

ϵ -pT_EX 取扱説明書

北川 弘典 (H7K)*

build 90218 (2009/2/18)

1 はじめに

ϵ -pT_EX は、東京大学理学部数学科 3 年生対象の 2007 年度の授業「計算数学 II」^{*1}において北川が作成したプログラムであり、pT_EX 3.1.10 を基盤として、 ϵ -T_EX 2.2 相当の機能と、10 進 21 桁の浮動小数点演算とを追加したものである。

製作の動機や作業過程などについては、詳しくは [1] を参照して欲しいけれども、大雑把に言うと、動機は以下のように要約できる。

- pT_EX は、T_EX が持っている「レジスタ 1 種類につき 256 個まで」という制限をひきずっており、現状でも非常に多数のパッケージを読み込ませたりすると制限にぶち当たってしまう。
- 一方、 ϵ -T_EX 拡張ではこれが「レジスタ 1 種類につき 32768 個まで」と緩和されており、欧文で標準となっている pdfT_EX やその後継の LuaT_EX でも ϵ -T_EX の機能が取り込まれている。
- そうすると、pT_EX だけが制限をレジスタ制限を引きずっているのは世界から取り残されることになるのではないか。

2 ϵ -T_EX 拡張について

前に述べたように、 ϵ -T_EX は T_EX の拡張の一つである。 ϵ -T_EX のマニュアル [2] には、開発目的が以下のように述べられている。

The $\mathcal{N}\mathcal{T}\mathcal{S}$ project intends to develop an ‘New Typesetting System’ ($\mathcal{N}\mathcal{T}\mathcal{S}$) that will eventually replace today’s T_EX3. The $\mathcal{N}\mathcal{T}\mathcal{S}$ program will include many features missing in T_EX, but there will also exist a mode of operation that is 100% compatible with T_EX3. It will, necessarily, require quite some time to develop $\mathcal{N}\mathcal{T}\mathcal{S}$ to maturity and make it widely available.

* Web page: <http://sourceforge.jp/projects/eptex/wiki/>, e-mail: h_kitagawa2001@yahoo.co.jp

^{*1} <http://ks.ms.u-tokyo.ac.jp/>

Meanwhile ε -T_EX intends to fill the gap between T_EX3 and the future $\mathcal{N}_T\mathcal{S}$. It consists of a series of features extending the capabilities of T_EX3.

$\mathcal{N}_T\mathcal{S}$ がどうなったのか僕は知らない。しかし、少なくとも ε -T_EX 拡張自体は実用的な物であり、そのせいか \aleph (Aleph), pdfT_EX, X_YT_EX などの他の拡張にもマージされており、かなりの人が ε -T_EX 拡張を使うことができるようになっている。

ε -T_EX 拡張で追加される機能について、詳しくは [2] を参照して欲しい。しかしそうやって丸投げするのはよろしくなさそうな気もするので、ひとまず [1] 中の 4.2 節「 ε -T_EX の機能」を引用することにする（一部改変）:

ε -T_EX には Compatibility mode と Extended mode の 2 つが存在し、前者では ε -T_EX 特有の拡張は無効になるのでつまらない。後者がおもしろい。

拡張機能を使うにはファイル名を渡すときに * をつけるかコマンドラインオプションとして `-etex` スイッチをつければいいが、 ε -T_EX 拡張に関わる追加マクロは当然ながらそれだけでは駄目である。「plain マクロ for ε -T_EX」(`etex.fmt` というのが一番マシかな)では自動的に追加マクロである `etex.src` が呼ばれる。L^AT_EX 下ではちょうど `etex.src` に対応した `etex` パッケージを読み込む必要がある。

レジスタの増加

最初に述べたように、T_EX では 6 種類のレジスタが各 256 個ずつ利用できる。それぞれのレジスタには `\dimen75` などのように 0 ~ 255 の番号で指定できる他、予め別名の定義をしておけばそれによって指定することもできる。これらのいくつかは特殊な用途に用いられる（例えば `\count0` はページ番号などのように）ことになっているので、さらに user が使えるレジスタは減少する。

ε -T_EX では、追加のレジスタとして番号で言うと 256 ~ 32767 が使用できるようになった。上の pdf によると最初の 0 ~ 255 と違って若干の制限はあるようだが、それは些細な話である。追加された（各種類あたり） $32768 - 256 = 32512$ 個のレジスタは、メモリの効率を重視するため *sparse register* として、つまり、必要な時に始めてツリー構造の中で確保されるようになっている。

式が使用可能に

T_EX における数量の計算は充実しているとはいえない。例えば、

$$\backslash\dimen123 \leftarrow (\backslash\dimen42 + \backslash@tempdima)/2$$

という計算を元々の T_EX で書こうとすると、

$$\backslash\dimen123=\backslash\dimen42\advance\backslash\dimen123by\backslash@tempdima\backslash\dimen123=0.5\backslash@tempdima$$

のように書かないといけない。代入、加算代入、乗算代入、除算代入ぐらいしか演算が用意されていない状態になっている（上のコードのように、 $d_2 += 0.8d_1$ というような定数倍を冠することは平気）。

ε -T_EX では、そのレジスタの演算に、他のプログラミング言語で使われているような数式の表現が使えるようになった。上の PDF では実例として

`\ifdim \dimexpr (2pt-5pt)*\numexpr 3-3*13/5\relax + 34pt/2<\wd20`

が書かれている．これは，

$$32\text{pt} = (2\text{pt} - 5\text{pt})(3 - \text{div}(3 \cdot 13, 5)) + \frac{34\text{pt}}{2} < \text{\box20 の幅}$$

が真か偽かを判定していることになる．

`\middle primitive`

TeX に`\left`, `\right` という primitive があり，それを使えば括弧の大きさが自動調整されるのはよく知られている． ε -TeX では，さらに`\middle primitive` が追加された．

具体例を述べる．

$$\left\{ n + \frac{1}{2} \mid n \in \omega \right\} \left\{ n + \frac{1}{2} \mid n \in \omega \right\}$$

これは以下の source で出力したものである：

```
\def\set#1#2{\setbox0=\hbox{\$\displaystyle #1,#2$}%
\left\{\,\,\,\vphantom{\copy0}\#1 \,\,\right|\!\!\left.\,\,\,\vphantom{\copy0}\#2 \,\,\right\}}
\def\eset#1#2{\left\{\,\,\,\#1 \,\,\middle|\,\,\,\#2 \,\,\right\}}
\[[ \set{n+\frac{1}{2}}{n\in \omega} \eset{n+\frac{1}{2}}{n\in \omega} \]
```

両方とも集合の表記を行うコマンドである．TeX 流の`\set` では 2 つの`\left`, `\right` の組で実現させなければならず，そのために | の左側と右側に入る式の最大寸法を測定するという面倒な方法を使っている．その上，この定義では`\textstyle` 以下の数式（文章中の数式とか）ではそのまま使えず，それにも対応させようとする面倒になる．一方， ε -TeX 流の`\eset` では，何も考えずに`\left`, `\middle`, `\right` だけで実現できる．

TeX--XqT (TeX--XeT)

`left-to-right` と `right-to-left` を混植できるという機能であるらしい．ヘブライ語あたりの組版に使えるらしいが，よく知らない．ここでの `RtoL` は `LtoR` に組んだものを逆順にしているだけのような気がする．

とりあえず一目につきそうな拡張機能といったらこれぐらいだろうか．他にも `tracing` 機能や条件判断文の強化などあるが，そこら辺はパツとしないのでここで紹介するのは省略することにしよう．

ε -pTeX ではここに述べた代表的な機能を含め，ほとんどすべての機能を実装しているつもりである．ただ，TeX--XqT を和文で使うと約物の位置がずれたり空白がおかしかったりするけれども，その修正は大変に思えるので放置してある．

`\lastnodetype` と `\currentiflevel` の挙動については，[1] にもあるが，pTeX や浮動小数点演算実装のため，（それらで追加された部分に関しては） ε -TeX 本来の挙動では起こり得ない値をとることがある．

まず`\lastnodetype` については，以下のような値をとる．

−1: none (empty list) 6: adjust node 13: penalty node

0: char node	7: ligature node	14: unset node
1: hlist node	8: disc node	15: math mode nodes
2: vlist node	9: whatsit node	16: direction node
3: rule node	10: math node	17: displacement node
4: ins node	11: glue node	
5: mark node	12: kern node	

次に、`\currentiflevel` における条件判断文とそれを表す数字との対応は、以下のようになっている。

1: <code>\if</code>	8: <code>\ifmmode</code>	15: <code>\iftrue</code>	22: <code>\ifydir</code>
2: <code>\ifcat</code>	9: <code>\ifinner</code>	16: <code>\iffalse</code>	23: <code>\ifmdir</code>
3: <code>\ifnum</code>	10: <code>\ifvoid</code>	17: <code>\ifcase</code>	24: <code>\iftbox</code>
4: <code>\ifdim</code>	11: <code>\ifhbox</code>	18: <code>\ifdefined</code>	25: <code>\ifybox</code>
5: <code>\ifodd</code>	12: <code>\ifvbox</code>	19: <code>\ifcsname</code>	26: <code>\iffp</code>
6: <code>\ifvmode</code>	13: <code>\ifx</code>	20: <code>\iffontchar</code>	
7: <code>\ifhmode</code>	14: <code>\ifeof</code>	21: <code>\iftdir</code>	

3 浮動小数点演算

この機能は時間が余ったためにネタで入れたようなものである。非常に原始的なコマンドしか提供しておらず、僕自身ほとんど使わないためにもはや改善しようという意欲もあまりない（バグが見つければ、もちろん直す気ですが）。

ここでいう浮動小数点数とは、BASIC や FORTRAN で見られるような `-235.673578432E-534` のように、符号と 10 進小数からなる仮数部に、必要に応じて E or e で始まる指数部が続いたものである。符号は複数あってもよく、そのときは全部掛け算したものが最終的な符号となる。小数点は日米などで使われるピリオドも、欧州大陸で使われるコンマも許容される。

浮動小数点数に纏わるエラーは、オーバーフロー (Floating arithmetic overflow) と例外 (Floating arithmetic exception) の 2 種類のみである。エラーメッセージを出して一旦停止するが、無限大値や NaN 値をそれぞれ代入して続けることができる。

3.1 初期化

浮動小数点演算を行うときには、eTeX でいうところの extended mode で処理する必要がある、さらに計算に使用する一時領域や定数 π などを確保する必要がある。この一時領域確保は自動では行われない。

- `\fpinit` 一時領域その他を確保する。
- `\fpdest` 一時領域その他を解放する。

初期化を忘れて演算を行おうとしても、特に $\mathrm{T}_\mathrm{E}\mathrm{X}$ 側でエラーチェックは行わない。そのため、セグメンテーション違反のようなエラーを引き起こす危険性がある。解放忘れや二重確保は単に ($\mathrm{T}_\mathrm{E}\mathrm{X}$ 内の) メモリを無駄にするだけで、それ以外に問題はない。

3.2 代入, 型変換, 入出力

- `\real` $\langle real \rangle$ 浮動小数点数を表現する glue (以下, $\langle f-glue \rangle$ と称する) を返す.
- `\fpfrac` $\langle f-glue \rangle$ 引数の仮数部を返す.
- `\fpexpr` $\langle f-glue \rangle$ 引数の指数部を返す.
- `\fptoint` $\langle f-glue \rangle$ 引数を整数に変換したものを返す. 範囲内に収まらないときは, Number too big エラーを返す. なお, 引数が整数でないときは, 0 に近い方に丸められる.
- `\fptodim` $\langle f-glue \rangle$ 引数を dimension に, 1.0 がちょうど 1pt になるように変換したものを返す. 範囲内に収まらないときは, Dimension too large エラーを返す. なお, 引数が $1/65536$ (1sp に対応) でないときは, 同じように 0 に近い方に丸められる.

3.3 四則演算等

- `\fpadd` $\langle f-reg \rangle \langle real \rangle$ $\langle f-glue \rangle$ が格納された skip レジスタ (以下, $\langle f-reg \rangle$ と称する) と浮動小数点数を引数にとり, 2 つの浮動小数点数の和を計算して, $\langle f-glue \rangle$ に上書きする.
- `\fpsub` $\langle f-reg \rangle \langle real \rangle$ 同様に差を計算して, $\langle f-glue \rangle$ に上書きする.
- `\fpmul` $\langle f-reg \rangle \langle real \rangle$ 同様に積を計算して, $\langle f-glue \rangle$ に上書きする.
- `\fpdiv` $\langle f-reg \rangle \langle real \rangle$ 同様に商を計算して, $\langle f-glue \rangle$ に上書きする.
- `\fppow` $\langle f-reg \rangle \langle real \rangle$ 同様に累乗を計算して, $\langle f-glue \rangle$ に上書きする. $x^y = \exp(y \log x)$ で計算するため, 第 1 引数が負では当然ながらエラーが生じる.
- `\fppowi` $\langle f-reg \rangle \langle num \rangle$ `\fppow` と同様に累乗を計算するが, 第 2 引数, つまり指数部分は整数に限られる. その代わり, 第 1 引数は負でもかまわない.

3.4 単項演算

以下は単項演算で, $\langle f-reg \rangle$ を 1 つとり, 演算をし, 結果をその $\langle f-reg \rangle$ に上書きする. よって, 以下はコマンドの引数も省略し, 演算内容しか書かない. 引数の内容は便宜的に x で表す.

- `\fpneg` -1 倍を計算する.
- `\fpsqr` 平方根 \sqrt{x} を計算する.
- `\fpexp` 指数関数 $\exp x$ を計算する.
- `\fplog` 対数関数 $\log x$ を計算する. 底は $e = 2.71828 \dots$ である.

- `\fpabs` 絶対値を計算する .
- `\fpceil` 天井関数 $\lceil x \rceil$, つまり x を越えない最小の整数を計算する .
- `\fpfloor` 床関数 $\lfloor x \rfloor$, つまり x 以下の最大の整数を計算する .
- `\fpsin, \fpacos, \fpatan` それぞれ三角関数 $\sin x, \cos x, \tan x$ を計算する .
- `\fpsinh, \fpacosh, \fpatanh` それぞれ双曲線関数 $\sinh x, \cosh x, \tanh x$ を計算する .
- `\fpasin, \fpacos, \fpatan` それぞれ逆三角関数 $\arcsin x, \arccos x, \arctan x$ を計算する .
- `\fpasinh, \fpacosh, \fpatanh` それぞれ逆双曲線関数 $\operatorname{arcsinh} x, \operatorname{arccosh} x, \operatorname{arctanh} x$ を計算する .

逆三角関数では主値をとっている . すなわち , 計算結果は $\arcsin x, \arctan x \in [-\pi/2, \pi/2]$, $\arccos x \in [0, \pi]$ となる . また , 逆双曲線関数の計算結果は $\operatorname{arcsinh} x, \operatorname{arctanh} x \in \mathbf{R}$, $\operatorname{arccosh} x \in [0, \infty]$ の範囲に収まる .

これらの初等関数の計算は , 関係式を用いて引数 x を適度な大きさに変換し . そして Taylor 展開を用いて行っている .

参考文献

- [1] 北川 弘典 , 「計算数学 II 作業記録」, 2008 .
<https://sourceforge.jp/projects/epetex/document/resume/ja/1/resume.pdf> 他
- [2] The $\mathcal{N}\mathcal{T}\mathcal{S}$ Team. *The ε -T_EX manual*.
http://www.tug.org/texlive/Contents/live/texmf-dist/doc/etex/base/etex_man.pdf