と等価であることがいえる。

項には、様々なものが存在している。例えば、「+」の項、「×」の項といったように、ほぼオペレータの数だけあるとあってよい。そして、それぞれの項においてその構成要素や配置の規則が異なる(項の構成)。また、ある項の構成要素を成す項はすべての項が許されているわけではなく、ある決まった項だけが構成要素となり得る。しかし、すべての項が決まった異なる項だけを構成要素としているわけではなく、同じような項をその構成要素とするグループが存在する(項の階層)。このような項の構成と階層を定義したものを数式文法と呼ぶ。

項作成処理では、(1)で作成した連結をたどりながら項を次々と作成する。各モジュールでは、一つの階層の項を作りその結果を返す。具体的には、数式文法にもとづき、その階層の項の構成要素をそろえる。もし、その構成要素として項が定義してあれば、その階層の項を作成するモジュールを呼び出す。このように、モジュールを呼び合うことにより、最終的に一つの大きな項ができあがる。これが、入力された一つの数式の構造を示す。

## 5. オンライン手書き数式認識システム「METAH」の実現

筆者は、タブレットとペンを入力デバイスとし、上述のような処理方法を用いて、オンライン手書き数式認識システム「METAH」を開発した。次にその実行画面の一例を示す。

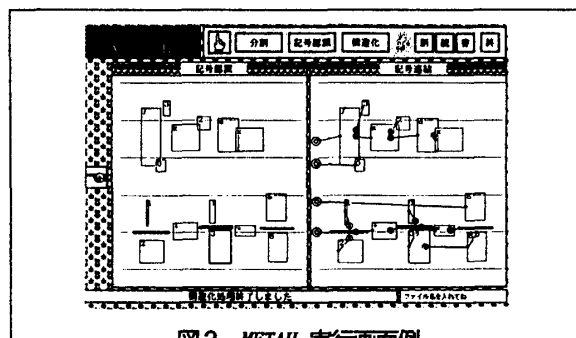


図2 METAH 実行画面例

## 6. おわりに

今回筆者が開発したシステム「METAH」は、まだプロトタイプ段階である。したがって、非常に多くのことを今後行う必要があるが、特に、

- ・充実した修正インタフェース
  - ・統合的な手書きシステムへの組み込み
- などの実現が今後の課題として挙げられる。

## 参考文献

- [1] 本間他：オンライン手書き文字認識システム JOLIS-2E の開発，情報処理学会第 40 回全国大会，4E-3 (1990)
- [2] 村瀬他：手書き入力による数式認識システム，情報処理学会ヒューマンインタフェース研究会報告，36-1 (1991)