ヒューマンインタフェース 68－9
（1996. 9. 12）

科学技術文書の点訳システムについて

鈴木昌和† 幸地司‡ 井上浩一† 謝 明†

† 九州大学大学院数理学研究科 〒812 福岡市東区箱崎 6-10-1
‡ (株) リコー情報通信研究所 〒227 横浜市港北区新横浜 3-2-3

Abstract

近年、コンピュータ技術の進歩により、点訳に必要な労力は飛躍的に軽減されてきた。英語、日本語のテキストファイルを点字に変換するソフトウェアも開発され、点訳作業の自動化も進んでいる。最近のOCR普及により、視覚障害者が印刷原稿を一人で点字に翻訳して読むことも可能になりつつある。

しかしながら、数式や化学式などの入った科学技術文書については状況はあまり改善されていない。これらをふまえ、我々は科学技術分野で標準となりつつあるLaTeX文書から点字への自動変換を実現した。また、数式を含む点字文書を墨字表示のまま編集するソフトの開発も行った。さらに、LaTeXをベースにして、数式文書の編集・処理を計算機上で容易に行うためのデータ形式としてMATH TEXTを設計した。

Keyword: 文書処理, 点字

Braille Transformation System of Science Document

Masakazu Suzuki† Tsukasa Kochi‡ Kouichi Inoue† Xie Ming†

† Graduate School of Mathematics,Kyushu University 6-10-1 Hakozaki,Higashi-ku,Fukuoka 812 japan
‡Information and Communication R & D Center, Ricoh Company,Ltd.
3-2-3 ShinYokohama,Kouhoku-ku,Yokohama 227 Japan

Abstract

Recently,the labor which is necessary for the braille transformation was reduced by the improvement of the computer technology.The software which changes the documents of English, Japanese into the braille is developed and it advances in the automation of the braille transformation work.With the recent OCR spread,the visually impaired person is becoming able to read a print manuscript by translating it into the braille alone. However, as for the science and techology document which a mathematical expression and a chemical formula and so on were stored in isn't too much improved. To make the situation better,We build a software translates LaTeXinto braille automatically.We also developed a softwar on which you can edit braille documents contain mathematical expression, seeing visible images. In addition, we designed MATH TEXT which was based on LaTeXdocument as the data form to do the editing and the processing of a mathematical expression document easily on the computer.

Keyword:document processing,braille transformation

1 はじめに

現在の点訳におけるコンピュータの利用形態としては、

1. 点字エディタによる点字文書の入力、編集、印刷[1]
2. テキストファイルから点字への自動変換（自動点訳）[2]

が一般的である。点字エディタは画面に点字イメージあるいは墨字で内容を表示しながら対話的に編集できるソフトで、入力後の修正が用意である、複数部の印刷が可能であるなどの理由で広く点訳の現場で用いられている。自動点訳に付いても、テキストファイルになった文書なら視覚障害者が他人の手を煩わせずに点字に変換できること、点字を知らない人でもレイアウトに関する規則に多少注意すれば簡単に点字文書がつくれることから広く普及している。その日本語変換精度や速度も非常に高い水準にあると言える。

しかしながら、それらはいずれも「普通の文章」を主たる対象としており、数式などに対する配慮は殆どみられない。数式の入った点字文書を作成するには複雑な点字数式規則を憶えている必要があるし、それを知っている人が点字エディタ上で点訳作業をする場合でも、画面上で数式イメージを見ながら編集、確認をすることはできない。文字が2次元的に配列される数式文書に対応するOCR技術は実用に至っていない。OCRで読みとっても必ず人手による大幅な書き直しが必要である。このように数式を含む文書の点訳環境が未整備である理由に、理系図書を必要とする盲学生は少数と考えられがちな点があることも事実である。しかし、教科書以外の理系図書の不足が盲学生の理系離れを助長している可能性も否めない。

数式の電子的表現方法には、大学や研究機関ではTEXがその標準としての位置を占めつつある。特にLATEXは各種参照機能、パッケージによる拡張性、可読性において優れており、広く普及している。また、LATEXは文書の論理構造に着目した言語構造となっているので、意味を保ったまま他の形式に翻訳しようとする場合に都合がよい。

そこで、我々はまず、LATEXからいくつかの中間過程を経て点字に翻訳するシステムを構築した。また、数式をそのイメージのまま画面で確認しながら点字編集が行えるソフトの開発も行った。

しかし、TEXは本来美しい印刷を得るためのシステムで、その記述の中には論理構造とはそれほど関係のない部分もおおい。また、TEXはバッチ処理用の言語で、エディタで編集する場合のデータ構造としては不向きである。そこでエディタによる編集にも耐えるものとしてMATH TEXTという形式を定義し、それをベースにした数式文書処理統合環境について考察する。

2 LATEX自動点訳と数式点訳システムについて

2.1 概要

我々の数式文書点訳システムは次のように区分することができる。

1. 数式対応点字エディタ+仮名訳前処理 (Quick Braille)
2. LATEX自動点訳システム (TEXBR[6])

LATEX文書を原稿として点字に翻訳しようとする試みはこれまでにも、佐藤らのMathBraille[4]がある。[4]は日本点字の問題点を指摘し、その改善を提言するなど積極的姿勢がみられるが、我々のシステムの設計思想とは以下の点で異なる。

1. 対象とする数式の範囲が初等的なものに限られる。我々は大学の参考図書や講義資料を点訳の対象とし、できる限り複雑な数式にも対応できるものを目指した。

[1]「こーたくん」、IBM-BE、ブレールスターII、フリーソフトではBASEが多く用いられる。
[2]EXTRA、「がってんだ」が代表的。他に80点等。

2. 数式以外の LaTeX コマンドを積極的にサポートしていない。LaTeX は自動章番号付けや箇条書きなど多くの便利な機能を持っており、既存の LaTeX 文書の殆どがそれらを利用している。既存分書の点訳を考えると、これらの機能を無視することはできない。タイトルや章の見出しなどは点字でも特定の書き方を要するので、我々は LaTeX のそれらコマンドを点字文書の整形に活用することにした。(2.3.3参照)

3. 現在、日本で普及している点字数学の記述法は墨字との対応が完全にはとれないという問題があり、[4] ではその改善を含めて提言を行っている[3]。しかし、我々のシステムでは、あえて現行のもの [1][3] を採用し、現在大学等で学んでいる盲学生に容易に利用できる出力を心がけた。

2.2 前処理および仮名訳処理 (Quick Braille)

我々のシステムでは漢字→仮名変換、分かち書き処理は行わない。そこで、LaTeX 文書はまず市販の点訳ソフト EXTRA の仮名訳機能を用いて、分かち書きされた仮名と英数字のみの文書に変換し、`*.hra` を出力する。TeX は明示的に指定しない限り改行を無視するので、その入力文書は日本語の意味とは関係なく改行されているのが普通である。EXTRA は入力文書の改行をそのまま出力に反映するので、例えば「現,れる」の ',' の位置で改行があると「げん, れる」と変換されて「あらわれる」にならない。また EXTRA に改行を無視させると本来の LaTeX 文書で意味のある改行も削除されてしまうため、EXTRA に渡す前に LaTeX で意味のない改行のみを削除する処理が必要である。さらに、墨字数学書では句読点の代わりに ',' や '.' が使われるが、点字では常に句読点を用いるのでその置換処理も行う。

加えて、現行の点字規則では数学と情報処理は異なった記号体系を用いることになっている。LaTeX ではプログラム等は`\verbatim` 環境で書くことが多いので、前処理でその部分だけを取り出し、情報処理点字に独自に変換した後、全ての処理が終わってから挿入、結合することとした。これらの処理は Quick Braille の中で行う。

2.3 LaTeX 自動点訳システム TEXBR

TEXBR は仮名分かち書きされた LaTeX 文書を点字に翻訳するシステムで、完全に独立したシステムである [6]。現在は C++で開発され、ms-dos 上で動作する。TEXBR は三つのサブシステムからなっており、以下の三つのプロセスに分けて動作する。入力は仮名分かち書きされた LaTeX 文書ファイル (`*.hra`)、最終出力は NABCC コードによる点字文書 (`*.brl`) である。中間ファイルとして 1 次中間ファイル (`*.stx`)、2 次中間ファイル (タグ付き点字コード、`*.tag`) を使用する。

2.3.1 TeX 点訳前処理

第一段階では、入力 LaTeX ファイルの空白処理やコメント削除、あるいはいくつかの点訳のための前処理がおこなわれる。またファイル中に、マクロ定義コマンドが存在すればそれを展開する。それらの結果は、一次中間ファイルとして出力される。

LaTeX は改行が二つ続くか強制改行命令が来るまで一つのパラグラフとして通常の改行は無視する。また、明示的に指定しない限り複数の半角スペースは一つに縮められる。さらに LaTeX 文書には点訳に関係のない空白記号なども含まれている。TEXBR はそれらのコマンドを削除し、LaTeX の仕様に合わせて強制改行、スペースの処理を行う。LaTeX ではマクロ定義のコマンドとして`\newcommand`, `\renewcommand`, `\def` があり、TEXBR ではこの三つを展開して一次中間ファイルの該当個所に挿入する。

[3] 現在その改訂作業が日本点字委員会でも進められている。

2.3.2 タグ付き点字コードへの変換

第二段階では一次中間ファイルを入力し、テキストモード部と数式モード部をそれぞれ別のアルゴリズムで点字に変換して、二次中間ファイルを出力する。

点字ではテキスト部と数式部で文字の役割が全く異なる。その為、変換はそれぞれ独立したアルゴリズムと辞書を用いて行うこととした。墨字と点字の対応は辞書ファイルから与え、容易に変更が可能である。現在のところ数式部は「点字数学記号解説 [1]」の規定を拡張して、大学専門書点訳のためにボランティアなどの間で広く用いられている方式 [3] を採用している。また点字では日本語中に英語が混在しているときは外国語引用符で囲むが、この処理もここで行う。

ただしここで出力された状態はまだ完全な点字書式にはなっていない。ここでは LaTeX 環境命令やセクション命令あるいは一部の空白命令などの文書構造に関わるコマンドに対しては、その環境名あるいはコマンド名を中カッコでくくり、マークアップして二次中間ファイルの対応する箇所に埋め込む。このマークアップされた点字コードをタグ付き点字コードと呼ぶ。また、この段階では点字の一行（通常３２マス）で折り返す処理は行っていない。

2.3.3 文書構造の解析と点字文書整形

最後の段階で LaTeX コマンドの環境命令で指定する文書構造を、タグ情報を手がかりに点字書式に従って文書整形をおこない最終ファイルを出力する。

点字文書は一般的に一頁が 22 × 32 と定められており、そのほかにも行替えやページ替え、見出しの書き方など、墨字にはみられない点字特有の数多くの細かい書式の指定が存在する。

そのため最終的に出力するファイルは、開発者側の責任において点字文書に定められている細かい指定に沿って整形されていなければならない。

ここでの処理は次のようである。

1. 数式符を挿入
   テキスト部から数式部に移行するとき、または数式の中でスペースが現れた時に、数式の前に数式符（5,6 の点）を挿入する。ただし、行列の中の空白や (x, y) のコンマの後、矢印の前後には例外的に挿入しない。

2. セクション自動番号付け
   本システムでは LaTeX のサポートする自動番号付けの内、`\section`, `\subsection`, 等をサポートしている。番号付けの他、章タイトルの適切な字下げも行う。

3. その他の環境命令タグ
   式番号付け（`\equation`, `\eqnarray`）や行列の処理、左右寄せ、箇条書きの環境命令タグを解釈して点字特有の書式に変換して出力する。

4. 自動ワードラップ機能 テキスト部 / 数式部
   点字は通常１行３２マスごとに改行して記述される。この際文書の意味や可読性を損なわない工夫が必要となる。テキスト部では分かち書きに従ってその切れ目で改行/改ページ処理を行えば良い。点字数式が２行以上にわたるとき、演算子の前で、それもできるだけ意味の切れ目で改行することが定められている。点字の数式には空白は存在しないので、本システムでは点字コードから直接数式の構成単位を解析して、適切な改行位置を判定しながらワードラップを行う。

   数式部の改行は等・不等号や演算子の前で行い、且つ、できるだけ対応するかっこは１行内に記述する。また、改行が少ない、すなわち１行が長い方が望ましい。これらの条件を満たすため、TEXBR

では演算子、記号、改行位置、括弧の深さ等に配点を行うことで、適切な改行位置を推測する工夫がなされている。

最終的な出力は、指定された行数、マス数でフォーマットされたNABCCコードによる点字文書である。

2.4 数式を含む点字文書編集システム (Quick Braille)

Quick Brailleはこれまでに述べてきた各処理を統合し、さらに画面上での点字編集を可能にし、6点入力もサポートする。既存の点字編集ソフトは点訳者向けに細やかな配慮[4]がなされているが、残念ながら点訳中の数式を墨字のイメージで画面に表示したり、その入力を支援する機能はない。Quick Brailleはひらがなや英字に加え、数学記号を墨字イメージで表示して編集処理するものである。それらの機能に加え、仮名訳前処理、TEXBRを用いたLaTeX点訳処理を統合し、科学技術文書点訳の統合環境とすべく開発した。他のソフトで点訳された数学書データを読み込んで、数式を含めた全体を墨字表示しながら校正作業を行うこともできる。

2.5 問題点

これまで述べてきた科学技術文書点訳システムであるが、現在のところ以下のような問題が確認されている。

- TEXBR
  1. 数式内ワードラップ処理に改善の余地がある。
  2. 出力する点字数学体系が固定で、Nemeth Code [5] やUBC [6]等他の体系に変更できない。
  3. マクロ展開処理で、引数の出現順序に正しく対応できない。
  4. \array環境で横幅が広い行列を扱えない。
  5. 一つの環境が大きくなると正しく処理できない。
  6. 処理速度が遅い。
- Quick Braille
  1. TEXBRはC++で記述されているのに対し、Quick BrailleはQuick basicであるため、他機種への移植が困難である。
  2. 仮名訳前処理時に情報処理点字を書き出すが、最後にそれを点訳された文書に埋め込む機能がない。
- システム全体
  1. ${}_nC_r$のように、点字では決められた書き方があるにも関わらずLaTeXで書き方が定まっていないものに関しては、本システムでも点訳できない。LaTeXのためのマクロパッケージで書き方を定義するなどの対策が必要である。
  2. 仮名訳前処理が独立していないため、Quick Brailleを起動しなければLaTeX→点字変換の一括処理ができない。

[4] 英語略字の展開表示、点図のサポート、自動ページ番号付け等
[5] アメリカで広く普及している数学点字体系 [5]
[6] Unified Braille Code(統一点字コード)。英語点字システムを拡張して理工系分野まで統一的にカバーする目的で米国を中心に検討されている点字体系

3. 現在は情報処理点字に対して仮名訳前処理で対応しているが、点訳処理本体 (TEXBR) で行う方が望ましい。

また、電算化されていない文書は人が入力しなければならない。その労力軽減には数式 OCR 技術が必須であり、緊急課題である。

3 システムとデータ形式の改善

ここでは、上に述べた問題点を解消するために現在取り組みつつある試みについて述べる。

3.1 TEXBR の再構築

TEXBR は当初、出力する点字の数学表記を日本で用いられる形式の他、Nemath コード、UBC にも変更可能とする予定であった。また、日本の点字数学体系も現在改訂作業が始まっている。しかし、LaTeX→点字変換の実現を急いだ結果、プログラムコード内部に日本点字数学表記に依存するコードが多く含まれることとなった。さらに、仮名訳前処理、情報処理点字、英語点字略字への対応、処理の高速化等が必要である。そのため、基本的アルゴリズムを受け継ぎながら、より柔軟で安定したものとすべく、現在、システムの再構築を検討している。

3.2 Quick Braille

Quick Braille は点字数学文書編集、既存の点字ファイル読み込み等数々の機能を持ちながら、Quick Basic で記述されているために移植性が損なわれている。現在、点字→LaTeX 変換部分を C で書き直す作業が行われている。今後は画面出力や 6 点入力などその性質上機種に依存する部分とそれ以外を独立させ、徐々に C や C++等の移植性の高い言語で再構成する。さらに、次節で述べる MATH TEXT の処理系として、科学技術文書を扱う統合環境とする方向で開発を進める。

3.3 MATH TEXT

Quick Braille は数式ワープロである Quick Note [7] とその基本部分のアルゴリズムを共有している。編集対象はいずれも「数学文書」であるが、入出力形式が「点字」、「墨字」と異なるにすぎない。数学文書というオブジェクトに対して行うべき操作は、単位ごとの挿入や削除等、両者ほぼ一致するものである。

数学文書のデータ形式としては LaTeX 文書が考えられるが、編集操作が容易であるためには以下のような問題がある。

- LaTeX はコンパイラでのバッチ処理を目的とした言語である。そのため、ある位置で現在着目している文字のモード (テキスト/数式)、フォントの種類、属性がすぐには判定できず、最悪の場合ファイル先頭からのスキャンが必要となる。
- 一つのコマンドの終わりを示す記号がない。
- 印刷イメージにのみ必要な情報と論理的に必要な情報が混在している。

そこで、Quick Note で用いられているデータ形式と LaTeX の文書形式をベースに、エディタ構築に適し、数式としての論理構造を表現できるものとして MATH TEXT と呼ぶ新たなデータ形式を設計した。

MATH TEXT は LaTeX の`\documentstyle` に対応する出力書式や文書のツリー構造等からなるヘッダ部分と、プロパティコードを含む文書部分からなり、

[7] 日本評論社「Quick-Note Ver.4」 1994. 現在はフリーソフトとして Nifty-Serve fgaldc に ver.6.08 がある。

- MATH TEXT→LaTeX→MATH TEXT の変換で不変。
- 使用するコードの範囲がテキストファイルに収まる。すなわち E-mail 等ネットワーク経由での送受が容易。

等の特徴を持つ。

3.3.1 MATH TEXT ヘッダ

ヘッダには識別コード、バージョン、印刷用紙の大きさ、マージン、ページ初期値、印刷スタイル等が入る。数学文書を実際に印字、画面表示するときに用いられる。さらに章や節のファイル中での位置をツリー構造で記録する。これは特定の部分のみの編集、挿入や削除、入れ替えを容易にする。また、点字文書では文章中の大きな構成単位 (部>章>節) になるほど見出しに大きな字下げを加えるが、このツリー構造により項目の深さが分かるので文書整形に役立つ。

3.3.2 プロパティコード

プロパティコードは一つ一つの文字に対する数式/テキストのモード属性、イタリックあるいはボールド属性の有無、フォントに関する情報を持ち、文書内の 1 文字あたり 1 バイトが割り当てられる。プロパティコードにより、その文字のモードや属性は前後をスキャンすることなく直ちに確定できる。

最初の 1 ビットは 0 で固定である。これはプロパティコード自体を ASCII コード 1 文字として記録するためである。

次の 2 ビットは文章/数式属性である。01 が文章属性、10 が数式属性を示す。理由は上に同じである。

次の 1 ビットはイタリク属性、または関数属性の ON/OFF である。ビットは立っていれば ON を示す。文書属性の場合はイタリク属性を、数式属性の場合は関数属性を示す。

最後の 3 ビットは、ローマン、カリグラフその他のフォント属性を示す。

3.3.3 数式構文の分類とその書式

MATH TEXT での数式表現は、LaTeX での書き方、コマンドを元に次のようにする。

- コマンド開始・終了記号: 各コマンドは{\で始まり}で終わるものとする。引数があればコマンド名の後、}の前に ';' で区切って置く。
- シンボル: ギリシャ文字 (小文字と大文字を含む)、括弧、関数名、演算記号 (2 項演算子、関係演算子とその他の記号を含む)、矢印、大型の特殊記号などは、引数のないコマンドとして扱う。すなわち、{*command name*} のようにする。
- 引数が一つの数式構文: 数式モードのアクセント、関数類 ($\lim_{x\to\infty}$ と $\max_{1<i<n}$ など) は {*command name*; 引数 }のように、引数が 1 として扱う。
- 引数が二つの数式構文: 級数記号類、分数、積分記号、根号は{*command name*; 引数 1; 引数 2}のように、引数が 2 として扱われる。
- 行列など: 行列、行列式、場合分けなどは特殊なタイプの数式として、各行を一つの引数とし、列の区切りは '&' とする。すなわち、{*matrix*; 引数 1 & 引数 2 & $\cdots$ & 引数 n; 引数 1 & 引数 2 & $\cdots$ & 引数 n; 引数 1 & 引数 2 & $\cdots$ & 引数 n }のように表現する。
- 添字: 添字、指数は LaTeX と同様に_{}, ^{}で表現する。

- 環境命令: その他の環境命令は、LaTeX に準ずる。

今後、LaTeX と MATH TEXT 間の相互変換、Quick Braille や TEXBR の MATH TEXT データ対応への書き換えが予定されている。

4 結び

我々は、これまでに LaTeX 文書から点字への変換、数式イメージを画面で確認しながらの点字編集などを実現した。しかし、現在システムで主に用いる文書データ形式は LaTeX である。LaTeX がこのような用途に向いていることは以前に述べたとおりであるが、LaTeX 自体が今も変化しつつあり、本来組版ソフトであることに起因する不適合性もあるので、これをそのまま標準とすることは無用な複雑さを生じかねない。今後、我々のシステムでは LaTeX↔MATH TEXT の可逆的な相互変換を維持しながら MATH TEXT を中核たるフォーマットとし、数式文書処理システムの統合化を目指す。

参考文献

[1] 日本点字委員会「点字数学記号解説」1989.8.8

[2] 全国点字図書館協議会「点訳のてびき」第 2 版 1995.4.30

[3] 井上浩一「数学専門書点訳のてびき」九州大学数理学研究科資料 1996.1

[4] 宮崎紀子 樋口美佳 佐藤浩史 原俊介 大武信之「数式自動点訳システム」信学技報 ET94-119 1995.1

[5] ICU 点訳サークル「NEMETH CODE による数学点訳のてびき」1986.11.1

[6] 幸地司「コンピュータによる LaTeX 文章自動点訳システム」九州大学大学院数理学研究科修士論文 1996.3