数理解析研究所講究録
第811巻 1992年 160-164

TeX-REDUCEインタフェイスの機能拡張

九州大学理学部基礎情報学研究施設
仁木直人 (Naoto NIKI)

1．はじめに

数式処理による結果はしばしば膨大な式になる。したがって、それを詳細に検討する際、あるいは論文中に取り込んで公表する際、計算機の外に出さずに、より自然で見易い形式に整える必要がある。

出版までを考慮した可搬性を重視すれば、T_EX (Knuth, 1984) 形式での出力が現状では最も適当と思われる。また、使用する機材によってはPostScript (Adobe Systems Inc., 1985) による記述の方が便利な場合もあるが、T_EX → PostScript 変換は既存の多くのシステムがサポートしているので、T_EX 形式をまず考えておくのが得策であろう。

現在、多くの数式処理システムにおいて T_EX 形式出力が可能となっているが、長大な結果が得られたとき、それをそのまま論文等に使えることはまずなく、かなりの煩わしい手作業を必要とする。特に、印刷幅に応じて複数行に分ける作業が問題である。

REDUCE (Hearn, 1991) には、ユーザ開発のソースライブラリ (reduce-netlib) として、TRI (T_EX REDUCE Interface; Antweiler, Strotmann and Winkelmann, 1991ab) と RLFI (REDUCE LaT_EX Formula Interface; Liska and Drska, 1991) があるが、次節に掲げるように、利用上の観点からだけでも多くの不満足な点を残している。

著者は、このため、実用性の高い TRI の改良版を作成し、論文作成等にすでに使用している。この改良版では、TRI の問題点が解消されるとともに、いくつかの機能が追加されているため、より広い範囲での応用が可能となっている。

2．既存ライブラリでの問題点

ふたつのプログラム・パッケージ TRI と RLFI の特徴を次頁の表1に示す。この表を見ると、両者の優れた所を兼ね備えれば、どうやら我々の目的に合うパッケージとなりそうである。いま両者を比較すると、最も手間のかかる「(必要なら変形した後)

表1　TRI と RLFI の特徴　（☆：他方に優る。△：場合による。）

	TRI (TeX REDUCE Interface)	RLFI (REDUCE LaTeX Formula Interface)
目的	REDUCE の出力形式をそのまま見易く印刷する。	(1) 数式を LaTeX 形式に簡単に書くため、REDUCE を変換に用いる。 (2) REDUCE による結果を文書中に使えるようにする。 ☆
複数行	印刷幅に合わせて数式を複数行に切り分ける機能がある。その際、必要ならば、分数式および平方根をそれぞれ (...) / (...) および $(...)^{1/2}$ の形に変形する。また、適当にインデントする機能も備えている。 ☆ また、手による修正も容易にできるような形式になっている。 ☆	通常 TeX で数式を記述するときのように、各行にうまく収まるよう、人手により行を切り分ける必要がある。
積	積の記号として常に・(\cdot) を置くことにより、変数の積と2文字以上の変数と区別している。 △	2文字以上の変数があると警告を出す。 △
変換	ギリシャ文字自動変換、大文字小文字変換の他、置き換えもできる。 ☆	ギリシャ文字自動変換の他、置き換えもできる。
添字	添字を扱う機能はない（内部に持っているが、ユーザが使えるようにはなっていない)。	operator の引数を、通常の形式以外にも、上下左右の添字として出力することができる。 ☆
行列	行列表示が可能である。複数行出力を指定して大きい行列を表示しようとすると、「行の切り分けがうまくいかなかったので溢れる」との警告が出る。 ☆	行列表示ができず、要素毎の表示に限る。
相性	plain TeX にいくつかのマクロ定義を追加して用いる。 AMS TeX とは相性がいいが、LaTeX のマクロ定義とぶつかるものがあるため、LaTeX および AMS LaTeX では事実上使えない。 △	LaTeX に依存した形式になっているため、plain TeX または AMS TeX で用いる際は、修正またはマクロ定義の追加を要す。 △ （AMS LaTeX との相性は未確認）
依存	PSL (Portable Standard Lisp) の特性に強く依存したプログラミングがされており、他の処理系では必ずしも期待通り動作しないことがある。	verbatim 環境の内容が空行のみであるとエラーになるような LaTeX 処理系では、出力の修正を要す場合がある。

数式を複数行に切り分ける機能」があり、また LaTeX (Lamport, 1986) でなく、より基本的な plain TeX 形式出力を目指している点からも、TRI の方が利用価値が高いように思われる。そこで、TRI を改良した新版を作成することとした。

3．TRI の改良版

この節では、TRI の問題点毎に、その改良版での解決方法および未解決課題について、やや詳しく述べる。

[目的] 我々の目標には「Reduce による計算結果を論文中などに取り込むこと」が含まれている。TRI の出力は「計算結果を見易く印刷すること」に力点が置かれているため、中点 (·) を用いた積の表示形式や、行をその位置で切る得失の評価基準である badness (Knuth (1984) 参照) の計算方法等の点において、論文中などにそのまま取り込むには不適当な面が多々見られる。また、LaTeX との相性を良くする必要があることは言うまでもない。これらの解決方法等については、各項目中で分けて述べる。

[積] 論文等の中では、中点 (·) を積の記号として頻用することは稀である。しかし一方、「見誤りを起こさず、計算結果を見易くする」観点からは、実用的な手段とも言える。そこで、この両方の表現を選択する switch を設けることとした。

```
ON TEXCDOT;    中点を積を表す記号として挿入する (default)
OFF TEXCDOT;   中点を挿入しない
```

中点挿入の有無に加えて、積の位置での行分割に関する badness 計算も調整している。当然、OFFの場合、ONの場合に比べてはるかに行分割が起き難くなる。ON側を default としたのは、既存版との互換性を考慮したためである。

[添字] Reduce には添字表現がないため、添字は operator の引数として与えることが多い。したがって、論文等の中で添字を用いる際には、煩わしい手直しが必要である。RLFI にある添字変換は、単に上左・下左・上右・下右の添字として引数の一部を並べるもので、それで間に合うものも多い。しかし、添字の間にコンマ (,) を入れたい、添字を各種の括弧でくくりたい、などの要求もまた案外多いものである。そこで、ある程度柔軟性のある変換と ready-made の変換の両方を用意する必要があ

りそうである。現在のところ、TEXLET 関数を拡張し、optional な第3の引数として変換形式を表現した単一化パターン (unification pattern) を与えることで、前者に対する要望のみを満たしている。より使い易い ready-made 変換については、どのような種類を用意し、どのような書式で指定すべきか、考慮中である。TEXLET 関数の使い方は

`TEXLET(symbol, item);` または `TEXLET(symbol, item, pattern);`

で、symbol には変換対象の operator 名、item には変換後の名称を表す文字列（例えば `"G"`, `"\alpha "` など）を指定し、pattern には単一化パターン（unification tag と特定の文字定数からなる LISP の list）を与える (この辺りも使い易さの面で改良する必要がある)。例えば、`S(2,3,4,5)` を $\sigma_{<23>}(4, 5)$ と表現したいときには、

```
TEXLET(S, "\sigma ", '( (F) !_!{ !\!< (Z) (Z) !\!> !}
                                !\!L!L!( (L !,) !\!R!R!) ) )
```

とする。なお、単一化パターンについては、Antweiler, Strotmann and Winkelmann (1991a) の5頁に説明がある。

[相性] TRI と LaTeX の両方で使われているコントロールシーケンス `\[`, `\]`, `\{`, `\}`, `\(`, `\)` の定義が異なるため、両者の共存は難しい。その対策として、まず `\[`, `\]` は、タイプセット時に読み捨てるものであるから、始めから出力しないこととした。残りのシーケンスは、それぞれ `\LL\{`, `\RR\}`, `\LL(`, `\RR)` のように変え、同時にタイプセット時の追加マクロ定義である reduce.tex の内容をそれらに合わせて変更を加えた。

[依存] TRI は、PSL (Griss, Benson and Maguire, 1982) の Cyber あるいは MicroVAX (Ultrix) 上のある implementation に依存する部分があり、他の Standard Lisp 系統の処理系でも正しく動かないことがある。以下に、問題点を列挙する。

(1) `EXPLODE2` のような Standard Lisp 規格 (Marti *et al*, 1978) では定義されていない関数あるいはマクロがいくつか使われている。

(2) コンパイラによっては、論理演算の結果を `T` または `NIL` で返すオブジェクトを生成する場合があるが、結果が真であるとき（インタプリタと同じく）確定した最後の値を返すことを前提としている。

(3) 大文字出力を大前提として、小文字出力が標準の処理系もあることに注意を

払っていない。

これらの問題点は、すでにプログラムの修正により解決している。また、 (3) に関しては、 TEXASSERTSET 関数の引数として大文字→小文字自動変換を指定する際に用いる LOWERCASE 文字セットに加え、 UPPERCASE 文字セットを追加定義した。

4．おわりに

本稿で述べてきた TRI の改良版は、行分割に関する badness 計算のチューニングが不十分であることと、添字の項で述べた変換指定の仕様が未だ固まらないため、公開を見合わせている。早期に公開できる形にし、 E-mail で配布する態勢を整える予定である。

参考文献

Adobe Systems Inc. (1985) *PostScript Language Reference Manual*, Addison-Wesley.

Antweiler, W., Strotmann, A. and Winkelmann V. (1991a) *Typesetting REDUCE Output with T_EX — A REDUCE-T_EX-Interface —*, Univ. Cologne, Germany (stored in *reduce-netlib*).

Antweiler, W., Strotmann, A. and Winkelmann V. (1991b) *A T_EX-REDUCE-Interface*, Univ. Cologne, Germany (stored in *reduce-netlib*).

Marti, J. B. , Hearn, A. C., Griss, M. L. and Griss, C. (1978) *Standard LISP Report*, Univ. Utah, Salt Lake City.

Griss, M. L., Benson, E. and Maguire, G.Q. (1982) PSL: A portable LISP system, *Proc. 1982 ACM Symp. LISP & Func. Programming.*

Hearn, A. C. (1991) *REDUCE User's Manual,* RAND Corp., Santa Monica.

Knuth, D. E. (1984) *The $T_EX$book*, Addison-Wesley.

Lamport, L. (1986) *A Document Preparation System LAT_EX — User's Guide and Reference Manual*, Addison-Wesley.