ε-pTEX について

北川 弘典 *

version 160201

目次

1 はじめに

ε-pTEX 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る．もともとは pTEX 3.1.10 を基盤として，ε-TEX 2.2 相当の機能や 10 進 21 桁の浮動小数点演算を追加したものであったが，今では次の点が変わっている．

-
- TEX Live 2011 に取り込まれるにあたり，ε-TEX をベースにして，その上に pTEX 拡張やその他追加機能を載せる方針へと変更された．
- 浮動小数点演算の機能は 090927 版 (2009/9) から削除されている[*2]．

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．

- pTEX は，TEX 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-TEX 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 pdfTEX やその後継の LuaTEX，及び XETEX でも ε-TEX の機能が取り込まれている．

* http://osdn.jp/projects/eptex/wiki/, e-mail: h_kitagawa2001(at)yahoo.co.jp

*1 http://ks.ms.u-tokyo.ac.jp/.

*2 TEX ソース中で浮動小数点演算を行う手段としては，例えば LATEX3 の機能 (l3fp) や，xint パッケージバンドルがあるので，そちらを利用して欲しい．

- そうすると，pTEX 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-TEX 拡張について

前に述べたように，ε-TEX は TEX の拡張の一つである．ε-TEX のマニュアル [11] には，開発目的が以下のように述べられている.

> The $\mathcal{NTS}$ project intends to develop an 'New Typesetting System' ($\mathcal{NTS}$) that will eventually replace today's TEX3. The $\mathcal{NTS}$ program will include many features missing in TEX, but there will also exist a mode of operation that is 100% compatible with TEX3. It will, necessarily, require quite some time to develop $\mathcal{NTS}$ to maturity and make it widely available.
>
> Meanwhile ε-TEX intends to fill the gap between TEX3 and the future $\mathcal{NTS}$. It consists of a series of features extending the capabilities of TEX3.

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

ε-TEX 拡張で追加される機能について，詳しくは [11] を参照して欲しいが，[1] 中の 4.2 節「ε-TEX の機能」から一部改変して引用する.

> ε-TEX には Compatibility mode と Extended mode の 2 つが存在し，前者では ε-TEX 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-TEX 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-TEX」（`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．LATEX 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，TEX 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-TEX 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

TEX における数量の計算は充実しているとは言い難い．例えば，

$$\texttt{\textbackslash dimen123} \leftarrow (\texttt{\textbackslash dimen42} + \texttt{\textbackslash @tempdima})/2$$

という計算を元々の TEX で書こうとすると，

```
\dimen123=\dimen42\advance\dimen123by\@tempdima\dimen123=0.5\@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-TEX 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，

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

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．

$$\left\{ n + \frac{1}{2} \,\middle|\, n \in \omega \right\} \left\{ n + \frac{1}{2} \,\middle|\, n \in \omega \right\}$$

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\}}
\[ \set{n+\frac12}{n\in \omega} \eset{n+\frac12}{n\in \omega} \]
```

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（文中数式とか）ではそのまま使えず，それにも対応させようとすると面倒になる．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--XETEX** (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，そこら辺はパッ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--XETEX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，そこの修正は大変に思えるし，苦労して実装する意味があるのか疑問なので放置している．

pTEX 拡張では，TEX と比較して *dir_node* と disp_node という 2 種類の node が追加された．前者は，現在のリストの中に違う組方向の box を挿入する際に寸法を補正するために作られ，\hboxや\vboxのコンテナとなっている．また後者は，欧文文字のベースライン補正のために使われる．

ε-pTEX-110102 まではこれらの node も \lastnodetypeの値として出力させるようにしたが，両者ともにユーザーが意識する必要はないことから，ε-pTEX-110227 以降では *dir_node* と disp_node は \lastnodetypeの値として出力しないようにしている．

−1:	none (empty list)	5:	mark node	11:	glue node
0:	char node	6:	adjust node	12:	kern node
1:	hlist node	7:	ligature node	13:	penalty node
2:	vlist node	8:	disc node	14:	unset node
3:	rule node	9:	whatsit node	15:	math mode nodes
4:	ins node	10:	math node		

\currentiftype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．21–25 が，pTEX 拡張で追加された条件判断文に対応する．28 の \ifpdfprimitiveは pdfTEX 由来のプリミティブ（後述）である．

1:	\if	8:	\ifmmode	15:	\iftrue	**22:**	\ifydir
2:	\ifcat	9:	\ifinner	16:	\iffalse	**23:**	\ifmdir
3:	\ifnum	10:	\ifvoid	17:	\ifcase	**24:**	\iftbox
4:	\ifdim	11:	\ifhbox	18:	\ifdefined	**25:**	\ifybox
5:	\ifodd	12:	\ifvbox	19:	\ifcsname	**28:**	\ifpdfprimtive
6:	\ifvmode	13:	\ifx	20:	\iffontchar		
7:	\ifhmode	14:	\ifeof	**21:**	\iftdir		

3　Ω 由来の機能（旧名称：FAM256 パッチ）

ε-pTEX には，掲示板 TEX Q & A の山本氏の書き込み [2] に刺激されて作った，本節でに説明する Ω の一部機能を使えるようにするパッチが存在する．これは FAM256 パッチと呼ばれ，今までは「ε-pTEX 本体とは一応別扱いで，ε-pTEX の配布，及び W32TEX, TEX Live のバイナリでは標準で有効になっていただけ」という扱いであったが，それでは利用者が混乱するので**「FAM256 パッチは ε-pTEX-160201 以降からは切り離さない」**とここで宣言する．本ドキュメントの最後のページ[*3]にちょっとしたサンプルを載せてある．

本節で述べる追加機能は extendend mode でなくても有効になっている．ただし，後に説明する「レジスタが各種類 65536 個まで」は，extended mode の時に限り有効になる．

■数式フォント制限の緩和

Ω の大きな特徴としては，TEX 内部のデータ構造を倍の領域を用いるように改変し[*4]，TEX に従来から存在していた「256 個制限」を 2^{16} 個にまで緩和したことが挙げられる．同様に，Ω では（[2] にもあるように）数式フォントを同時に 256 個まで用いることができ，各フォントも 65536 文字まで許されるようになっている．

ε-pTEX では，中途半端だが，数式フォント 1 つあたりの使用可能文字数は 256 個のままで，同時に数式フォントを 256 個まで使えるようにしている．基本的には Ω と同様の方法を用いているが，内部でのデータ構造に違いがある（数字はすべて bit 幅）：

[*3] ただし，ソースファイルで言えば fam256d.tex（本文）と fam256p.tex（preamble 部）に対応する．

[*4] 詳しい話は texk/web2c/texmfmem.h 中の共用体 memoryword の定義を参照．大雑把に言うと，1 つの「メモリ要素」に 2 つの 32 bit 整数を同時に格納できるようになっている．

	category	family	char	math code	delimiter code
TEX82	3	4	8	$3+4+8=15$	$3+2\cdot(4+8)=27$
Ω	3	8	16	$3+8+16=27$	$(3+8+16,\ 8+16)=(27,24)$
ε-pTEX	3	8	8	$3+8+8=21$	$(3+8+8,\ 8+8)=(19,16)$

TEX に本来あったプリミティブは互換性維持のために同じ動作とする必要があるので，16 番から 255 番のフォントを利用する際には別のプリミティブが必要となる．（実装自体に Ω の流儀を使っているから）ここでは，Ω のプリミティブ名を流用することにした．すなわち，以下の primitive が追加されている[*5]．

- `\omathcode` ⟨8-*bit number*⟩=⟨27-*bit number*⟩
- `\omathcode` ⟨8-*bit number*⟩
- `\omathchar` ⟨27-*bit number*⟩
- `\omathaccent` ⟨27-*bit number*⟩
- `\omathchardef` ⟨*control sequence*⟩=⟨27-*bit number*⟩
- `\odelcode` ⟨8-*bit number*⟩=⟨27-*bit number*⟩ ⟨24-*bit number*⟩
- `\odelimiter` ⟨27-*bit number*⟩ ⟨24-*bit number*⟩
- `\oradical` ⟨27-*bit number*⟩ ⟨24-*bit number*⟩

ここで，27 bit とか 24 bit の自然数の意味については，上の表の Ω の行を参照して欲しい．上に書いた実装から，character code の指定に使われる 16 bit の数値で，実際に使われるのは下位 8 bit であり，上位 8 bit は無視される．なお，`\odelcode`⟨8-*bit number*⟩ として delimiter code を取得しようとしても，現時点のパッチでは，うまく動作しない[*6]．

LATEX において数式 fam を 16 個以上使うには，`\omathchar`などのプリミティブに対応したマクロを使う必要がある．最近の LATEX（少なくとも 2016-01-01 以降）では，**`\DeclareMathAlphabet`を用いて数式用アルファベットを使うだけなら**

```
\makeatletter
\mathchardef\e@mathgroup@top=256
\makeatother
```

をプリアンブルに記述すれば良い．だが, これだけでは `\DeclareMathSymbol`や `\DeclareMathDelimiter` が `\omathchar`や `\odelcode`を使用しないので不十分である．実験的と書かれてはいるが，山本氏による「最低限のパッケージ」[4] が手っ取り早いような気がする．

■無限のレベル

TEX では，glue の伸縮量に 3 つの無限大のレベルが存在した：`fil`, `fill`, `filll` であり，`l` が多いほど無限大のオーダーが高い．Ω では，「inter-letter spacing のために」`fi` という，有限と `fil` の中間にあたる無限大のレベルが付け加えられ，`\hfi`, `\vfi`という 2 つの primitive も追加された．そこで，この無限大レベル `fi` も採用することにした．

実装方法は，大まかには Ω で `fi` の実装を行っている change file `omfi.ch` の通りであるのだが，これに pTEX や ε-TEX に伴う少々の修正を行っている．

[*5] Ω では ⟨8-*bit number*⟩ のところが ⟨16-*bit number*⟩ になっている．
[*6] 51 bit 自然数を返さないといけないですからねえ．やる気があれば検討してみます．

- プリミティブ \pagefistretchを新たに定義している．
- \gluestretchorder, \glueshrinkorderの動作を ε-TEX のそれと合わせた．具体的には，ある適当な glue \someglueの伸び量を $\langle stretch \rangle$ とおくとき，

$$\texttt{\textbackslash gluestretchorder\textbackslash someglue} = \begin{cases} 0 & \langle stretch \rangle \text{ が高々 } \texttt{fi} \text{ レベルの量} \\ 1 & \langle stretch \rangle \text{ がちょうど } \texttt{fil} \text{ レベルの無限量} \\ 2 & \langle stretch \rangle \text{ がちょうど } \texttt{fill} \text{ レベルの無限量} \\ 3 & \langle stretch \rangle \text{ がちょうど } \texttt{filll} \text{ レベルの無限量} \end{cases}$$

となっている．内部では fi レベルが 1，fil レベルが 2，……として処理している．

■レジスタについて

Ω では（前にも書いたが）データ構造の変更が行われ，それによってレジスタが各種類あたり 65536 個使えるようになっている．

一方，ε-TEX では，256 番以降のレジスタを専用の sparse tree に格納することにより，32767 番までのレジスタの使用を可能にしていた．このツリー構造を分析してみると，65536 個までレジスタを拡張するのはさほど難しくないことのように思われた．具体的には，ツリーの階層を 1 つ増やしてみた（だから，おそらく各種類あたり $16 \cdot 32768 = 524288$ 個まで使えるとは思うが，これはきりが悪い）．そこで，ε-pTEX では ε-TEX 流の方法を用いながらも，レジスタをさらに 65536 個まで増やしている．

4 pdfTEX 由来の機能

開発中の LATEX3 では，ε-TEX 拡張の他に，pdfTEX で導入された \pdfstrcmp（又はその同等品）が必要となっており，もはや純粋な ε-TEX ですら LATEX3 を利用することはできない状況である ([5, 6, 7])．その他にも，pdfTEX 由来のいくつかのプリミティブ ([13]) の実装が日本の TEX ユーザからあり，ほとんど pdfTEX における実装をそのまま真似する形で実装している．

現在の ε-pTEX で利用できる pdfTEX 由来のプリミティブの一覧を以下に示す．これらは extended mode でないと利用できない．

\pdfstrcmp ⟨*general text*⟩⟨*general text*⟩

2 つの引数を文字列化したものを先頭バイトから比較し，結果を −1（第 1 引数の方が先），0（等しい），1（第 2 引数の方が先）として返す．比較する文字列中に和文文字がある場合には，（ε-pTEX の内部漢字コードにかかわらず）UTF-8 で符号化して比較する．

\pdfpagewidth, \pdfpageheight

ページの「幅」「高さ」を表す内部長さであるが，**ここで言う「幅」は「字送り方向」のことではなく，物理的な意味である**．

\pdflastxpos, \pdflastypos

\pdfsaveposが置かれた場所の，dvi における出力位置を返す内部整数（読み取り専用）．原点は紙の（物理的な意味の）左下隅であり，y 軸は（物理的な）上方向に向かって増加する．

pTEX では横組・縦組と組方向が複数あるので，\pdflastxpos, \pdflastyposの値の座標系を「物理的な」向きとすべきか，それとも「組方向に応じた」向きとすべきかは悩みどころである．110227 版以降，現在までの版では上記のように物理的な向きとしている．

\pdfcreationdate, \pdffilemoddate, \pdffilesize

\pdffiledump [**offset** ⟨*offset*⟩]**length** ⟨*length*⟩ ⟨*filename*⟩　⟨*filename*⟩ で与えられたファイル名の ⟨*offset*⟩ バイト目（先頭は 0）から ⟨*length*⟩ バイトを読み込み，16 進表記（大文字）したものに展開される．

本プリミティブは Heiko Oberdiek 氏による bmpsize パッケージを ε-pTEX でも使うために角藤さんが実装したものである (2014.5.6).

\pdfshellescape

\write18による shell-escape が利用可能になっているかを示す内部整数（読み取り専用）．0 ならば不許可，1 ならば許可，2 ならば restricted shell-escape[*7]である．

本制御綴は TEX ユーザの集い 2014 でリクエストを受けて実装したものである ([9]).

\pdfmdfivesum

\pdfpritimive, \ifpdfprimitive

\pdfprimitiveは次に続く制御綴がプリミティブと同じ名称であった場合に，プリミティブ本来の意味で実行させるものである．例えば

```
\pdfprimitive\par
```

は，\parが再定義されていようが，本来の \parの意味（段落終了）となる．また，\ifpdfprimitiveは，次に続く制御綴が同名のプリミティブの意味を持っていれば真，そうでなければ偽となる条件判断文である．

これらのプリミティブは 2015/7/15 版の expl3 パッケージで使われた ([10]) ことを受けて実装されたものだが，現在ではこれらのプリミティブは使われていない．

5　\lastnodecharプリミティブ

……

参考文献

[1] 北川 弘典,「計算数学 II　作業記録」，2008.
https://osdn.jp/projects/eptex/document/resume/ja/1/resume.pdf 他

[2] 山本 和義,「数式 fam の制限と luatex」，掲示板「TEX Q & A」52744 番書き込み，2009.2.12,
http://oku.edu.mie-u.ac.jp/~okumura/texfaq/qa/52744.html

[3] 山本 和義,「Re: 数式 fam の制限と luatex」，掲示板「TEX Q & A」52767 番書き込み，2009.2.16,
http://oku.edu.mie-u.ac.jp/~okumura/texfaq/qa/52767.html

[4] 山本 和義,「数式 fam 拡張マクロ for e-pTeX 等」，掲示板「TEX Q & A」52799 番書き込み，2009.2.21，http://oku.edu.mie-u.ac.jp/~okumura/texfaq/qa/52799.html

[5] 河原,「パッケージとディストリビューションについて」，掲示板「TEX Q & A」55464 番書き込み，2010.12.16，http://oku.edu.mie-u.ac.jp/~okumura/texfaq/qa/55464.html

[6] 角藤 亮,「Re: パッケージとディストリビューションについて」，掲示板「TEX Q & A」55478 番書き込み，2010.12.19，http://oku.edu.mie-u.ac.jp/~okumura/texfaq/qa/55478.html

[*7] あらかじめ「安全」と認められたプログラム（texmf.cnf 中で指定する）のみ実行を許可する仕組み.

[7] zrbabbler,「LaTeX3 と expl3 パッケージ」, ブログ「マクロツイーター」内, 2010.12.22, http://d.hatena.ne.jp/zrbabbler/20101222/1293050561

[8] 角藤 亮,「Re: e-pTeX 101231」, 掲示板「TEX Q & A」55528 番書き込み, 2011.1.1, http://oku.edu.mie-u.ac.jp/~okumura/texfaq/qa/55528.html

[9] Dora TeX,「Re: \pdfshellescape, \lastnodechar の実装」, TEX Forum, 2014.11.19, http://oku.edu.mie-u.ac.jp/tex/mod/forum/discuss.php?d=1435#p8053

[10] tat tsan,「[expl3 / e(u)ptex] 2015/07/15 版 expl3 パッケージが、(u)platex で通らない」, TEX Forum, 2015.7. 26, http://oku.edu.mie-u.ac.jp/tex/mod/forum/discuss.php?d=1632

[11] The $\mathcal{NTS}$ Team. *The ε-TEX manual* (v2.0).
$TEXMFDIST/doc/etex/base/etex_man.pdf

[12] J. Plaice, Y. Haralambous. *Draft documentation for the Ω system*, 1999.
$TEXMFDIST/doc/omega/base/doc-1.8.tex

[13] Hàn Thế Thành et al. *The pdfTEX user manual*, 2015.
$TEXMFDIST/doc/pdftex/manual/pdftex-a.pdf

Test source for FAM256 patch

本ソースは山本和義氏による「数式 fam の制限と luatex」(qa:52744) 中のコードをベースにしたものである.

■ **More than 16 math font families.**

A**BCDEFGHIJKL fam = 19**

AaAbAcAdAeAfAgAhAiAjAkAlAmAnAoAp fam35

AaAbAcAdAeAfAgAhAiAjAkAlAmAnAoAp fam51

AaAbAcAdAeAfAgAhAiAjAkAlAmAnAo**Ap fam67**

Aa fam68 Ab fam69 Ac fam70 Ad fam71 roman

■ **\omathchar etc.**

\mathchar"7F25 : %, \omathchar"7420125 : %

meaning of \langle: macro:->\delimiter "426830A ,

meaning of \lx: macro:->\odelimiter "4450068"030001

\lx : h, \bigl\lx :), \Bigl\lx : $\Big)$

\the\mathcode‘\f: 7166, \the\omathcode‘\f: 7010066（どちらも 16 進に変換した）

t $\rangle\big\rangle\big)\big)\langle\rangle$

$\surd$, p $\sqrt{a}$, p■ $\sqrt{\int_V f\,d\mu}$, $\sqrt{\int_V f\,d\mu}$

\odelcode primitive による **delimiter code の取得**はうまく動かない：

\the\delcode‘\|: 2532108, \the\odelcode‘\|: −1（どちらも 10 進）

■ **Infinite level “fi”**

■(fi)■(fil)■(fill)■		(filll)		■
■(fi)■(fil)■		(fill)		■
■(fi)■		(fil)		■
■	(fi) ■	(fi)	(fi)	■

■ **65536 registers**

fuga! a 漢字仮名asdf TEX ほげほげTEX

589