

Руководство по ε -TeX

Версия 2, февраль 1998

Команда $\mathcal{N}\mathcal{T}\mathcal{S}$

Peter Breitenlohner, Max-Planck-Institut für Physik, München

Подготовка этого отчёта частично поддержана DANTE,
Deutschsprachige Anwendervereinigung TeX e.V.
‘TeX’ торговая марка Американского Математического Общества.

Перевёл на русский к. т. н. Жуков И. Б.
Ленинград, 2013 г.

(Предварительный вариант перевода)

Последняя версия перевода размещена на сайте:

http://ibzh.eko3.ru/tex/etex_man_ru.htm

О неточностях перевода сообщайте по e-mail: ibzh@yandex.ru.

1 Введение

Проект $\mathcal{N}\mathcal{T}\mathcal{S}$ нацелен на разработку ‘Новой Системы Вёрстки’ ($\mathcal{N}\mathcal{T}\mathcal{S}$), которая в конечном счёте заменит сегодняшний TeX3. Программа $\mathcal{N}\mathcal{T}\mathcal{S}$ включит многие возможности, упущенные в TeX, но будет также режим 100% совместимости с TeX3. Конечно, потребуется определённое время, чтобы довести $\mathcal{N}\mathcal{T}\mathcal{S}$ до зрелости и сделать общедоступной.

Тем временем ε -TeX будет заполнять разрыв между TeX3 и будущей $\mathcal{N}\mathcal{T}\mathcal{S}$. Он состоит из нескольких расширений возможностей TeX3.¹

Поскольку совместимость между ε -TeX и TeX3 — главная забота, ε -TeX имеет два режима работы:

(1) В режиме совместимости с TeX он полностью соответствует TeX и поэтому ни расширяет возможности, ни добавляет новые примитивы. Это, в частности, означает, что ε -TeX проходит TRIP тест [1] без любых ограничений. Однако, есть небольшие изменения, допустимые в любом воплощении TeX.

(2) В расширенном режиме есть дополнительные примитивные команды, и доступны расширенные возможности ε -TeX.

Мы попытались сделать ε -TeX совместимым с TeX, насколько это возможно, даже в расширенном режиме. В немногих случаях есть тонкие различия, которые подробно описаны ниже. Поэтому возможности ε -TeX, до-

¹Программа TeX3; сейчас нет намерений расширять программное обеспечение, связанное с TeX.

ступные в расширенном режиме поделены на два разряда:

(1) Большинство из них семантически ни на что не влияют, пока пока дополнительные примитивы не будут исполняться; эти «расширения» всегда разрешены.

(2) Остальные необязательные ε -TeX свойства («улучшения») могут по отдельности разрешаться и запрещаться; исходно они все запрещены. Для каждого улучшения есть переменная состояния `\...state`; улучшение разрешается или запрещается присвоением положительного или отрицательного значения, соответственно, этой переменной.

Для ε -TeX версий 1 и 2 есть только одно улучшение: смешанное направление вёрстки (TeX--XeL) с переменной состояния `\TeXXeTstate`.

Версия 1.1 ε -TeX выпущена в ноябре 1996, версия 2.0 в феврале 1998. Ожидается около одной версии ε -TeX в год, где каждая последующая версия будет добавлять новые возможности. Было бы желательно, если бы эти версии ε -TeX включались во многие существующие разработки TeX3 без больших задержек.

С каждой версией ε -TeX будет поставляться тест e-TRIP [2] для проверки, что данная разработка заслуживает имени ε -TeX таким же образом, как тест TRIP [1] помогает проверить, что разработка заслуживает имени TeX.

2 Создание ε -TeX

2.1 Создание программы ε -TeX

Воплощение TeX'a состоит из WEB-файла изменений `tex.ch`, содержащего все системно-зависимые изменения для данной системы. Программа TANGLE системы WEB применяет этот файл изменений к системно-независимому файлу `tex.web`, определяющему программу TeX, чтобы произвести паскалевский файл TeX'a для этой системы [3]. Подобным образом, реализация ε -TeX состоит из системно-зависимого файла `etex.sys`, применяемого к системно-независимому файлу `e-tex.web`, определяющему программу ε -TeX. Т. к. ε -TeX отличается от TeX относительно небольшой частью своего кода, `e-tex.web` физически не существует, как файл; вместо этого, он определён в понятиях системно-независимого файла изменений `e-tex.ch`, применяемого к `tex.web`. Подобным образом, можно определить системно-зависимый файл изменений `etex.sys` для частной системы в понятиях его отличий от соответствующего файла `tex.ch` [4].

2.2 Создание форматных файлов для ε -TeX

Когда программа TeX (версия INITEX или VIRTEX) начинает работу, она смотрит первую непустую входную строку из файла или с терминала (с подсказкой `**`): Первый непробельный знак этой входной строки может быть `&`, за которым непосредственно следует имя загружаемого формата; иначе

VIRTEX использует формат по умолчанию, а INITEX начинает работу без загрузки форматного файла.

Для eINITEX (INITEX версия ε -TEX) есть дополнительная возможность: если первый небельный входной знак суть `*`, программа начинает работу в расширенном режиме, не загружая форматный файл. Если первый небельный знак ни `&`, ни `*`, то eINITEX начинает работу без загрузки форматного файла в режиме совместимости. Когда форматный файл загружается eINITEX'ом или eVIRTEX'ом режим (совместимости или расширенный) принимается из форматного файла.

Советуем использовать входной файл `etex.src` вместо `plain.tex` при создании формата ε -TEX в расширенном режиме. Этот файл сперва читает `plain.tex` (не читая `hyphen.tex`) и затем добавляет макро определения, поддерживающие дополнительные возможности ε -TEX.

3 ε -TEX Расширения

3.1 Совместимость и расширенный режим

Когда ε -TEX вошёл в режим совместимости, он работает как любая другая система TEX. Отсутствуют все дополнительные команды ε -TEX'а, и получить доступ к любым улучшениям или расширениям нельзя. Однако, eINITEX имеет исходно возможность выбирать между режимом совместимости и расширенным режимом.

Остаток этого документа посвящён подробному техническому описанию всех сторон, где ε -TEX (в расширенном режиме) ведёт себя отлично от TEX. Подразумевается, что читатель хорошо знаком с *The TEXbook*², подробно описывающим TEX.

Все расширения и улучшения ε -TEX'а доступны в расширенном режиме и включаются или выполнением некоторого нового примитива (примитивной команды), или присвоением ненулевого значения некоторому новому параметру или переменной состояния. Т. к. все эти новые переменные исходно равны нулю³, ε -TEX работает как TEX, пока не встретит новую управляющую последовательность ε -TEX'а, за следующими исключениями, которые тем не менее не должны влиять на вёрстку не содержащих ошибок документов TEX (производимы с помощью не содержащих ошибок форматных файлов):

- (1) Когда `\tracingcommands` имеет значение 3 или выше, или когда `\tracinglostchars` имеет значение 2 или выше, ε -TEX будет отображать дополнительные сведения, недоступные в TEX.
- (2) Когда используются регистры счётчика, размера, клея, мат. клея рамки или лексемы с номером в диапазоне 256–32767, ε -TEX будет использо-

²В русском переводе «Всё про TEX». — Прим. И. Б.

³Чтобы быть точным, все переменные состояния устанавливаются в нуль, когда eINITEX или eVIRTEX начинает работу; целые параметры, не являющиеся переменными состояниями, или устанавливаются в нуль при запуске eINITEX без загрузки форматного файла, или наследуются из форматного файла.

вать один из своих дополнительных регистров, а не выводить сообщения об ошибке и использовать регистр с номером нуль, как \TeX .

3.2 Улучшения работы (Optimization)

Когда значение присваивается (внутренней величине) в `save` группе, предыдущее значение восстанавливается в после окончания группы, обеспечивая неглобальность присвоения. Это выполняется сохранением предыдущего значения в «стеке сохранений» («`save stack`») \TeX 'а. ε - \TeX не сохраняет в этом стеке элементы, когда старое и новое значения одинаковы.

Лексемы `\aftergroup` также сохраняются в стеке сохранений \TeX 'а. Когда текущая группа заканчивается, \TeX преобразует каждую лексему `\aftergroup` в список лексем и вставляет этот список как новый «уровень ввода» [`input level`] во входной стек. ε - \TeX собирает все лексемы `\aftergroup` из одной группы в один список лексем и таким образом сбегает уровни ввода.

Когда полная страница пишется в DVI-файл, \TeX умножает соответствующие составляющие растяжимости или сжимаемости узлов клея в рамке на коэффициент расширения клея [`glue expansion factor`] этой рамки и преобразует итог в единицы DVI. Чтобы избежать переполнения, каждое итоговое значение x искусственно ограничивается величиной $|x| \leq 10^9$. Рассмотрим пример:

```
\shipout\vbox to100pt{
  \hrule width10pt
  \vskip 0pt plus1000fil
  \vskip 0pt plus1000fil
  \vskip 0pt plus-2000fil
  \hrule
  \vskip 0pt plus0.00005fil
}
```

Здесь три клея между двумя линейками сводятся к нулю; когда \TeX преобразует каждую составляющую растяжимости отдельно, они будут доходить до 10^9 единиц DVI благодаря упомянутому выше усечению. ε - \TeX собирает соответствующие составляющие растяжимости и сжимаемости следующих друг за другом узлов клея (возможно разделённых узлами вставки, метки, керны, подгонки [`adjust`] или штрафа) перед преобразованием их в DVI единицы. Во время этого узлы клея могут преобразовываться в соответствующие узлы керны, и некоторые описания клея могут быть удалены; это может повлиять на данные об использовании памяти, отображаемой после вывода страницы в dvi-файл.

3.3 Трассировка и диагностика

Когда `\tracingcommands` имеет значение 3 и выше, также показываются команды с префиксом (`\global`, и т. д.), например:

```
\global\count0=0    =>  {\global}
                       {\count}
```

Когда `\tracinglostchars` имеет значение 2 и выше, на терминале показываются пропущенные символы, даже если значение `\tracingonline` есть 0 или меньше.

Когда `\tracingscantokens` имеет значение 1 и выше, открытие и закрытие псевдофайлов (создаваемых командой `\scantokens`) записывается также для любого другого файла, с ‘_’ в качестве файла имени.

Когда программа компилируется с кодом для сбора статистики, и `\tracingassigns` имеет значение 1 и выше, все присвоения, подчинённые механизму группирования Т_ЭX, отслеживаются, например:

```
\def\foo{\relax}    =>  {changing \foo=undefined}
                       {into \foo=macro:->\relax }
\global\count17=7  =>  {globally changing \count17=0}
                       {into \count17=7}
\count17=7         =>  {reassigning \count17=7}
```

Когда `\tracingifs` имеет значение 1 и выше, отслеживаются все условия (включая `\unless`, `\or`, `\else` и `\fi`), вместе с начальной строкой и уровнем вложения; команда `\showifs` показывает состояние всех текущих активных условий. Таким образом, ввод

```
\unless\iffalse
  \iffalse
  \else
  \showifs
\fi
\fi
```

может породить

```
{\unless\iffalse: (level 1) entered on line 1}
{\iffalse: (level 2) entered on line 2}
{\else: \iffalse (level 2) entered on line 2}
### level 2: \iffalse\else entered on line 2
### level 1: \unless\iffalse entered on line 1
{\fi: \iffalse (level 2) entered on line 2}
{\fi: \unless\iffalse (level 1) entered on line 1}
```

Когда `\tracinggroups` имеет значение 1 и выше, начало и конец каждой группы сохранения отслеживается, вместе с начальной строкой и уровнем группирования; команда `\showgroups` отображает состояние всех текущих активных групп сохранения. Таким образом, ввод

```
\begingroup
{
  \showgroups
```

```

    }
  \endgroup

```

может породить

```

{entering semi simple group (level 1) at line 1}
{entering simple group (level 2) at line 2}
### simple group (level 2) entered at line 1 ({)
### semi simple group (level 1) entered at line 1 (\begingroup)
### bottom level
{leaving simple group (level 2) entered at line 2}
{leaving semi simple group (level 1) entered at line 1}

```

Иногда условия и/или группы сохранения могут неверно быть вложены по отношению к `\input` файлам. Хотя это могло бы быть совершенно законно, такие отклонения обычно непреднамеренны и могут вызвать совершенно непонятные ошибки. Когда `\tracingnesting` имеет значение 1 и выше, эти отклонения показываются; Когда `\tracingnesting` имеет значение 2 и выше, также показывается текущий контекст [current context] (отслеживаемый снизу вверх). Так, ввод

```

\newlinechar='^^J
\begingroup
  \iftrue
    \scantokens{%
  \endgroup
^^J\fi
^^J\bgroup
  ^^ \tracingnesting=2
  ^^J\iffalse
  ^^J\else
    }%
  \egroup
\fi

```

может породить⁴

```

Warning: end of semi simple group (level 1) entered at line 2 of
a different file
Warning: end of \iftrue entered on line 3 of a different file
Warning: end of file when simple group (level 1) entered at line
3 is incomplete
Warning: end of file when \iffalse\else entered on line 5 is inc
omplete
1.7 \else

1.11    }
        %

```

⁴команда `\scantokens` обсуждается после.

Команда `\showtokens{⟨список лексем⟩}` отображает список лексем, и позволяет отображать величины, которые не могут отображаться командами `\show` и `\showthe`, например:

```
\showtokens\expandafter{\jobname}
\showtokens\expandafter{\topmarks 27}
```

3.4 Запросы статуса

Множеству внутренних величин \TeX 'а можно присвоить значения, но эти значения нельзя прочитать в \TeX . ε - \TeX вводит несколько новых примитивов, которые возвращают сведения об их внутреннем состоянии.

`\eTeXversion` возвращает старший номер версии ε - \TeX 'а;

`\eTeXrevision` раскрывает в список символьных лексем, представляющих номер пересмотра (младший). Таким образом,

```
\message{\number\eTeXversion\eTeXrevision}
```

запишет полный номер версии, показываемый ε - \TeX в начале работы.

Когда используется как номер, `\interactionmode` возвращает одно из значений 0 (batchmode), 1 (nonstopmode), 2 (scrollmode) или 3 (errorstopmode).

Назначение одного из этих значений `\interactionmode` изменяет текущий уровень взаимодействия; такие присвоения всегда глобальны.

`\currentgrouplevel` возвращает текущий уровень группы сохранения;

`\currentgrouptype` возвращает номер типа самой внутренней группы:

| | |
|---|--|
| 0: верхний уровень (нет группы) | 9: math group |
| 1: простая группа | 10: disc group |
| 2: группа рамки hbox | 11: группа вставки |
| 3: группа подогнанной рамки hbox ⁵ | 12: группа vcenter |
| 4: группа рамки vbox | 13: math choice group |
| 5: группа рамки vtop | 14: semi simple group |
| 6: группа выравнивания | 15: math shift group |
| 7: группа noalign ⁶ | 16: группа математического math left group |
| 8: output group | |

`\currentiflevel` возвращает номер текущего активного условия;

`\currentifbranch` указывает выбранную ветвь самого внутреннего условия: 1 если условие истинно, -1 если условие ложно, или 0 если ещё не выбрано;

`\currentiftype` возвращает 0, если сейчас нет обрабатываемых условий, положительное число указывает вид самого внутреннего обрабатываемого условия, если перед условием стоял префикс `\unless`, то это число будет со знаком минус:

| | | |
|---------------------|-------------------------|--------------------------|
| 1: <code>\if</code> | 8: <code>\ifmode</code> | 15: <code>\iftrue</code> |
|---------------------|-------------------------|--------------------------|

⁵В оригинале [adjusted hbox group] — прим. И. Б.

⁶В оригинале [no align group] — прим. И. Б.

| | | |
|-------------|-------------|-----------------|
| 2: \ifcat | 9: \ifinner | 16: \iffalse |
| 3: \ifnum | 10: \ifvoid | 17: \ifcase |
| 4: \ifdim | 11: \ifhbox | 18: \ifdefined |
| 5: \ifodd | 12: \ifvbox | 19: \ifcsname |
| 6: \ifvmode | 13: \ifx | 20: \iffontchar |
| 7: \ifhmode | 14: \ifeof | |

`\lastnodetype` возвращает номер, указывающий тип последнего узла, если он был, в текущем (вертикальном горизонтальном или математическом) списке:

| | |
|--------------------------------|---------------------------------|
| -1: ни чего (пустой список) | 8: узел переноса по усмотрению |
| 0: узел символа | 9: узел нечто |
| 1: узел горизонтального списка | 10: математический узел |
| 2: узел вертикального списка | 11: узел клея |
| 3: узел линейки | 12: узел керна |
| 4: узел вставки | 13: узел штрафа |
| 5: узел метки | 14: неустановленный узел |
| 6: узел подгонки [adjust] | 15: узлы математического режима |
| 7: узел лигатуры | |

Команды `\fontcharht`, `\fontcharwd`, `\fontchardp` и `\fontcharic`, за которыми указаны шрифт и код символа, возвращают размер: высоту, ширину, глубину или поправку на курсив символа в шрифте, или `0pt`, если нет такого символа; условие `\iffontchar` проверяет, есть ли такой символ.

Когда используется как номер, `\parshape` возвращает число строк текущего описания формы абзаца (или нуль).

ε -TeX's `\parshapeindent`, `\parshapelength` и `\parshapedimen` с последующим номером n возвращает размеры описания формы абзаца:

`0pt` для $n \leq 0$, или когда не включено задание формы абзаца, иначе `\parshapeindent n` и `\parshapedimen $2n-1$` когда возвращается отступ строки n (явно указанный или подразумеваемый повторением последним описанием [last specification]),

`\parshapelength n` и `\parshapedimen $2n$` оба возвращают длину строки n .

3.5 Выражения

ε -TeX вводит нотацию выражений для типов число, размер, клей или мат. клей, которые могут использоваться везде, где нужны величины данного типа. Выражения вычисляются считывающим механизмом ε -TeX'a; они начинаются одной из команд `\numexpr`, `\dimexpr`, `\glueexpr` или `\muexpr` (определяя тип t) и заканчиваются необязательным одним `\relax` (который поглотится считывающим механизмом). Выражение состоит из одного и более членов того же типа, складываемых или вычитаемых; член типа t

состоит из коэффициента того типа, необязательно умножаемого и/или делимого на числовые коэффициенты; наконец, коэффициент, коэффициент типа t суть или выражение в скобках, или величина (число, и т. д.) этого типа. Так, условие

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

истинно тогда и только тогда, когда ширина рамки 20 превышает 32 pt. Обратите внимание, `\relax` завершает только внутреннее (числовое) выражение, внешнее выражение (размера) прерывается само собой лексемой `<_12`, которая не подпадает под синтаксис выражения.

Арифметика, выполняемая над выражениями ϵ -TeX'a не делает ни чего, что не может быть сделано арифметическими операциями TeX'a `\advance`, `\multiply` и `\divide`, хотя есть примечательные различия: каждый коэффициент проверяется на допустимость значений, числа должны быть меньше 2^{31} по абсолютной величине, размеры и составляющие клея должны быть меньше 2^{14} pt, `mu`, `fil` и т. д. соответственно. Арифметические действия выполняются отдельно, кроме действий „масштабирования“ (умножения, за которым непосредственно следует деление), которое выполняется как одна объединённая операция с 64-битным произведением в качестве промежуточного значения. Итог каждого действия опять проверяется на допустимость значений. Окончательные итоги деления и масштабирования округляются, тогда как `\divide` TeX'a усекает их.

Важная новая особенность состоит в том, что вычисление выражений происходит без присвоений и поэтому может выполняться в обстоятельствах, где присвоения недопустимы, например, внутри `\edef` или `\write`. Это также позволяет определения чисто раскрываемых конструкций циклов:

```
\def\foo#1#2{\number#1
  \ifnum#1<#2,
    \expandafter\foo
    \expandafter{\number\numexpr#1+1\expandafter}%
    \expandafter{\number#2\expandafter}%
  \fi}
```

так, например, `'\foo{7}{13}'` раскроется в `'7, 8, 9, 10, 11, 12, 13'`.

Команды `\gluestretch` и `\glueshrink` возвращают растяжимость и сжимаемость следующего за ними клея, как размеры (единицы `fil` и т. д. заменяются на `pt`), команды `\gluestretchorder` и `\glueshrinkorder` возвращают порядок бесконечности: 0 для `pt`, 1 для `fil`, 2 для `fill` и 3 для `filll`.

Команды `\gluetomu` и `\mutogluе` преобразуют клей в мат.клей и наоборот, просто приравнивая `1pt` к `1mu` также, как делает TeX (в дополнение к сообщению об ошибке), когда используется неверный тип клея.

3.6 Дополнительные регистры и метки

ε -TeX увеличивает число регистров счётчиков, размеров, клея, мат. клея, рамок и лексем TeX'a с 256 до 32768. Дополнительные регистры имеют номера 256–32767, и могут использоваться также, как и первые 256, за исключением того, что они не могут использоваться для классов вставок.

Как в TeX, первые 256 регистров каждого вида выполнены как статические массивы, являющиеся частью «таблицы эквивалентов», значения, чтобы восстанавливаться, когда группа сохранения заканчивается, хранятся в стеке сохранений. Дополнительные регистры выполнены как разреженные массивы и строятся в основной памяти TeX'a и поэтому менее эффективны. Они используют четырёхуровневую структуру указателей, и отдельные регистры присутствуют только при необходимости. Значения для восстановления по окончании некоторой группы сохранения хранятся в связанном списке (который также строится в главной памяти) с одним элементом стека сохранения, указывающим на этот список.⁷

ε -TeX обобщает понятие меток TeX'a до классов меток 0–32767, с классом меток 0 используемым для меток TeX'a.

Команда `\marks`, за которой следует класс метки n и текст метки добавляет узел метки к текущему списку; `\marks0` суть синоним `\mark`. Построитель страниц и команда `\vsplit` записывают сведения об узлах меток, найденных на странице или произведённых рамок, отдельно для каждого класса меток. Сведения для меток класса 0 хранятся в маленьком статическом массиве, как в TeX, сведения для дополнительных классов меток снова хранятся в разреженном массиве, элементы которого присутствуют только при необходимости.

Команда `\firstmarks` n раскрывается в текст меток для первой метки класса n , встреченной на новой странице, и т. д., и снова `\firstmarks0` суть синоним `\firstmark`.

3.7 Управление вводом

Команда `\readline(число) to` (управляющая последовательность) определяет управляющую последовательность как макрос без параметров, текст замещения которого суть содержимое следующей строки, прочитанной из указанного файла, как для `\read`. Различие в том, что текущие коды категории ни на что не влияют, и все символы в этой строке (включая знак конца строки) преобразуются в символьные лексемы с категорией 12 ('иной'), кроме символа с кодом 32, получающего категорию 10 ('пробел').

Команда `\scantokens{...}` собирает список нераскрытых лексем, преобразует его в строку символов, которая рассматривается так, как если бы она была внешним файлом, и начинает читать из этого «псевдофайла». Схожее действие дают команды

⁷С тем, что порядок восстановления (или удаления) сохранённых значений может быть слегка удивительным.

```

\toks0={...}
\immediate\openout0=file
\immediate\write0{\the\toks0}
\immediate\closeout0
\input file

```

В частности, каждый встречающийся текущий символ начала новой строки рассматривается как начало новой строки, и входные символы будут преобразовываться в лексемы как обычно. Команда `\scantokens`, однако, раскрываема и не использует регистры лексем, потоки записи, или внешние файлы. Преобразование из внутренних ASCII кодов `TeX`'а во внешние и обратно не производится. Наконец, текущий контекст (отслеживание снизу вверх) показываемый, например, как часть сообщения об ошибке идёт за пределы входной строки из псевдофайла, пока не найдётся входная строка из настоящего файла (или с терминала).

Когда механизм ввода ϵ -`TeX`'а пытается читать за концом вводимого (`\input`) файла или `\scantokens` псевдофайла, и перед проверкой на 'сбежавшие' условия и закрытием файла, он сперва читает список лексем, предопределённых командой `\everyeof={〈список лексем〉}`.

3.8 Разбиение абзацев на строки

Традиционно вёрстка со свинцовой печатью использовала для подгонки (растяжки или сжатия) межсловных пробелов в последней строке абзаца ту же величину, что и в предыдущей строке. Однако, в `TeX` последняя строка обычно верстается своей естественной ширины благодаря бесконечной растяжимости клея `parfillskip`. ϵ -`TeX` позволяет интерполировать между этими двумя крайностями, указав подходящее значение для `\lastlinefit`. Для значений 0 и меньше, ϵ -`TeX` ведёт себя как `TeX`, значения от 1 до 1000 указывает коэффициент поправки клея в тысячных долях f , значения выше 1000 воспринимаются как $f = 1$.

Новый алгоритм используется только если

1. `\lastlinefit` положительно;
2. `\parfillskip` имеет бесконечную растяжимость; и
3. растяжимость `\leftskip` плюс `\rightskip` конечна.⁸

Таким образом, последняя строка абзаца будет свёрстана обычным образом с естественной шириной, и растяжимость клея `parfillskip` будет использоваться, чтобы достичь желаемой ширины строки. Этот алгоритм действует, как обычно, рассматривая все возможные последовательности возможных точек разрыва и собирает дефектности (`demerits`) для растяжимости или сжимаемости строке, а также визуальной несовместимости строк. Когда кандидат на последнюю строку найден, проверяются следующие условия:

4. предыдущая строка не «бесконечно плоха» и была растянута с положительной конечной растяжимостью или сжата с положительной сжимаемо-

⁸Как обычно для параметров, влияющих на алгоритм разбивки строк `TeX`'а, используются значения, текущие в конце (неполного) абзаца.

стью;

5. последняя строка бесконечно растяжима всецело благодаря клею `parfillskip`;

6. если предпоследняя строка сжата или растянута, то последняя, соответственно, имеет положительную конечную сжимаемость или растяжимость. Если все три условия удовлетворены, коэффициент подгонки клея, использованный для предыдущей строки, с коэффициентом f применяется к соответствующим компонентам растяжимости или сжимаемости всех узлов клея в последней строке, и вычисляются соответствующие штрафы. (Тем не менее, последняя строка не будет растягиваться больше заданной ширины.)

Когда все возможные кандидаты на последнюю строку абзаца проверены, выбирается одна, имеющая наименьшую общую дефектность [demerits]. Если используется изменённый алгоритм ε -TeX'a к последней строке, действительная растяжимость или сжимаемость устанавливается соответствующим изменением узла клея `parfillskip`.

Все описанные вычисления выполняются с машинно-независимой целочисленной арифметикой. Заметьте, что действительная растяжимость требует машинно-зависимой арифметики с плавающей запятой. Поэтому, когда абзац прерывается выключенной формулой, и строка, предшествующая выключенной формуле, подлежит подгонке, как только что описано, выключенной формуле (`display`) будет вообще предшествовать клей `abovedisplayskip`, а не `abovedisplayshortskip`.

После разрыва абзаца на строки, TeX вычисляет межстрочные штрафы, добавляя значения:

```
\interlinepenalty между двумя любыми строками,  
\clubpenalty после первой строки (неполного) абзаца,  
\widowpenalty перед последней строкой абзаца,  
\displaywidowpenalty перед строкой, непосредственно предшествующей  
выключенной формуле, и  
\brokenpenalty после строк, заканчивающихся разрывом по усмотрению.  
 $\varepsilon$ -TeX обобщает идею этих interline, club, widow и display widow штрафов,  
позволяя заменять их массивами штрафных значений командами  
\interlinepenalties,  
\clubpenalties,  
\widowpenalties и  
\displaywidowpenalties.
```

За каждой из этих команд должен следовать необязательный знак равенства и число n . Если $n \leq 0$, соответствующий массив сбрасывается, и соответствующее простое значение TeX'a используется как обычно; положительное значение n объявляет массив длины n , и за ним должно следовать n значений штрафов. Когда один из этих массивов установлен, его значения используются взамен соответствующего простого значения TeX'a следующим образом (повторяя последнее значение при необходимости):

i - значение межстрочного штрафа используется после строки i абзаца;

i - значение `club` штрафа используется после i строки неполного абзаца;

i^- значение `widow` штрафа используется после $m - i$ строки абзаца без выключенных уравнений или последнего неполного абзаца длины m ;
 i^- значение `display widow` штрафа используется после строки $m - i$ неполного абзаца длины m , за которой следует выключенное уравнение.

Эти же четыре команды выдают массивы штрафов, когда используются после `\the`, или в случаях, где Т_ЭX ждёт число. Указав, например, `\clubpenalties`(число) с числом n вернёт 0 для $n < 0$, или когда массив `club` штрафов сброшен, длина объявленного массива `club` штрафов для $n = 0$ или n^- значение `club` штрафа для $n > 0$ (повторяя последнее значение, при необходимости).

3.9 Математические формулы

Т_ЭX'овские `\left`(разделитель)...`\right`(разделитель) дают два разделителя с общим размером, настроенным на высоту и глубину заключённого в них материала. В ε -Т_ЭX это обобщается командами `\middle`(разделитель), делящими заключённый материал на части, и дающими в итоге последовательности разделителей с общим размером, настроенным на наибольшую высоту и глубину заключённых в них частей. Пробелы между частью и разделителем слева или справа такие же, как и для Т_ЭX'овских левого или правого разделителей, соответственно.

3.10 Переносы

Т_ЭX использует значения `\lccode` для двух совершенно различных целей:

(1) когда `\lowercase` преобразует лексемы символов в их эквивалентные нижнего регистра (таким же образом `\uppercase` использует значения `\uccode`);

и

(2) когда образцы переноса или исключения читаются, и когда слова переносятся во время работы алгоритма разбиения на строки.

ε -Т_ЭX вводит понятие (зависимых от языка) кодов переноса слов, которые используются взамен значений `\lccode` для целей переноса. Чтобы объяснить подробности поведения ε -Т_ЭX'а, нам нужны некоторые технические подробности образцов переноса. Когда INITEX начинает работу, не читая форматный файл, (исходно пустой) образцы переноса находятся в форме, пригодной для вставки новых шаблонов, указываемых командами `\patterns`; когда INITEX пытается выполнить перенос или готовится записать форматный файл, они сжимаются в более краткую форму, пригодную для поиска переносов. Только эти сжатые шаблоны могут считываться из форматного файла (INITEX'ом или VIRTEX'ом).

В ε -Т_ЭX образцы переносов дополняются кодами переносов. Когда eINITEX начинает, не читая форматный файл они обе исходно пусты; когда команда `\patterns` выполняется, и `\savingshyphcodes` положительно, текущие значения `\lccode` сохраняются как коды переносов для текущего языка. Эти сохранённые коды переносов позднее сжимаются вместе с образцами и пишутся в форматный файл или читаются из него. Когда образцы сжаты

(всегда верно для eVIRTEX), и коды переноса сохранены для текущего языка, они используются взамен значений `\lccode` для целей переноса (чтение исключений переносов и переносимых слов).

3.11 Отбрасываемые элементы

Когда построитель страниц \TeX 'а переносит (вертикальный режим) материал из «недавних вкладов» в «страницу до сих пор», он отбрасывает узлы клея, керны и штрафа (отбрасываемые элементы), предшествующие первой рамке или линейке на строящейся странице, и вставляет узел `topskip` клея прямо перед этой рамкой или линейкой. Заметьте, что клей `topskip` не обязательно будет первым узлом на странице, ему могут предшествовать узлы вставки, метки и нечто. Также, когда команда `\vsplit` отделила первую часть вертикальной рамки, отбрасываемые элементы удаляются из вершины остающейся вертикальной рамки, и узел клея `splittopskip` вставляется сразу перед первой рамкой или линейкой.

Когда параметру ϵ - \TeX 'а `\savingdiscards` присвоено положительное значение, эти «отбрасываемые элементы» сохраняются в двух списках и могут восстанавливаться командами `\pagediscards` и `\splitdiscards`, которые действуют как «unboxing» гипотетических регистров рамок, содержащих вертикальную рамку с отброшенными элементами.

Список элементов, отбрасываемых построителем страниц, освобождается в конце программы вывода или командой `\pagediscards`; новые элементы могут добавляться, пока новая «страница до сих пор» [‘page so far’] не содержит рамку или линейку.

Список элементов, отброшенных командой `\vsplit`, освобождается в начале операции `vsplit` или командой `\splitdiscards`; новые элементы добавляются в конце операции `vsplit`.

3.12 Раскрываемые команды

Глава 20 *The \TeX book* даёт полный список всех раскрываемых команд \TeX 'а и всех тех случаев, где раскрываемые лексемы не раскрываются. Для ϵ - \TeX есть следующие дополнительные условия:

- `\ifdefined⟨лексема⟩` (проверяет, определена ли лексема)

Истина, если `⟨лексема⟩` определена; не создаёт новых элементов в хэш-таблице.

- `\ifcsname... \endcsname` (проверяет, определена ли управляющая последовательность)

Истина, если управляющая последовательность `\csname... \endcsname` определена; не создаёт новых элементов в хэш-таблице.

- `\iffontchar⟨шрифт⟩⟨8-битное число⟩` (проверяет, есть ли символ)

Истина, если символ `\char`(8-битное число) в `\font`(шрифт) существует.

Это дополнительные раскрываемые команды ϵ -TeX'a:

- `\unless`.
Следующая (unexpanded) лексема должна быть булевым условием (т. е., не `\ifcase`); истинное значение этого условия обращается (делается обратным по сравнению с истинным).
- `\TeXrevision`.
Раскрытие этого суть список лексем символов категории 12 («иной символ»), представляющих номер переработки ϵ -TeX'a (младшая часть номера), например, '.0' или '.1'.
- `\topmarks`(15-битное число), `\firstmarks`(15-битное число), `\botmarks`(15-битное число), `\splitfirstmarks`(15-битное число) и `\splitbotmarks`(15-битное число).
Эти команды обобщают TeX'овские `\topmark` и т. д. до 32768 различных классов меток; особый случай `\topmarks0` суть синоним `\topmark` и т. д.
- `\unexpanded`(обычный текст).
Раскрытие суть список лексем (сбалансированный текст).
- `\detokenize`(обычный текст).
Раскрытие суть список символьных лексем, представляющих список лексем (сбалансированный текст). Как со списками символьных лексем производимых TeX'овским `\the` и ϵ -TeX'овским `\readline`, эти лексемы имеют категорию 12 («иной»), кроме символа с кодом 32, дающего категорию 10 («пробел»).
- `\scantokens`(обычный текст).
Раскрытие пусто, но ϵ -TeX создаёт псевдофайл, содержащий символы, представляющие список лексем (сбалансированный текст) и готовится читать из этого псевдофайла перед тем, как читать лексемы из его текущего источника.

Есть дополнительные случаи ϵ -TeX, когда раскрываемые лексемы не раскрываются:

- Когда ϵ -TeX читает лексему аргумента для `\ifdefined`.
- Когда ϵ -TeX собирает список лексем для `\unexpanded`, `\detokenize`, `\scantokens` или `\showtokens`.
- Защищённые макросы (определённые с префиксом `\protected`) не раскрываются, когда строится раскрываемый список лексем (для `\edef`, `\xdef`, `\message`,

`\errmessage`, `\special`, `\mark`, `\marks`, или когда пишется список лексем для `\write` в файл), или когда готовится в выравнивании для `\noalign` или `\omit`.⁹

- Когда строится список раскрываемых лексем, лексемы, полученные из раскрытия `\unexpanded`, не раскрываются далее (так же ведут себя лексемы, полученные из раскрытия `\the`(переменная лексемы) как в \TeX , так и в $\varepsilon\text{-}\TeX$).

4 Улучшения $\varepsilon\text{-}\TeX$

Выполнение большинства новых примитивов, связанных с улучшениями, запрещено, когда соответствующее улучшение не разрешено, и ведёт к сообщению об ошибке `'Improper...'`. Негодная команда тем не менее может оказать некоторое влияние, например, перевести $\varepsilon\text{-}\TeX$ в горизонтальный режим.

4.1 Вёрстка текста в разных направлениях

Эта возможность обеспечивает вёрстку текста слева направо и справа налево и вводит четыре примитива для установки направления текста `\beginL`, `\endL`, `\beginR` и `\endR`. Встраиваемый код отличается от использованного в $\TeX\text{-}\XET$ [6].

Чтобы избежать конфликтов с $\TeX\text{-}\XET$ настоящее воплощение вёрстки текста в разных направлениях называется $\TeX\text{-}\text{-}\XET$. Оно использует те же примитивы направлений текста, но отличается от $\TeX\text{-}\XET$ в нескольких важных частях:

- (1) $\varepsilon\text{-}\TeX$ обращает текст справа налево явно и записывает в обычный DVI файл без команд каких-либо `begin_reflect` или `end_reflect`;
- (2) математический узел `is (ab)used` взамен узла «нечто», чтобы записывать примитивы изменения направления текста для того, чтобы минимизировать влияние на алгоритм разрыва строк для чистого текста слева направо;
- (3) текст справа налево прерывается выделенным уравнением и автоматически возобновляется после него;
- (4) материал выключенной формулы всегда печатается слева направо, даже в конструкциях вроде этой:

```
\hbox{\beginR\ vbox{\noindent$$abc\eqno(123)$$}\endR}
```

$\TeX\text{-}\text{-}\XET$ разрешается или запрещается присвоением положительного или неположительного значения, соответственно, переменной состояния `\TeXXeTstate`. Пока $\TeX\text{-}\text{-}\XET$ запрещён, $\varepsilon\text{-}\TeX$ и $\TeX\text{-}\text{-}\text{-}\XET$ строят горизонтальные списки и абзацы точно также. Даже $\TeX\text{-}\text{-}\XET$, в общем, выдаст те же результаты,

⁹Несмотря на то, что защищённые макросы были введены в $\varepsilon\text{-}\TeX$ версии 1, подавление их раскрытия в выравниваниях было введено в версии 2.

что и \TeX для чистого текста слева направо. Есть, однако, обстоятельства, при которых могут возникнуть некоторые различия. Это лучше всего показывает пример:

```
\vbox{\noindent
      $\hfil\break
      \null\hfil\break
      \null$\par
```

Здесь \TeX произведёт три строки, содержащие следующие узлы:

1. `mathon`, клей `hfil`, штраф за разрыв [`break penalty`] и клей `rightskip`;
2. пустую `hbox`, клей `hfil`, штраф за разрыв [`break penalty`] и клей `rightskip`;
3. пустую `hbox`, `mathoff`, штраф `nobreak`, клей `parfillskip` и клей `rightskip`.

Эти строки можно получить через:

```
\setbox3=\lastbox
\unskip\unpenalty
\setbox2=\lastbox
\unskip\unpenalty
\setbox1=\lastbox
```

Позднее эти строки могут распакованы [`'unhboxed'`] как часть нового абзаца и возможно проанализировать из содержимое. Как следствие, в \TeX (и $\varepsilon\text{-}\TeX$ в режиме совместимости) могут быть горизонтальные списки, где узлы `mathon` и `mathoff` не спарены должным образом. Поэтому \TeX может пытаться переносить «слова» принадлежащие исходно математическому режиму, или запретить перенос слов, исходно принадлежащих горизонтальному режиму.

Материал математического режима всегда верстается слева направо \TeX - \TeX -’ом, даже когда он содержится внутри текста справа налево. Поэтому \TeX -- \XET вставит дополнительные математические узлы `beginM` и `endM`, так что материал, начинающийся с математического узла, всегда заключён между правильно спаренными узлами. Поэтому \TeX -- \XET никогда не пытается ни переносить «слова», начинающиеся с математического узла, ни предотвращать перенос слов, начинающихся в горизонтальном режиме.

Дополнительные математические узлы, введённые \TeX -- \XET , прозрачны для таких операций, как `\lastpenalty`, проверяющих или удаляющих последний узел горизонтального списка.¹⁰

Когда \TeX -- \XET включается или отключается при построении рамки, рамка может содержать неверно спаренные команды направления текста [`text-direction directives`] или математические узлы. Такие непарные узлы могут вызывать предупреждения, когда рамка отправляется в `dvi`-файл [`shipped out`]. Поэтому следует включать и отключать \TeX -- \XET только в вертикальном режиме.

¹⁰Это не так для некоторых ранних воплощений \TeX -- \XET .

5 Синтаксис расширений для ε -TeX

5.1 Команды, независимые от режима

Синтаксис независимых от режима команд TeX'a, описанных в первой части главы 24 *The TeX book*, расширен изменениями существующих и добавлениями новых команд.

Первое, ε -TeX имеет 32768 `\count`, `\dimen`, `\skip`, `\muskip`, `\box` и `\toks` регистров вместо TeX'овских 256. Поэтому он позволяет $\langle 15$ -битное число \rangle вместо $\langle 8$ -битного числа \rangle почти во всех синтаксических конструкциях, относящихся к этим регистрам, исключение составляют команда `\insert`: классы вставок ограничены областью 0–254 в ε -TeX, так же как и в TeX.

Следующее, ε -TeX расширяет список внутренних величин TeX'a:

\langle внутреннее целое $\rangle \rightarrow$ всё, что определяет *The TeX book* | `\TeXversion`
| `\interactionmode` | \langle штрафы \rangle \langle число \rangle
| `\lastnodetype` | `\currentgrouplevel` | `\currentgrouptype`
| `\currentiflevel` | `\currentiftype` | `\currentifbranch`
| `\gluestretchorder` \langle клей \rangle | `\glueshrinkorder` \langle клей \rangle
| `\numexpr` \langle целое выражение \rangle \langle необязательные пробелы и `\relax` \rangle
 \langle штрафы $\rangle \rightarrow$ `\interlinepenalties` | `\clubpenalties`
| `\widowpenalties` | `\displaywidowpenalties`
 \langle внутренний размер $\rangle \rightarrow$ всё, что определяет *The TeX book*
| `\parshapeindent` \langle число \rangle | `\parshapelength` \langle число \rangle
| `\parshapedimen` \langle число \rangle
| `\gluestretch` \langle клей \rangle | `\glueshrink` \langle клей \rangle
| `\fontcharht` \langle шрифт \rangle \langle 8-битное число \rangle | `\fontcharwd` \langle шрифт \rangle \langle 8-битное число \rangle

| `\fontchardp` \langle шрифт \rangle \langle 8-битное число \rangle | `\fontcharic` \langle шрифт \rangle \langle 8-битное число \rangle

| `\dimexpr` \langle выражение размеров \rangle \langle необязательные пробелы и `\relax` \rangle
 \langle внутренний клей $\rangle \rightarrow$ всё, что определяет *The TeX book* | `\mutoglu` \langle клей \rangle

| `\glueexpr` \langle выражение клея \rangle \langle необязательные пробелы и `\relax` \rangle
 \langle внутренний мат. клей $\rangle \rightarrow$ всё, что определяет *The TeX book* | `\gluetomu` \langle клей \rangle

| `\muexpr` \langle выражение мат. клеев \rangle \langle необязательные пробелы и `\relax` \rangle

Дополнительные возможности для \langle целого параметра \rangle :

`\TeXxTstate` (положительное, если разрешён набор текста слева направо и с права налево)
`\tracingassigns` (положительное, если показываются присвоения)
`\tracinggroups` (положительное, если показываются группы сохранения)
`\tracingifs` (положительное, если показываются условия)
`\tracingscantokens` (положительно если показывается открытие и закрытие псевдофайлов `\scantokens`)

`\tracingnesting` (положительно если показывается негодное вложение групп и условий в файлах)
`\predisplaydirection` (направление текста, предшествовавшее выключенной формуле)
`\lastlinefit` (коэффициент поправки для последней строки абзаца, умноженное 1000)
`\savingsdiscards` (положительно, если сохраняемые элементы отбрасываются из вертикальных списком)
`\savingshyphcodes` (положительно, если `\patterns` сохраняет значения `\lccode`, как коды переноса)

Заметьте, что переменная состояния ε -TeX `\TeXeTstate` (пока только одна) суть \langle целый параметр \rangle . Это не обязательно будет иметь место для всех будущих переменных состояния; возможно, некоторые будущие улучшения будут разрешаться и запрещаться не только глобально, и будут подлежать механизму группирования.

Дополнительные возможности для \langle списка параметров \rangle :

`\everyeof` (лексемы для вставки, когда `\input` файл закончится)

Вот синтаксис выражений ε -TeX'a:

\langle целое выражение $\rangle \rightarrow \langle$ целый терм \rangle
 | \langle целое выражение \rangle \langle сл. или выч. \rangle \langle целый терм \rangle
 \langle целый терм $\rangle \rightarrow \langle$ целый коэффициент \rangle
 | \langle целый терм \rangle \langle умн. или дел. \rangle \langle целый коэффициент \rangle
 \langle целый коэффициент $\rangle \rightarrow \langle$ число \rangle
 | \langle левая скобка \rangle \langle целое выражение \rangle \langle правая скобка \rangle
 \langle выражение размеров $\rangle \rightarrow \langle$ терм размера \rangle
 | \langle выражение размеров \rangle \langle сл. или выч. \rangle \langle терм размера \rangle
 \langle терм размера $\rangle \rightarrow \langle$ коэффициент размера \rangle
 | \langle терм размера \rangle \langle умн. или дел. \rangle \langle целый коэффициент \rangle
 \langle коэффициент размера $\rangle \rightarrow \langle$ размер \rangle
 | \langle левая скобка \rangle \langle выражение размеров \rangle \langle правая скобка \rangle
 \langle выражение клея $\rangle \rightarrow \langle$ терм клея \rangle
 | \langle выражение клея \rangle \langle сл. или выч. \rangle \langle терм клея \rangle
 \langle терм клея $\rangle \rightarrow \langle$ коэффициент клея \rangle
 | \langle терм клея \rangle \langle умн. или дел. \rangle \langle целый коэффициент \rangle
 \langle коэффициент клея $\rangle \rightarrow \langle$ клей \rangle
 | \langle левая скобка \rangle \langle выражение клея \rangle \langle правая скобка \rangle
 \langle выражение мат. клеев $\rangle \rightarrow \langle$ терм мат. клея \rangle
 | \langle выражение мат. клеев \rangle \langle сл. или выч. \rangle \langle терм мат. клея \rangle
 \langle терм мат. клея $\rangle \rightarrow \langle$ коэффициент мат. клея \rangle
 | \langle терм мат. клея \rangle \langle умн. или дел. \rangle \langle целый коэффициент \rangle
 \langle коэффициент мат. клея $\rangle \rightarrow \langle$ клей \rangle
 | \langle левая скобка \rangle \langle выражение мат. клеев \rangle \langle правая скобка \rangle
 \langle необязательные пробелы и `\relax` $\rangle \rightarrow \langle$ необязательные пробелы \rangle
 | \langle необязательные пробелы \rangle `\relax`
 \langle сл. или выч. $\rangle \rightarrow \langle$ необязательные пробелы $\rangle_{+12} \mid \langle$ необязательные пробелы \rangle_{-12}

$\langle \text{умн. или дел.} \rangle \longrightarrow \langle \text{необязательные пробелы} \rangle *_{12} \mid \langle \text{необязательные пробелы} \rangle /_{12}$
 $\langle \text{левая скобка} \rangle \longrightarrow \langle \text{необязательные пробелы} \rangle (_{12}$
 $\langle \text{правая скобка} \rangle \longrightarrow \langle \text{необязательные пробелы} \rangle)_{12}$

Далее, ε -TeX расширяет синтаксис для присвоений:

$\langle \text{префикс} \rangle \longrightarrow \text{всё, что определяет } The \text{ TeXbook} \mid \backslash \text{protected}$
 $\langle \text{простое присвоение} \rangle \longrightarrow \text{всё, что определяет } The \text{ TeXbook}$
 $\quad \mid \langle \text{присвоение штрафа} \rangle$
 $\quad \mid \backslash \text{readline} \langle \text{число} \rangle \text{ to } \langle \text{управляющая последовательность} \rangle$
 $\langle \text{присвоение штрафа} \rangle \longrightarrow \langle \text{штрафы} \rangle \langle \text{равно} \rangle \langle \text{число} \rangle \langle \text{значения штрафов} \rangle$
 $\langle \text{присвоение режима взаимодействия} \rangle \longrightarrow \text{всё, что определяет } The \text{ TeXbook}$
 $\quad \mid \backslash \text{interactionmode} \langle \text{равно} \rangle \langle \text{2-битное число} \rangle$

В $\langle \text{присвоении штрафов} \rangle$, для которых $\langle \text{число} \rangle$ суть n , $\langle \text{значения штрафов} \rangle$ суть $\langle \text{пусто} \rangle$, если $n \leq 0$, иначе они состоят из n последовательных вхождений $\langle \text{числа} \rangle$.

Наконец, оставшиеся независимые от режима команды ε -TeX:

- $\backslash \text{showgroups}$, $\backslash \text{showifs}$, $\backslash \text{showtokens}$ $\langle \text{обычный текст} \rangle$. Эти команды предназначены для того, чтоб помочь вам понять, что ε -TeX думает, он делает. Команды $\backslash \text{showtokens}$ отображают список лексем $\langle \text{сбалансированный текст} \rangle$.
- $\backslash \text{marks}$ $\langle \text{15-битное число} \rangle \langle \text{обычный текст} \rangle$. Эта команда обобщает команду TeX'a $\backslash \text{mark}$ до 32768 различных классов меток; особый случай $\backslash \text{marks0}$ суть синоним $\backslash \text{mark}$.

5.2 Команды вертикального режима

Синтаксис команд вертикального режима TeX'a, описанных во второй части главы 24 *The TeXbook*, расширен ε -TeX так:

- $\backslash \text{pagediscards}$, $\backslash \text{splitdiscards}$. Эти две команды подобны $\backslash \text{unvbox}$. Когда $\backslash \text{savingvdiscards}$ положительно, элементы, отброшенные строителем страниц и командой $\backslash \text{vsplit}$, собираются в два особых списка. Один из этих особых списков добавляется к текущему вертикальному списку (таким же образом, как $\backslash \text{unvbox}$ добавляет вертикальный список внутрь вертикальной рамки) и становится пустым.
- Вот дополнительные возможности для $\langle \text{команд горизонтального режима} \rangle$:

$\langle \text{команда горизонтального режима} \rangle \longrightarrow \text{всё, что определяет } The \text{ TeXbook}$
 $\quad \mid \backslash \text{beginL} \mid \backslash \text{endL} \mid \backslash \text{beginR} \mid \backslash \text{endR}$

5.3 Команды горизонтального режима

Синтаксис команд горизонтального режима \TeX 'а, описанный в главе 25 *The \TeX book*, расширен ε - \TeX так:

- Вот дополнительные возможности для \langle команд вертикального режима \rangle :

\langle команда вертикального режима $\rangle \rightarrow$ всё, что определяет *The \TeX book*
| `\pagediscards` | `\splitdiscards`

- `\beginL`, `\endL`, `\beginR`, `\endR` (команды управления направлением текста).

Использование этих команд недопустимо, если улучшение \TeX --Х \TeX запрещено; иначе `\beginL`, и т. п. узел направления текста (новый вид математического узла) добавляется к текущему горизонтальному списку. Эти узлы ограничивают начало и конец сегментов горизонтального списка, содержащих текст слева-направо (L) или справа-налево (R). Перед абзацем, разбиваемым на строки, узлы `\endL` и `\endR` добавляются, чтобы завершить любой незавершённый L или R сегмент; когда абзац продолжается после выключенной формулы, любые такие незавершённые сегменты возобновляются сами, начиная новый горизонтальный список с нужного узла `\beginL` или `\beginR`.

- `\marks`(15-битное число) \langle обычный текст \rangle . Эта команда обобщает команду \TeX 'а `\mark` до 32768 различных классов меток; особый случай `\marks0` суть синоним для `\mark`.

5.4 Команды математического режима

Синтаксис команд \TeX 'а для математического режима, описанный в главе 26 *The \TeX book* («Всё про \TeX »), расширен в ε - \TeX так:

- `\left` \langle разделитель \rangle \langle материал мат. режима \rangle
`\middle` \langle разделитель \rangle \langle материал мат. режима $\rangle \dots \code{\right} \langle разделитель \rangle
(Обобщая \TeX 'овский `\left` \langle разделитель \rangle \langle материал мат. режима \rangle `\right` \langle разделитель \rangle).$

Для каждого \langle материала мат. режима \rangle ε - \TeX начинает новую группу, начиная с нового математического списка (всегда в том же стиле), который начинается на левой границе, содержащей всё обработанное до сих пор. Эта группа должна заканчиваться или «`\middle`», или «`\right`», каждый раз, когда внутренний математический список завершается новым элементом границы, содержащим новый разделитель. В случае «`\middle`», новая группа начинается опять, в случае «`\right`», ε - \TeX добавляет внутренний атом в текущий список; ядра этого атома содержат только что завершённый внутренний математический список.

Список литературы

- [1] *A torture test for T_EX*, by Donald E. Knuth, Stanford Computer Science Report 1027.
- [2] *A torture test for ε -T_EX*, by The $\mathcal{N}\mathcal{J}\mathcal{S}$ Team (Peter Breitenlohner and Bernd Raichle). Version 2, January 1998.
- [3] *The WEB system of structured documentation*, by Donald E. Knuth, Stanford Computer Science Report 980.
- [4] *How to generate ε -T_EX*, by The $\mathcal{N}\mathcal{J}\mathcal{S}$ Team (Peter Breitenlohner and Phil Taylor). Version 2, January 1998.
- [5] *The T_EXbook* (Computers and Typesetting, Vol. A), by Donald E. Knuth, Addison Wesley, Reading, Massachusetts, 1986.
- [6] *Mixing right-to-left texts with left-to-right texts*, by Donald E. Knuth and Pierre MacKay, *TUGboat* **8**, 14–25, 1987.