

5 D-7

村瀬敦史, 福島英洋, 曾谷俊男, 中川正樹
(東京農工大学 工学部 電子情報工学科)

1. はじめに

計算機を用いて数式を作成するとき, 次に挙げる3つのタイプが考えられる.

- (1) 通常のワードプロセッサ
- (2) WYSIWYG 方式のエディタ
- (3) 文書整形システム

一貫性のある, 美しい出力を得るには, (3)を用いれば可能である. しかし, (3)は, 整形コマンドの挿入が必要なことや, 入力時にユーザが数式の構造を把握しづらいという問題点がある. 数式は, 単なる文字列ではなく, 文字の大きさ・位置により, その構造を表現している. それゆえ, テキスト入力方式では問題が生じる. 人間にとって, 数式の構造を一番自然に表現できる入力方式は手書きである. そこで, 手書き入力による, 数式認識システムを構築する.

2. ヒューマンインタフェースの設計

“数式を認識する”とは, 入力された文字列を数式の意味的な関係につなぎ合わせ, その構造と共に計算機内に格納するという処理である. 数式の構造認識処理のタイプには, 構造を組み立てる方法と認識のタイミングにより, それぞれ2種類ずつの方式が存在する. 前者には, 次の2つがある.

[構造抽出型]

入力された文字の大きさや配置から, 構造を抽出する.

[構造定義型]

構造を先に定義して, 後から各部分に内容を挿入する.

一方, 後者には,

(後処理型)

すべての入力が終了した後で, 一括して認識する.

(逐次処理型)

ある入力単位ごとに, 逐次認識する.

がある.

構造抽出型は, 何の制限もなくタブレット上に書かれた数式を, その文字の大きさや位置からすべてシステムがその構造を認識する方式である. つまり, この方式は, 通常我々がペンで紙面上に数式を書い

ている状態である. この方式であれば, ユーザがタブレットに数式を書いているときに思考の中断が生じることがない. しかし, 構造を誤認識する場合があるので, 訂正の手間を要する. 構造定義型は, 構造抽出型と逆で, ユーザが各部分の文字を書く前に, 先に構造を定義して, それから各部分の内容を書いてもらうという方式である. この方式であれば, 構造を誤認識することはなくなる. しかし, 入力中に思考の中断が生じる.

後処理型は, ユーザが区切りと考えるところで認識処理を行う方式である. これは, ユーザが計算機を使っていることを忘れて, あたかもペンで紙面上に数式を書いているように, つまり, 思考の中断を起させないことを狙いとする方式である. しかし, 思考の中断は生じないという利点がある反面, 認識する対象が大きいので, 構造認識に限らず文字認識における誤認識も増え, 訂正の手間が大きくなる. 逐次処理型は, システムが構造の区切りと考えるところで細かく認識処理を行う方式である. これは, 誤認識やリジェクトの訂正処理による思考の中断に関してはこだわらないが, 筆記とともに計算機入力が進んでいくことを望むユーザ向けである. 認識の対象が小さくなるので, 誤認識やリジェクトも少なくなると考えられる.

以上挙げたように, 方式としてそれぞれ2種類存在するので, 組合せとしては4種類考えられる. 次にそれぞれの組合せの性質を挙げる.

(1) 構造抽出型×後処理型

ユーザが区切りと考える所まで自由に書くことができるので, 思考の中断が存在しない. しかし, 認識後の訂正には多くの手間を要する.

(2) 構造抽出型×逐次処理型

逐次処理のため思考の中断が生じる. 認識する対象が小さいので, 前に誤認識した結果が後々影響することは少なくなるが, 文字の誤認識だけでなく, 構造の誤認識も起こりうる.

(3) 構造定義型×後処理型

構造の誤認識はないが, 構造定義のため思考は中断される. 文字の訂正を認識後に最初から行わなくては行けない.

(4) 構造定義型×逐次処理型

思考の中断は生じるものの、構造の訂正はなく、文字の訂正も前に入力したものに戻らなくてよい。つまり、訂正の手間が少なくて済む。数式に限らず文章にもいえることだが、文書を作成するときには、その状況として2種類のものが存在する。ひとつには、計算機を使っているという感覚をなるべく持ちたくないもの、もうひとつには、思考の中断にはこだわらず、訂正処理の手間をなるべく少なくし、着実に作業を進行させたいものである。前者を創造系、後者を複写系と呼ぶ。上で述べた組合せの性質を見ると、(2)、(3)は互いの利点をつぶしてしまっているの、意味のある組合せとはいえない。(1)は、訂正処理の手間は大きくなるが、思考の中断が存在しないので、創造系に適している。(4)は、思考の中断は生じるが、訂正処理の手間が少なく済むので、複写系に適している。

3. 数式用文字・記号の文字認識システムの実現

我々の研究室では、JOLIS-2E というオンライン手書き文字認識システムの開発を行っている。JOLIS-2E は、構造的解析手法にもとづく1文字認識方式のシステムである。このシステムでは、対象文字セットを定義することにより、字体表現辞書を半自動的に作成するツール群を有している。そこで、これらのツール群を用い、数式用文字・記号の辞書を作成した。つまり、現在の時点では、数式認識における文字認識の部分として、1文字認識方式による文字認識処理を実現している。これでは、およそ80%の認識率となっている。誤認識を起こしやすい文字としては、括弧類、“0”と“0”，などがある。

数式を認識するには、1文字認識方式でなく、枠なし文字認識方式を用いる必要がある。さらに、文字認識としては、数式用の文字・記号固有の特質を生かした認識処理にするために、現在の認識処理を改造するか、新しい認識処理を開発する必要がある。

4. 数式中の文字バタンの特質に適した認識処理

数式に用いられる文字バタンの持つ特質として、現在のところ次に挙げる3つが考えられる。

- (1) 低画数である
- (2) 対象文字ボタン数が少ない
- (3) 点か丸を含む文字ボタンが多い

構築する数式認識システムで対象とする文字ボタンは、およそ200文字で、最大4画である。JOLIS-2E は、2265文字を対象とし、最大画数は、23である。これからみても、(1)、(2)が言える。そして、点か丸を含む文字は、数式用の文字バタンの場合、約20%となる。それでは、文字ボタンを認識処理の対象としてみた場合に、それぞれの特質に対し、どのよう

な認識処理を施すべきかについて考察してみる。

(1) 低画数である

概して、今までの文字認識システムにおいて、高画数文字に比べ、低画数文字の認識率が低い。これは、各ストロークから抽出する情報量の不足と、1文字認識という処理方式が原因と思われる。高画数文字の場合、ストロークの位置関係などを用いて文字の識別を行えるが、低画数文字の場合は、各ストロークの持つ細かな特徴が文字の大きな識別要素となっている。そこで、低画数文字の認識処理としては、入力されたストロークから、今までの文字認識処理よりも細かな情報を抽出できるものにする必要がある。さらに、高画数文字(漢字)の場合は、文字単体に意味を持つものが多いのに比べ、低画数文字(平仮名、片仮名、アルファベット)では、それらの組合せにより、意味を持つものとなる場合が多い。つまり、低画数文字の認識には、前後の文字を用いた(意味的情報を用いた)処理が適していると考えられる。

(2) 対象文字ボタン数が少ない

文字ボタン数が少ないということは、辞書がコンパクトになるということである。そこで、個人用の辞書を作成したり、各文字カテゴリに対し、複数のボタンを登録することが容易になる。

(3) 点か丸を含む文字ボタンが多い

JOLIS-2E では、ストロークの構造を認識処理に用いている。しかし、点は構造自体にはあまり意味がなく、大きさによって他の文字との識別を行っている。そこで、認識に大きさの情報を用いた処理を用いることにより、点を識別することができると考えられる。次に、認識処理の手法として、ストロークの始終点座標を用いる場合が多いが、これは入力ストロークが丸の場合、誤認識となる可能性が高い。このような処理を用いる場合は、入力ストロークが丸の場合、回転不変な処理が必要と思われる。

5. おわりに

現在、JOLIS-2E をもとにした認識システムに4章の検討をふまえた改良を試みている段階である。数式認識における文字認識処理としては、上に挙げた他に、認識系の組合せや順番などによっても認識率の向上を図ることができると考えられる。数式認識システムとしては、いかに画面上に結果を出力するか、その他のアプリケーションとどのような関連性を持たせるかなどが今後の課題である。