

ИНСТИТУТ ФИЗИКИ ВЫСОКИХ ЭНЕРГИЙ

С.В. Клименко, М.В. Лисина

Л^AT_EX И ЕГО КОМАНДЫ

Draft, 1996 г.

Каталог команд L^AT_EX_α

Неалфавитные символы в этом каталоге приводятся в том порядке, в котором они находятся в шрифте пишущей машинки Кнута — для тех, кто с ним не знаком, он показан ниже — а командный символ бэкслэш в целях упорядочивания считается невидимым (за исключением того, что `\name` следует за `name`). Для указания того, что аргумент команды или командных скобок является подвижным, используется подчеркивание.

	0	1	2	3	4	5	6	7	8	9
0	`	´	^	~	¨	˘	˙	˚	˛	˜
10	·	‚	€	ı	<	>	“	”	ˆ	˜
20	˘	-	-	°	ı	j	ff	fi	fl	
30	ff	ff	ı	!	"	#	\$	%	&	'
40	()	*	+	,	-	.	/	0	1
50	2	3	4	5	6	7	8	9	:	;
60	<	=	>	?	@	A	B	C	D	E
70	F	G	H	I	J	K	L	M	N	O
80	P	Q	R	S	T	U	V	W	X	Y
90	Z	[\]	ˆ	˘	˙	a	b	c
100	d	e	f	g	h	i	j	k	l	m
110	n	o	p	q	r	s	t	u	v	w
120	x	y	z	{		}	˜	-		

`\` Командный символ, который начинает каждую команду, состоящую более чем из одного символа.

`_` Производит пробел между словами. Используется после команд, поскольку обычные пробелы после них TeX удаляет.

`\TeX` удаляет пробелы TeX удаляет пробелы

`\TeX\` удаляет пробелы TeX удаляет пробелы

Эта команда также используется в математической моде, когда надо явно задать пробел:

`$x>0,\ y\le 1,\ z\ge 2$` $x > 0, y \leq 1, z \geq 2$

Только что приведенная строка умышленно записана в такой неуклюжей форме. Элегантнее было бы написать $x > 0$, $y \leq 1$, $z \geq 2$ (начинать и заканчивать математическую моду каждый раз, когда это необходимо, и тогда достаточно обычного пробела). Прочная.

! ‘ Производит ! ‘ в абзацной или LR моде. Прочная.

\! Производит в математической моде отрицательный тонкий горизонтальный пробел ($-1/6$ квадрата — см. `\quad`). Используется очень часто! Этот мини-пробел нужен, чтобы “сдвигать” символы:

`\int\!\!\!\int_{\cal D}`

$$\frac{e^{x+y}}{x+y+1} dx dy$$
Прочная.

" Используется для задания шестнадцатиричной константы. Так, "F1 является шестнадцатиричной записью десятичного числа 17. См. команду `\char i`.

\"char Производит акцент умлаут (ö) над следующим одним символом char в LR или абзацной моде.

"e, "a, "o, "O ë, ä, ö, Ö
С буквами типа i надо использовать команду "\i (так как "i дает ï).
Прочная.

#i Указывает в определении макрокоманды или командных скобок место, где должен быть параметр (i должно лежать между 1 и 9 включительно).

\# Производит знак дизеля # в любой моде. Прочная.

$\$form$ Набирает формулу form в математической моде и текстовом стиле, чтобы получить формулу, включенную в текст. Так, $x - y > 3$ дает $x - y > 3$. В L^AT_EXе для этого имеются еще два альтернативных способа, а именно, `\(x - y > 3\)` и `\begin{math}x - y > 3\end{math}`, но трудно даже придумать разумную причину, по которой их можно использовать, поскольку оба они сложнее прямолинейных знаков доллара. Прочная.

\\$ Производит знак доллара \$ в любой моде. Прочная.

$\$\$form\$\$$ Может встретиться только в абзацной моде, где указывает, что формула form должна набираться в математической моде и выключенном стиле. Так, $\$\$x - y > 3\$\$$ дает

$$x - y > 3.$$

В L^AT_EXе для этого имеются еще два альтернативных способа, а именно, `\[x - y > 3\]` и `\begin{displaymath}x - y > 3\end{displaymath}`. Небольшое отличие между получением выключенной формулы с помощью двойных знаков доллара и двумя другими методами состоит в том, что если в

команде `\documentstyle` Вы задаете опцию `fleqn`, выключенные формулы, полученные командными скобками `displaymath` или парой команд `\[` и `\]`, не будут центрированы на странице — они будут отстоять от левого поля на расстояние, содержащееся в параметре длины `\mathindent` — в то время как выключенные формулы, полученные с использованием двойных знаков доллара, по-прежнему будут центрироваться. Так что, если Вы намереваетесь использовать в команде `\documentstyle` опцию `fleqn`, то для выключенных формул пользуйтесь командами `\[` и `\]`.

- % Специальный символ для записи комментариев. Когда Т_ЕX встречает этот символ, он прекращает чтение строки, после чего можно вставлять комментарии к программе, а также замаскировать символ конца строки (чтобы избежать лишнего пробела, например, в какой-либо макрокоманде). Все, что следует за этой командой до следующего символа конца строки — включая сам этот символ — рассматривается как комментарий (и поэтому игнорируется Т_ЕX'ом).
- \% Производит знак процента % в любой моде. Прочная.
- & Специальный символ. Используется в разных командных скобках для задания вертикального выравнивания.
- \& Производит знак амперсанда & в любой моде. Прочная.
- ' В математической моде производит знак прим, например, `x'` дает x' .
- \'char Производит акцент (ó) над следующим одним символом *char* в LR или абзацной моде, но *не* внутри командных скобок `tabbing`. Там следует использовать команду `\a'char`. Прочная.
- \(form\)) Может встретиться только в LR или абзацной моде, где набирает формулу *form* в виде включенной в текст математической формулы в текстовом стиле; эквивалентно конструкциям `\begin{math} form \end{math}` и `$form$`. Как `\(`, так и `\)` являются хрупкими.
- \) См. описание `\(`.
- *{i}{pre} Выражение, которое может встретиться только внутри *преамбулы* в командных скобках `array` или `tabular`. Оно эквивалентно i повторениям *pre*, где *i* — это любое целое положительное число, а *pre* — это любая разрешенная в преамбуле комбинация команд.
- * Обозначение возможного места расщепления формул для переноса их с одной строки на другую по знаку умножения: `$(x+y)*(z+t)$`. Если формула поместится на строке, получится обычное $(x + y)(z + t)$. Если же позиция знака умножения оказывается подходящей для переноса строки, получится $(x + y) \times$ в конце первой строки, и $(z + t)$ — в начале строки последующей. Такой знак умножения может встретиться только в математической моде.

- + Производит символ + в любой моде. В математической моде он является символом бинарного оператора.
- \+ Внутри командных скобок `tabbing` увеличивает значение параметра `left-margin-tab` (таб левого поля) на 1. См. `tabbing`. Хрупкая.
- \, Производит во всех модах тонкий горизонтальный пробел (обычно равный одной шестой квадрата). В математической моде такой пробел по умолчанию равен `3mu`. (см. `\quad`),

$$\int_a^b f(x) dx, \int_a^b f(x) \, dx \dots \dots \dots \int_a^b f(x) dx, \int_a^b f(x) dx$$
Прочная.
- В математической моде производит символ бинарного оператора −; в других модах — дефис (-). Используется для получения en-тире – (-) и em-тире — (--) в абзацной или LR моде.
- \- Вне командных скобок `tabbing` указывает на место в слове, где можно сделать перенос. Если слово переносится, то вставляется дефис. Внутри командных скобок `tabbing` уменьшает значение параметра `left-tab-margin` (таб левого поля) на 1. См. `tabbing`. Хрупкая.
- \.char Производит точечный акцент (ò) над следующим одним символом `char` в абзацной или LR моде. Прочная.
- \/ Команда курсивной поправки (вообще говоря, любая поправка, независимо от сочетания шрифтов). Речь идет о крошечном дополнительном пробеле, размеры которого зависят от сочетания букв:

$$\begin{aligned} \{\it Good\} \text{ buy} & \dots \dots \dots \text{ Good buy} \\ \{\it Good\}/ \text{ buy} & \dots \dots \dots \text{ Good buy} \\ \{\sl Good\} \text{ buy} & \dots \dots \dots \text{ Good buy} \\ \{\sl Good\}/ \text{ buy} & \dots \dots \dots \text{ Good buy} \end{aligned}$$
- : В абзацной моде — знак препинания двоеточие. В математической моде дает символ бинарного отношения : (разница в расстановке пробелов). Для двоеточия как знака препинания в математической моде этот символ использовать нельзя; там для этой цели служит команда `\colon`. См. также команду `\boldmath`.
- \> Производит средний горизонтальный пробел (обычно около двух девяток квадрата), но только в математической моде. По умолчанию его значение равно `4mu plus 2mu minus 4mu`. Заметим, что в plain TeX'e такой пробел задается командой `\>`. Прочная.
- \; Производит толстый горизонтальный пробел (обычно около пяти восемнадцати квадрата), но только в математической моде. По умолчанию равен `5mu plus 5mu`. Прочная.

< В математической моде производит символ бинарного отношения <. Нельзя использовать для угловой скобки — для этой цели используйте `\langle`. Вне математической моды дает символ !¹, за исключением области действия декларации `\tt`, где производит <.¹

\< Внутри командных скобок `tabbing` уничтожает действие одной предыдущей команды `\+`. См. `tabbing`. Хрупкая.

= Производит символ = во всех модах; в математической моде — это символ бинарного отношения.

\= Внутри командных скобок `tabbing` задает позиции табуляции. Хрупкая.

\=char Производит акцент макрон (̄) над одним следующим символом `char` в LR или абзацной моде, но не внутри командных скобок `tabbing`. Там используется `\a=char`.

`\=occam, \=assis` `\=occam, \=assis`

Чтобы провести черту над символом в *математической моде*, используйте команду `\bar`. Например, сопряженное \bar{z} комплексного числа z записывается как `\bar{z}`. Часто можно получить лучшие результаты, используя макрокоманду plain TeXa `\overline`:

`\overline{z}+\overline{k}` $\bar{z} + \bar{k}$

`\overline{\strut z}+\overline{\strut k}` $\bar{z} + \bar{k}$

Обратите внимание на действие команды `\strut` (подпорка): благодаря ей черточки располагаются на одной высоте. Макрокоманда `\overline` обязательна, если надо “надчеркнуть” сразу несколько символов.

`\overline{\mathstrut(u+v)}`=`\overline`

`{\mathstrut u}+\overline{\mathstrut v}` $\overline{(u+v)} = \bar{u} + \bar{v}$

Прочная.

> В математической моде дает символ бинарного отношения >. Этот символ нельзя использовать для угловой скобки; для этого служит `\rangle`. Вне математической моды получается символ ?², за исключением области действия декларации `\tt`, где получается >.²

\> Внутри командных скобок `tabbing` делает переход к следующей позиции табуляции. См. `tabbing`. Заметим, что в plain TeX'e эта команда используется для получения в математической моде среднего пробела. Хрупкая.

?‘ Производит ?‘ в абзацной или LR моде. Прочная.

¹В некоторых версиях русифицированного L^AT_EX'a этот символ дает знак открывающей кавычки <, который часто используется при наборе русских текстов.

²В некоторых версиях русифицированного L^AT_EX'a этот символ дает знак закрывающей кавычки >, который часто используется при наборе русских текстов.

`\@` Если стоит перед точкой, восклицательным или вопросительным знаком, то пробел, который следует за этими знаками препинания, получится такой же ширины, как и обычный пробел после знаков препинания (по умолчанию после точек, знака вопроса и восклицательного знака помещается пробел бóльшей величины, чем просто между словами). Обычно используется между заглавными буквами и знаками препинания, которые на самом деле оканчивают предложения.

Результат	Вход
Warner Bros. является компанией по производству фильмов.	Warner Bros.\ является компанией по производству фильмов.
Она находится в Великобритании. А точнее, в Шотландии.	Она находится в Великобритании\@. А точнее, в Шотландии.

Командная последовательность `_` вставляет обычный междусловный пробел. Заметим, что пробел после точки, восклицательного и вопросительного знаков оказывается больше, даже если за этими знаками следуют кавычки или скобки; в этих случаях указанные выше способы также хорошо работают.

`@{text}` Так называемое @-выражение, которое может встретиться только в преамбуле командных скобок `array`, `tabular` и `tabular*`. Помещает текст `text` в каждую строку результата. Убирает все пробелы, которые обычно вставляются в строку. Аргумент `text` является подвижным. Например, таблица значений функции x^x

x	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
x^x	0.794	0.725	0.697	0.693	0.707	0.736	0.779	0.837	0.910	1.0

была получена следующими командами:

```

$$
\arraycolsep=4pt
\begin{array}{|l|*{10}{r|}} \hline
x & 0 & 1 & 0 & 2 & 0 & 3 & 0 & 4 & 0 & 5 &
& 0 & 6 & 0 & 7 & 0 & 8 & 0 & 9 & 1 & 0 \\ \hline
%
x^x & 0 & 794 & 0 & 725 & 0 & 697 & 0 & 693 & 0 & 707 &
& 0 & 736 & 0 & 779 & 0 & 837 & 0 & 910 & 1 & 0 \\ \hline
\end{array}
$$

```

Значение параметра жесткой длины `\arraycolsep` равно половине величины горизонтального пробела, помещаемого между колонками, которые получаются командными скобками `array`. Присутствие в преамбуле вертикальных черточек дает вертикальные прямые в получаемом массиве, а выражение `*{i}{pre}` эквивалентно i копиям `pre`, где `pre` — это любая законная комбинация команд преамбулы. Встречающееся в преамбуле приведенного выше массива @-выражение `@{.}` вставляет в каждом ряду точку; кроме того действие @-выражений состоит в том, что их присутствие убирает пробелы, обычно вставляемые между колонками.

- `@article{key, field-list}` Такой вид элемента библиографической базы данных ВВТ_ЭX'a, который используется для статей или документов, опубликованных в периодических изданиях.
- `@book{key, field-list}` Вид элемента библиографической базы данных ВВТ_ЭX'a, используется для книг, имеющих имя издателя.
- `@booklet{key, field-list}` Этот вид элемента библиографической базы данных ВВТ_ЭX'a используется для работ, которые напечатаны и переплетены, но в них нет указания на то, кто их произвел.
- `@conference{key, field-list}` Такой вид элемента библиографической базы данных ВВТ_ЭX'a, который используется для статей, опубликованных в трудах какой-либо конференции. (Он в точности такой же, как `@inproceedings`.)
- `@inbook{key, field-list}` Вид элемента библиографической базы данных ВВТ_ЭX'a, используется для глав — или других частей — книг. Его можно использовать даже для выборки страниц из книги.
- `@incollection{key, field-list}` Вид элемента библиографической базы данных ВВТ_ЭX'a, используется для глав — или других частей — книг, которые имеют свое собственное название. Например, книги могут иметь главы, написанные различными авторами.
- `@inproceedings{key, field-list}` Вид элемента библиографической базы данных ВВТ_ЭX'a, используется для статей или документов, которые опубликованы в трудах некоторой конференции. (Это в точности то же самое, что и `@conference`.)
- `\@listi` Команда выполняется, когда командные скобки `list` не находятся внутри области действия других командных скобок `list`. Используется для задания начальных значений параметров длины, влияющих на внешний вид списка.
- `\@listii` Команда выполняется, когда командные скобки `list` находятся внутри одной открытой области действия командных скобок `list`. Используется для задания начальных значений параметров длины, влияющих на внешний вид списка.

- `\@listiii` Команда выполняется, когда командные скобки `list` находятся внутри двух открытых областей действия командных скобок `list`. Используется для задания начальных значений параметров длины, влияющих на внешний вид списка.
- `\@listiv` Команда выполняется, когда командные скобки `list` находятся внутри трех открытых областей действия командных скобок `list`. Используется для задания начальных значений параметров длины, влияющих на внешний вид списка.
- `\@listv` Команда выполняется, когда командные скобки `list` находятся внутри четырех открытых областей действия командных скобок `list`. Используется для задания начальных значений параметров длины, влияющих на внешний вид списка.
- `\@listvi` Команда выполняется, когда командные скобки `list` находятся внутри пяти открытых областей действия командных скобок `list`. Используется для задания начальных значений параметров длины, влияющих на внешний вид списка.
- `@manual{key, field-list}` Вид элемента библиографической базы данных ВВТ_ЭX'a, используется для руководств и других видов технической документации.
- `@masterthesis{key, field-list}` Вид элемента библиографической базы данных ВВТ_ЭX'a, используется для тезисов диссертации на соискание ученой степени магистра.
- `@misc{key, field-list}` Вид элемента библиографической базы данных ВВТ_ЭX'a, используется, когда нечто, на что Вы ссылаетесь, не входит ни в какую другую категорию.
- `@phdthesis{key, field-list}` Такой вид элемента библиографической базы данных ВВТ_ЭX'a, который используется для тезисов диссертации.
- `@proceedings{key, field-list}` Вид элемента библиографической базы данных ВВТ_ЭX'a, используется для таких трудов конференции, которые не являются одним документом.
- `@string{abbrv=text}` Встречается только внутри `bib`-файла, где делает `abbrv` сокращением для `text`, который должен быть заключен в двойные кавычки.
- `@techreport{key, field-list}` Такой вид элемента библиографической базы данных ВВТ_ЭX'a, который используется для научных и технических отчетов, выпускаемых такими учреждениями, как институты, факультеты университетов или промышленные исследовательские лаборатории.
- `@unpublished{key, field-list}` Вид элемента библиографической базы данных ВВТ_ЭX'a, используется для документов, например, напечатанных на машинке, имеющих название и автора, но нигде не издававшиеся. К этой категории можно отнести самиздат.

$$\frac{1}{1 + \frac{e^{-2\pi\sqrt{5}}}{1 + \frac{e^{-4\pi\sqrt{5}}}{1 + \frac{e^{-6\pi\sqrt{5}}}{1 + \dots}}}} = \left(\frac{\sqrt{5}}{1 + \sqrt[5]{5^{3/4} \left(\frac{\sqrt{5}-1}{2}\right)^{5/2} - 1}} - \frac{\sqrt{5}+1}{2} \right) e^{2\pi/\sqrt{5}}.$$

```


$$\frac{1}{1 + \frac{e^{-2\pi\sqrt{5}}}{1 + \frac{e^{-4\pi\sqrt{5}}}{1 + \frac{e^{-6\pi\sqrt{5}}}{1 + \dots}}}} = \left( \frac{\sqrt{5}}{1 + \sqrt[5]{5^{3/4} \left(\frac{\sqrt{5}-1}{2}\right)^{5/2} - 1}} - \frac{\sqrt{5}+1}{2} \right) e^{2\pi/\sqrt{5}}.$$


```

Рис. 1. Формула Раманяна и команды, которыми она была получена.

`\[form\]` Может встретиться только в абзацной моде, где набирает формулу *form* в математической моде, а затем помещает в виде выделенного уравнения в центре отдельной строки, если только не выбрана опция стиля документа `fleqn` — в этом случае формула печатается с отступом от левого поля, равным значению параметра длины `\mathindent`. Эквивалентно конструкции `\begin{displaymath}form\end{displaymath}`. Как `\[`, так и `\]` обе хрупкие. См. также описание `$$form$$` и пример на рис.1, стр.10.

`\[len\]` Команда конца строки. Если присутствует необязательный параметр *len*, то он задает величину вертикального пробела, вставляемого между строками. Удобно использовать для увеличения стандартного расстояния между строками, в том числе и в таблицах.

`\[*len\]` Аналогична команде `\[len\]`, за тем исключением, что место, где встретилась эта команда, никогда не попадет в конец страницы. Хрупкая.

`\]` Смотри `\[`.

`form1^form2` Делает формулу *form₂* верхним индексом формулы *form₁*. Например, `\$x^i\$` дает x^i . Если *form₂* состоит из более чем одного символа, она

должна быть заключена в фигурные скобки. Например, y^{i+1} дает y^{i+1} . Вместо символа шляпки можно использовать командную последовательность `\sp`. Так, $form_1 \sp form_2$ эквивалентно $form_1 \hat{form}_2$. См. пример на рис.1, стр.10. Прочная.

`\^char` Производит акцент циркумфлекс ($\hat{\circ}$) над следующим одним символом *char* в LR или абзацной моде. Прочная.

`form_1_form_2` Делает формулу $form_2$ нижним индексом формулы $form_1$. Например, x_i дает x_i . Если $form_2$ состоит из более чем одного символа, она должна быть заключена в фигурные скобки. Например, y_{i+1} дает y_{i+1} . Вместо символа подчеркивания можно использовать командную последовательность `\sb`. Так, $form_1 \sb form_2$ эквивалентно $form_1 \text{_} form_2$. Прочная.

`_` Для получения в любой моде символа `_`, который программисты называют *подчеркиванием* и который соединяет слова в пределах одного идентификатора.

`Plus_Grand_Commun_Diviseur` `Plus_Grand_Commune_Diviseur`
Прочная.

`'` Используется для задания восьмеричной константы. Так, `'12` является восьмеричной записью десятичного числа 10. См. команду `\char i`.

`\'char` Производит акцент гравис ($\grave{\circ}$) над следующим одним символом *char* в LR или абзацной моде, но не внутри командных скобок `tabbing`, где следует использовать `\a'char`. Прочная.

`{область}` Фигурные скобки — это специальные символы, которые используются для ограничения области действия команд или заключения обязательных аргументов команд.

`\{` Производит `{` во всех модах. В математической моде — это открывающий символ, который также можно получить командой `\lbrace`. См. рис.5 на стр.21 и рис.6 на стр.22. Прочная.

`|` В математической моде дает ординарный символ `|`, который также может быть получен командой `\vert`; однако после `\left` или `\right` дает соответствующий ограничитель. См. рис.5 на стр.21 и рис.6 на стр.22. Вне математической моды производит em-тире, кроме области действия декларации `\tt`, где дает `|`. Используется также в преамбулах командных скобок `array`, `tabular` и `tabular*` для задания вертикальной черты. См. например, таблицу, приведенную в описании `@{text}`. Прочная.

`\|` В математической моде дает ординарный символ `||`, который можно также получить командой `\Vert`; однако, после `\left` или `\right` дает соответствующий ограничитель. См. рис.5 на стр.21 и рис.6 на стр.22. Прочная.

} Смотри {.

\} Производит } во всех модах. В математической моде — это закрывающий символ, который также можно получить командой \rbrace. См. рис.5 на стр.21 и рис.6 на стр.22. Прочная.

~ Производит пробел между словами и запрещает разрыв строки.

\~char Производит акцент тильду ($\tilde{}$) над следующим одним символом *char* в LR или абзацной моде. Прочная.

11pt Возможная опция команды \documentstyle. Указывает, что при наборе документа используется размер шрифта 11 пунктов, а не принятый по умолчанию 10 пунктов. См. примеры на рис.16, стр.79 и на рис.7, стр.24.

12pt Возможная опция команды \documentstyle. Указывает, что при наборе документа используется размер шрифта 12 пунктов, а не принятый по умолчанию 10 пунктов.

a

\a'char Используется только внутри командных скобок `tabbing`, где производит акцент акут ($\acute{}$) над одним следующим символом *char*. Хрупкая.

\a=char Используется только внутри командных скобок `tabbing`, где производит акцент макрон ($\bar{}$) над одним следующим символом *char*. Хрупкая.

\a'char Используется только внутри командных скобок `tabbing`, где производит акцент гравис ($\grave{}$) над одним следующим символом *char*. Хрупкая.

\aa Команда для получения строчной скандинавской буквы а-с-кружком (\AA) только в абзацной и LR модах. Прочная.

\AA Команда для получения прописной скандинавской буквы А-с-кружком (\AA) только в абзацной и LR модах. Для получения $x = 3\text{\AA}$ введите `\$x=3\AA`. Если после \AA нужен пробел, не забудьте написать `\AA\` . Прочная.

abbrv Одна из возможных опций команды \bibliographystyle. Элементы библиографии расположены в алфавитном порядке и каждый помечен числовой меткой в квадратных скобках (например, [17]). Опция берет их имена исходя из того, что имена авторов, редакторов, названия месяцев и журналов сокращены.

{form₁ \above len form₂} Производит дробь с числителем *form₁*, знаменателем *form₂* и дробной чертой толщиной *len*. Команда примитивного Т_ЕX'а.

$$\$x+\{y+z\above 1pt v+w}\$ \dots\dots\dots x + \frac{y+z}{v+w}$$

a	<code>\abovedisplayskip + \baselineskip</code>
b	<code>\belowdisplayskip + \baselineskip</code>
c	<code>\abovedisplayshortskip + \baselineskip</code>
d	<code>\belowdisplayshortskip + \baselineskip</code>
e	$\geq 2 \text{ em}$

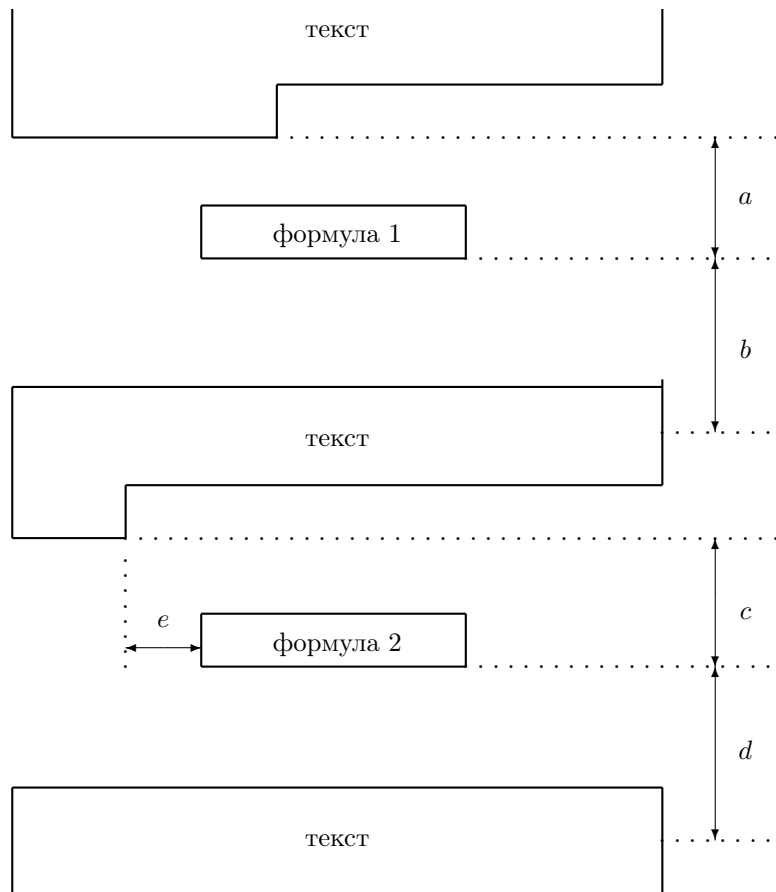


Рис. 2. "Короткие" и "длинные" выключенные формулы.

$$\$\displaystyle x+\overset{1pt}{y+z}v+w\$ \dots\dots\dots x + \frac{y+z}{v+w}$$

`\abovedisplayshortskip` Параметр эластичной длины, равный величине дополнительного пробела, который помещается над “короткой” выключенной формулой, если только в команде `\documentstyle` не выбрана опция `fleqn` (то есть, в дополнение к значению `\baselineskip`). “Короткая” формула — это такая формула, для которой расстояние ϵ на рис.2 на стр.13 больше чем $2em$, где величина em определяется размером шрифта предшествующего этой формуле абзаца. Величина этого параметра может изменяться некоторыми декларациями, изменяющими размер используемого шрифта. См. также ниже описание команды `\abovedisplayskip`. (Это команда примитивного `TeX`’а и все подробности приведены здесь только для информации. Во всяком случае это упрощение того, что происходит в действительности.) Прочная команда, перед которой не надо ставить `\protect`.

`\abovedisplayskip` Параметр эластичной длины, равный величине дополнительного пробела, помещаемого над “длинной” выключенной формулой, если только в команде `\documentstyle` не задана опция `fleqn` (то есть, в дополнение к значению `\baselineskip`). “Длинной” формулой считается такая, самая левая точка которой находится левее конца последней строчки предшествующего ей абзаца или если она правее этой точки и расстояние между этими точками меньше, чем $2em$ шрифта предыдущего абзаца. На рис.2 на стр.13 формула 1 является “длинной” в этом смысле. Величина этого параметра может быть изменена декларациями, меняющими используемый шрифт. Пробелы после формулы регулируются командами `\belowdisplayskip` и `\belowdisplayshortskip`. В формате `plain` предлагаются следующие значения этих параметров:

```
\abovedisplayskip=12pt plus 3pt minus 9pt
\abovedisplayshortskip=0pt plus 3pt
\belowdisplayskip=\abovedisplayskip
\belowdisplayshortskip=7pt plus 3pt minus 3pt
```

(Это команда примитивного `TeX`’а и все подробности приведены здесь только для информации. Во всяком случае, это упрощение того, что происходит на самом деле.). Прочная команда, перед которой не надо ставить `\protect`.

abstract Командные скобки, используемые в стилях документов `article` и `report` для получения аннотаций. Их надо помещать во входном файле внутри командных скобок `document` после команды `\maketitle`. Пример см. на рис.7, стр.24. По умолчанию в первой строке аннотации делается абзацный отступ, но его можно удалить командой `\noindent`.

`\acute` Команда для получения акцента в математической моде. Так, `\acute x` дает \acute{x} . Прочная.

`\addcontentsline{ext}{sec-unit}{entry}` Записывает элемент в файл, заданный параметром *ext*, который может быть равен `lof`, `lot` или `toc`. Аргумент *sec-unit* управляет тем, что записывается; если *ext* равен `lof`, то *sec-unit* должен быть равен `figure`, если *ext* равен `lot`, то *sec-unit* должен быть равен `table`, а если *ext* равен `toc`, то *sec-unit* должен быть `part`, `chapter` (но не в стиле документа `article`), `section`, `subsection`, `paragraph` или `subparagraph`. (Заметим, что *sec-unit* равен, например, `section`, а не `\section`). Аргумент *entry*, который является подвижным, — это реальный текст, который появится в списке рисунков, списке таблиц или оглавлении. Если Вы хотите, чтобы рисунки, таблицы или пункты оглавления были пронумерованы, то вместо просто текста в *entry* следует использовать команду `\numberline`. Обратите внимание на то, как она помещается относительно текста:

```
\addcontentsline{lof}{figure}
  {\protect\numberline{12.3}
   {Полевые позиции крикета.}}
```

Если *entry* — или соответствующий аргумент в `\numberline` — слишком длинный, то Вы получите сообщение об ошибке, которое говорит о превышении размера буфера.

address Поле элемента в библиографической базе данных ВВТ_EX'a, которое содержит адрес издателя написанной работы. Если издатель не очень известный, здесь можно поместить о нем дополнительную информацию.

`\address{text}` Декларация, которая может встретиться только в стиле документа `letter`. Используется для получения адреса отправителя. Внутри аргумента *text* для начала новой строки можно использовать одну или несколько команд `\\`. См. `letter`

`\addtocontents{ext}{text}` Записывает аргумент *text* в файл, указанный в аргументе *ext*, который может быть равен `lof`, `lot` или `toc`. Подвижный аргумент *text* может содержать как обычный текст, так и команды форматирования. Если *text* слишком длинный, то Вы получите сообщение об ошибке, которое говорит о превышении размера буфера Л_AT_EX'a.

`\addtocounter{ctr}{i}` Глобальная декларация, присваивает счетчику *ctr* результат сложения *i* и текущего значения *ctr*. Например, вставим здесь сноску³, а затем изменим значение счетчика `footnote`:

```
\addtocounter{footnote}{-1}
```

Обратите внимание на номер следующей сноски — несмотря на то, что мы уменьшили счетчик `footnote` на 1, номер сноски не изменился³. Хрупящая.

³ее метка определяется счетчиком `footnote`

³Не забывайте, что счетчики увеличиваются на 1 перед их использованием.

`\addtolength{cmd}{len}` Присваивает команде длины *cmd*, которая должна начинаться с бэкслэша, результат сложения *len* с ее текущим значением. Пример ее использования можно посмотреть на рис.3 на стр.17, а результат — на рис. 4 на стр.18. Прочная.

`\addvspace{len}` Обычно добавляет вертикальный пробел размера *len*. Две такие команды подряд: `\addvspace{len1}\addvspace{len2}`, добавляют только один пробел *len_i*, где *len_i* — это максимум из *len₁* и *len₂*. Используется в определениях многих командных скобок, в результате чего пробелы, окружающие вложенные командные скобки, получаются такими же, как пробелы над и под одинарными командными скобками. Хрупкая.

`\ae` Команда, доступная только в абзацной и LR модах для получения строчной латинской и скандинавской гласной лигатуры (æ) (для œ есть `\oe`).

```
ros\ae, \oe d\`eme .....rosæ, œdème
```

Теперь вы понимаете, почему пробел, следующий непосредственно за макрокомандой, на самом деле пробелом не является? Прочная.

`\AE` Команда, доступная только в абзацной и LR модах для получения прописной латинской и скандинавской гласной лигатуры (Æ). Последовательность `\AE SOPE` дает `ÆSOPE`. Обратите внимание, что пробел после `\AE` игнорируется. Прочная.

`\aleph` Производит ординарный символ ℵ, но только в математической моде. Прочная.

`alph` Один из параметров глобальной декларации `\pagenumbering`. Если он используется, номера страниц печатаются строчными латинскими буквами.

`\alph{ctr}` Значение счетчика *ctr* — которое должно быть целым положительным числом, меньшим 27 — выводится в виде строчной буквы. Например, на этой странице команда `\alph{page}` дает p. Прочная.

`Alph` Один из параметров глобальной декларации `\pagenumbering`. Если он используется, номера страниц печатаются прописными латинскими буквами. См. описание команды `\pagenumbering`.

`\Alph{ctr}` Значение счетчика *ctr* — которое должно быть целым положительным числом, меньшим 27 — выводится в виде прописной буквы. Например, на этой странице команда `\Alph{page}` дает P. Прочная.

`alpha` Одна из возможных опций команды `\bibliographystyle`. Элементы библиографии располагаются по алфавиту и каждый помечается меткой, составленной из трех первых букв имени автора и двух последних цифр года издания. Например, [Lak76].

`\alpha` Производит ординарный символ α, но только в математической моде. Прочная.


```

\begin{figure}[p]
{\parskip=-1.5pt
{\large \setlength{\parindent}{1in}
Цап-царап \addtolength{\parindent}{1em}\par
сказал мыш- \addtolength{\parindent}{4em}\par
ке: ‘‘Вот \addtolength{\parindent}{.5em}\par
какие \addtolength{\parindent}{0.5em}\par
делиш-\par} {\normalsize \setlength{\parindent}{1.9in}
ки, мы \addtolength{\parindent}{-1.5em}\par
пойдем \addtolength{\parindent}{-.5em}\par
с то- \addtolength{\parindent}{-1em}\par
бой в суд, \addtolength{\parindent}{-1em}\par
{\it я} тебя \addtolength{\parindent}{-.5em}\par
засу-\par} {\small \setlength{\parindent}{1.35in}
жу. И не \addtolength{\parindent}{1.5em}\par
смей \addtolength{\parindent}{1.5em}\par
. . .
потому\par} {\footnotesize \setlength{\parindent}{2.45in}
что все \addtolength{\parindent}{3em}\par
утро \addtolength{\parindent}{-2em}\par
я без дела \addtolength{\parindent}{-1.5em}\par
сiju.’’\par} {\scriptsize \setlength{\parindent}{2.1in}
И на это \addtolength{\parindent}{-1.75em}\par
. . .
ведут.’’ \addtolength{\parindent}{.5em}\par
--- Я и суд,\par}{\tiny \setlength{\parindent}{1.75in}
я и след- \addtolength{\parindent}{-1.5em}\par
ствие, --- \addtolength{\parindent}{-1.5em}\par
Цап-Царап \addtolength{\parindent}{-1em}\par
. . .
тебе \addtolength{\parindent}{-1.5em}\par
и ко- \addtolength{\parindent}{1.25em}\par
нец \addtolength{\parindent}{1em}\par
.~ \addtolength{\parindent}{.75em}\par~\par}}
\caption{Хвост мыши из ‘‘Алисы в стране чудес’’.}
\label{taleout}
\end{figure}

```

Рис. 3. Хвост мыши из ‘‘Алисы в стране чудес’’ — команды для получения.

Цап-царап
 сказал мыш-
 ке: “Вот
 какие
 делиш-
 ки, мы
 пойдем
 с то-
 бой в суд,
 я тебя
 засу-
 жу. И не
 смей
 отпи-
 раться,
 мы дол-
 жны рас-
 кви-
 таться,
 потому
 что все
 утро
 я без дела
 сижу.”
 И на это
 нахалу
 мышка так
 отвечала:
 “Без
 суда и
 без след-
 ствия,
 сударь,
 дел не
 ведут.”
 — Я и суд,
 я и след-
 ствие, —
 Цап-Царап
 ей от-
 ветст-
 вует. —
 Присужу
 тебя к
 смерти
 я, тут
 тебе
 и ко-
 нец

Рис. 4. Хвост мыши из “Алисы в стране чудес”.

- `\amalg` Дает символ бинарного оператора \amalg . Только в математической моде.
- `$x\amalg y$` $x \amalg y$
- Не путайте с большим оператором `\coprod`:
- `\amalg` `\coprod` `$\displaystyle\coprod$` \amalg \amalg \amalg
- Прочная.
- `and` В полях `author` и `editor` элемента библиографической базы данных ВивТ_EX'a разделяет имена нескольких авторов или редакторов.
- `and_others` Может прервать поле автора или редактора в элементе библиографической базы данных ВивТ_EX'a. В результирующем документе появляется в виде *'et al.'*
- `\and` Может встретиться только внутри аргумента декларации `\author`. Используется для разделения имен нескольких авторов.
- `\angle` Производит ординарный символ \angle , но только в математической моде.
- угол `$\angle AMB$` угол $\angle AMB$
- Иногда угол обозначают по-другому:
- угол `\widehat{ABM}` угол \widehat{ABM}
- Прочная.
- `annotate` Имя поля элемента в библиографической базе данных ВивТ_EX'a. Используется для аннотаций, которые не появятся в получаемой библиографии.
- `\appendix` Декларация, которая изменяет способ нумерации единиц рубрикации. В стилях документа `book` и `report` главы начинают "нумероваться" по алфавиту, а в стиле документа `article` по алфавиту начинают "нумероваться" секции. На нумерацию частей не действует. Аргументов не имеет.
- `\approx` Производит символ бинарного отношения \approx , но только в математической моде. А может быть, Вам потребуется символ `\simeq` (\simeq) или `\cong` (\cong)? Прочная.
- `arabic` Один из параметров глобальной декларации `\pagenumbering`. Если он используется, номера страниц печатаются арабскими цифрами. См. описание команды `\pagenumbering`.
- `\arabic{ctr}` Счетчик `ctr` выводится в виде арабского числа. Например, на этой странице команда `\arabic{page}` дает 19. Прочная.
- `\arccos` Производит обозначение элементарной функции (\arccos), но только в математической моде. Нижние и верхние индексы никогда не появляются в виде пределов, добавленных к получаемому символу. Прочная.

`\arcsin` Производит обозначение элементарной функции (\arcsin), но только в математической моде. Нижние и верхние индексы никогда не появляются в виде пределов, добавленных к получаемому символу. Прочная.

`\arctan` Производит обозначение элементарной функции (\arctan), но только в математической моде. Нижние и верхние индексы никогда не появляются в виде пределов, добавленных к получаемому символу. Можно определить и такую макрокоманду `\arctg`:

```
\def\arctg{\mathop{\rm arctg}\nolimits}
```

Прочная.

`\arg` Производит обозначение элементарной функции (\arg), но только в математической моде:

```
 $\arg(zz') = \arg z + \arg z'$  .....  $\arg(zz') = \arg z + \arg z'$ 
```

Нижние и верхние индексы никогда не появляются в виде пределов, добавленных к получаемому символу. Прочная.

`array` Командные скобки, которые можно использовать только в математической моде, где они производят двумерное пространственное расположение преимущественно математических символов. Вот простой пример:

Φ	Ψ	B	Γ	Γ	S	T	A	Z
--------	--------	-----	----------	----------	-----	-----	-----	-----

```

$$
\begin{array}{ccc}
\bf \Phi & \Psi & B \\
\mit\Gamma & \Gamma & S \\
T & A & Z
\end{array}
$$

```

Обратите внимание — внутри элементов массива можно использовать декларации, но их область действия — если только они неглобальные — ограничивается тем элементом, в котором они находятся. Другими словами, область действия декларации оканчивается следующими символами или командами: `&`, `\` или `\end{array}`.

`\arraycolsep` Параметр жесткой длины, равный половине величины горизонтального пробела, который оставляется между двумя колонками, получаемыми командными скобками `array`. Пример см. в описании `@{text}`. Прочная команда, поэтому перед ней не надо помещать `\protect`.

`\arrayrulewidth` Параметр жесткой длины, равный толщине каждой прямой, которая получается командными скобками `array`. Прочная команда, поэтому перед ней не надо помещать `\protect`.

`\arraystretch` Значением команды является вещественное число. Управляет величиной вертикального пробела между рядами, получаемыми командными

<code>\left(</code>	<code>\left\{</code>	<code>\left[</code>
<code>\tempest</code>	<code>\tempest</code>	<code>\tempest</code>
<code>\right)</code>	<code>\right\}</code>	<code>\right]</code>
<code>\left\lgroup</code>	<code>\left\lceil</code>	<code>\left\lfloor</code>
<code>\tempest</code>	<code>\tempest</code>	<code>\tempest</code>
<code>\right\rgroup</code>	<code>\right\rceil</code>	<code>\right\rfloor</code>
<code>\left\uparrow</code>	<code>\left\updownarrow</code>	<code>\left\downarrow</code>
<code>\tempest</code>	<code>\tempest</code>	<code>\tempest</code>
<code>\right\Uparrow</code>	<code>\right\Updownarrow</code>	<code>\right\Downarrow</code>
<code>\left\lmoustache</code>	<code>\left\arrowvert</code>	<code>\left\bracevert</code>
<code>\tempest</code>	<code>\tempest</code>	<code>\tempest</code>
<code>\right\rmoustache</code>	<code>\right\Arrowvert</code>	<code>\right\bracevert</code>
<code>\left\langle</code>	<code>\left\vert</code>	<code>\left\backslash</code>
<code>\tempest</code>	<code>\tempest</code>	<code>\tempest</code>
<code>\right\rangle</code>	<code>\right\Vert</code>	<code>\right/</code>
	<code>\def\tempest%</code>	
	<code>{\begin{array}{ccc}</code>	
	<code>2 & 7 & 6 \\\</code>	
	<code>9 & 5 & 1 \\\</code>	
	<code>4 & 3 & 8</code>	
	<code>\end{array}}</code>	

Рис. 5. Ограничители в Т_EX'e (примеры кодировки).

скобками `array`, `tabular` или `tabular*`, путем умножения на ширину, принятую по умолчанию. Значение по умолчанию равно 1, но если заменить его на 1.25, скажем, декларацией `\renewcommand` или `\def`, то расстояние между напечатанными рядами будет больше в 1.25 раза.

`\arrowvert` Доступна только в математической моде, где перед ней должна стоять команда `\left` или `\right` и где она, как это показано на рис.5 стр.21 и рис.6 стр.22, производит ограничитель.

`\Arrowvert` Доступна только в математической моде, где перед ней должна стоять команда `\left` или `\right` и где она, как это показано на рис.5 стр.21

$$\begin{array}{ccc}
\left(\begin{array}{ccc} 2 & 7 & 6 \\ 9 & 5 & 1 \\ 4 & 3 & 8 \end{array} \right) & \left\{ \begin{array}{ccc} 2 & 7 & 6 \\ 9 & 5 & 1 \\ 4 & 3 & 8 \end{array} \right\} & \left[\begin{array}{ccc} 2 & 7 & 6 \\ 9 & 5 & 1 \\ 4 & 3 & 8 \end{array} \right] \\
\\
\left(\begin{array}{ccc} 2 & 7 & 6 \\ 9 & 5 & 1 \\ 4 & 3 & 8 \end{array} \right) & \left[\begin{array}{ccc} 2 & 7 & 6 \\ 9 & 5 & 1 \\ 4 & 3 & 8 \end{array} \right] & \left| \begin{array}{ccc} 2 & 7 & 6 \\ 9 & 5 & 1 \\ 4 & 3 & 8 \end{array} \right| \\
\\
\uparrow \begin{array}{ccc} 2 & 7 & 6 \\ 9 & 5 & 1 \\ 4 & 3 & 8 \end{array} \uparrow & \updownarrow \begin{array}{ccc} 2 & 7 & 6 \\ 9 & 5 & 1 \\ 4 & 3 & 8 \end{array} \updownarrow & \left| \begin{array}{ccc} 2 & 7 & 6 \\ 9 & 5 & 1 \\ 4 & 3 & 8 \end{array} \right| \downarrow\downarrow \\
\\
\left(\begin{array}{ccc} 2 & 7 & 6 \\ 9 & 5 & 1 \\ 4 & 3 & 8 \end{array} \right) & \left| \begin{array}{ccc} 2 & 7 & 6 \\ 9 & 5 & 1 \\ 4 & 3 & 8 \end{array} \right| & \left| \begin{array}{ccc} 2 & 7 & 6 \\ 9 & 5 & 1 \\ 4 & 3 & 8 \end{array} \right| \\
\\
\left\langle \begin{array}{ccc} 2 & 7 & 6 \\ 9 & 5 & 1 \\ 4 & 3 & 8 \end{array} \right\rangle & \left| \begin{array}{ccc} 2 & 7 & 6 \\ 9 & 5 & 1 \\ 4 & 3 & 8 \end{array} \right| & \left\{ \begin{array}{ccc} 2 & 7 & 6 \\ 9 & 5 & 1 \\ 4 & 3 & 8 \end{array} \right\}
\end{array}$$

Рис. 6. Ограничители в \TeX 'е (результаты)

и рис.6 стр.22, производит ограничитель.

article Один из стилей документов, которые могут быть аргументом *doc-style* команды `\documentstyle`. По умолчанию действует односторонняя “печать”, то есть, все страницы являются (концептуально) правосторонними и нечетными. Это означает, что на физическом листе бумаги заголовок, тело и основание страниц относительно левой и верхней границ листа размещаются одинаково и для четных, и для нечетных страниц. По умолчанию принят стиль страниц `plain`, то есть, заголовок пустой, а номер страницы находится в центре основания. По умолчанию также действует декларация `\raggedbottom`, то есть, вертикальные пробелы между абзацами остаются постоянными и поэтому высота текста на разных страницах может слегка изменяться.

Если для библиографии вы собираетесь использовать $\text{Vi}\TeX$, надо поместить команду `\bibliographystyle` вскоре после начала командных скобок `document`, а поблизости от конца этих командных скобок поместить команду `\bibliography`. Таким образом, Ваш входной файл будет выглядеть примерно так, как показано на рис.7, стр.24. Если Вы получаете библиографию

командными скобками `thebibliography`, то эти командные скобки должны находиться непосредственно перед командой `\end{document}` (если надо, чтобы библиография была расположена в конце выходного документа).

Заголовок получается командой `\maketitle`, которая, если вообще встречается во входном файле, должна помещаться внутри командных скобок `document`. Если используется команда `\maketitle`, то она должна стоять после обеих деклараций `\title` и `\author`. Ей может также предшествовать декларация `\date`, хотя это не обязательно; если `\date` отсутствует, используется текущая дата. Заметим, что по умолчанию заголовок появляется *не* на отдельной странице. Если нужно, чтобы заголовок, автор и дата были вынесены на отдельную страницу, надо в команде `\documentstyle` задать опцию `titlepage`. (Это также приведет к тому, что аннотация — если она присутствует — тоже напечатается на отдельной странице.)

Сам заголовок является аргументом декларации `\title`. Для того, чтобы задать разбиение строк в заголовке, надо в аргументе декларации `\title` использовать команду `\\`. Внутри аргумента декларации `\title` также могут появиться одна или несколько команд `\thanks`. Они делают сноски.

Автор или авторы статьи задаются как аргумент декларации `\author`. Если у статьи несколько авторов, их имена должны разделяться командой `\and`. Так же, как и в декларации `\title`, можно использовать команду `\\` для принудительного разрыва строки и команду `\thanks` для получения сноски.

Необязательная декларация `\date` может использоваться для получения выбранной Вами даты (или, в действительности, любой другой информации). Если эта декларация опущена, в качестве даты используется дата обработки ЛАТ_EX'ом входного файла. Так же, как в случае декларации `\title`, можно для окончания строки использовать команду `\\`, а для получения сносок — команду `\thanks`.

Если Ваш документ должен иметь аннотацию, заключите ее текст в командные скобки `abstract`. Они должны находиться внутри командных скобок `document`; обычно наилучшим является место сразу после команды `\maketitle`.

В стиле `article` имеются следующие команды рубрикации: `\part`, `\section`, `\subsection`, `\subsubsection`, `\paragraph` и `\subparagraph`. С каждой командой рубрикации связан номер уровня — как показано на рис.8, стр.24 — который используется, чтобы определить, будет ли заголовок, связанный с командой рубрикации, помещаться в оглавлении, если оно есть, и должен или нет этот заголовок, появляясь в выходном документе, нумероваться.⁴ Имеются два счетчика, которые управляют такими вещами, а именно, `tocdepth` и `secnumdepth`. В оглавление (если оно есть) помещаются заголовки единиц рубрикации, номера уровней которых меньше или равны значению счетчика `tocdepth`, а нумероваться будут все заголовки единиц

⁴Статьи обычно не имеют оглавления, но его можно получить, используя команду `\tableofcontents`.

```

\documentstyle[11pt,own,russian]{article}
\title{Каталог команд \LaTeX a}
\author{С.Клименко\and М.Лисина}
\begin{document}
\bibliographystyle{plain}
\maketitle
\begin{abstract}
\noindent
В этой статье приводится каталог команд \LaTeX a
. . .
\end{abstract}
\noindent
Текст статьи.
\nocite{*}
\bibliography{pom}
\end{document}

```

Рис. 7. Пример использования стиля документа `article`

команда рубрикации	номер уровня	в оглав- лении	нумеру- ется
<code>\part</code>	0	да	да
<code>\section</code>	1	да	да
<code>\subsection</code>	2	да	да
<code>\subsubsection</code>	3	да	да
<code>\paragraph</code>	4	нет	нет
<code>\subparagraph</code>	5	нет	нет

Рис. 8. Команды рубрикации стиля документа `article`.

рубрикации, номера уровней которых меньше или равны значению счетчика `secnumdepth`. По умолчанию оба эти значения равны 3. Рис.8, стр.24 показывает, какие команды рубрикации нумеруются и какие заголовки заносятся в оглавление с учетом этих принятых по умолчанию значений.

С каждой командой рубрикации связан счетчик, имя которого совпадает с именем соответствующей ему команды рубрикации, но без начального символа бэкслэш. Например, счетчик, связанный с командой `\section`, называется `section`. В начале статьи все счетчики, связанные с командами рубрикации, установлены в нуль. Соответствующие команды рубрикации увеличивают эти счетчики *перед* тем, как формируется номер единицы рубрикации.

Если используются командные скобки `equation` и `eqnarray`, то уравнения

нумеруются во всем документе последовательно, начиная с 1. Если в документе встречаются рисунки, они нумеруются последовательно во всем документе, начиная с 1. Если в документе встречаются таблицы, они нумеруются последовательно во всем документе, начиная с 1. Заметим, что и уровень, и рисунок, и таблица могут одновременно иметь номер 1.

`\ast` Производит символ бинарного оператора $*$, но только в математической моде. Клавиша $*$ дает тот же символ. Прочная.

`\asymp` Производит символ бинарного отношения \asymp , но только в математической моде. Прочная.

`{form1 \atop form2}` Располагает символы в виде дроби, но без дробной черты: *form₁* находится над *form₂*:

$$\begin{aligned} & \text{\$x+{\u\atop v+w}\$} \dots\dots\dots x + \frac{u}{v+w} \\ & \text{\$ \displaystyle x+{\u\atop v+w}\$} \dots\dots\dots x + \frac{u}{v+w} \\ & \text{\$ \displaystyle S = \sum_{\scriptstyle 1 < i < n \atop \scriptstyle J, K \subset X} A_i B_{JK} } \dots\dots\dots S = \sum_{\substack{1 < i < n \\ J, K \subset X}} A_i B_{JK} \end{aligned}$$

В последнем примере обратите внимание на `\scriptstyle` в числителе и знаменателе. Это прием, который не дает индексам стать слишком маленькими. Иначе они окажутся в стиле `\scriptscriptstyle` (5 пунктов вместо 7), поскольку `\atop` сама уже в индексе. Команда примитивного TeX'a.

`{form1 \atopwithdelims delim1 delim2 form2}` Производит расположение символов типа дроби, такое, что *form₁* находится над *form₂*, дробной черты нет, а вся конструкция ограничена слева *delim₁*, а справа — *delim₂*:

$$\begin{aligned} & \text{\def\toto{\atopwithdelims<>}} \\ & \text{\$q+{x+u \toto x+vw}\$} \dots\dots\dots q + \left\langle \frac{x+u}{x+vw} \right\rangle \\ & \text{\$ \displaystyle q+{x+u \toto x+vw}\$} \dots\dots\dots q + \left\langle \frac{x+u}{x+vw} \right\rangle \end{aligned}$$

Команда примитивного TeX'a, которая используется в определении команд `\brace`, `\brack` и `\choose`.

`author` Имя поля элемента библиографической базы данных BibTeX'a, которое содержит имя или имена авторов работы.

`\author{text}` Декларация, объявляющая автора или авторов документа. Если *text* состоит из нескольких имен авторов, то они разделяются командами `\and`. Внутри *text* можно использовать команду `\` для разбиения строк, а также одну или несколько команд `\thanks`, которые делают сноски с маркерами нулевой ширины. Если команда `\thanks` не оканчивает строку, то после нее, чтобы вставить пробел, надо поставить команду `_`.

aux Расширение имени для вспомогательного файла, который содержит разную информацию, используемую для перекрестных ссылок.

b

`\b $_char$` или `\b{ $char$ }` Производит подчеркивающий акцент (\acute{o}) под следующим одним символом $char$ в LR или абзацной моде.

`\b x, \b o, \b k` $\acute{x}, \acute{o}, \acute{k}$

Сравните с командой `\underline`, которая выполняет подчеркивание и в математической моде. Последняя, ко всему прочему, может подчеркнуть более одного символа:

`$$\underline{PAO}$$, \underline{PAO}` $\underline{PAO}, \underline{PAO}$

Прочная.

`\backslash` Производит ординарный символ `\` (без пробелов слева и справа). Только в математической моде:

`$$H\backslash G/K$$` $H\backslash G/K$

Не путать эту команду с командой `\setminus` (вычитание множеств), которая задает бинарную операцию:

множество `$$H\setminus G$$` множество $H \setminus G$

класс `$$H\backslash G$$` класс $H \backslash G$

Чтобы напечатать `'\'` с текстом шрифта `\tt`, введите `\char'\'`.

Если перед командой `\backslash` стоит команда `\left` или `\right`, то получается ограничитель (см. рис.5 на стр.21 и рис.6 на стр.22). Прочная.

`\bar` Математический акцент надчеркивания (\bar{x}). Только в математической моде. Проводит черту над буквой, следующей непосредственно за командой: например, мнимая часть комплексного числа z задается в виде `$$\bar{z}$$`. В математической моде также можно использовать команду `\overline`:

`$$\bar{z}+\bar{k}$$` $\bar{z} + \bar{k}$

`$$\overline{z}+\overline{k}$$` $\bar{z} + \bar{k}$

`$$\overline{\mathstrut z}+\overline{\mathstrut k}$$` $\bar{z} + \bar{k}$

Обратите внимание на действие команды `\mathstrut` (невидимая подпорка величиной с круглую скобку, которая позволяет размещать черточки на одной высоте). Команда `\overline` может надчеркнуть более одного символа. Прочная.

`\baselineskip` Параметр эластичной длины, значение которого, умноженное на значение команды `\baselinestretch`, равно естественному расстоянию между базовыми линиями каждых двух последовательных строк абзаца. (На самом деле это команда примитивного Т_ЕX’а.) Во всем абзаце используется то значение этого параметра, которое он имеет в конце этого абзаца. При использовании команды внутри группы, не забудьте перед закрытием группы написать `\par`:

```
{\baselineskip=14pt...\par},
```

иначе интервал не изменится. Хотя это значение может быть эластичным, Кнут говорит, что если его сделать жестким, то результат будет выглядеть лучше. Но даже если значение этого параметра сделать жестким, расстояние между двумя базовыми линиями последовательных строк абзаца может измениться, если одна из этих строк содержит большой символ. Заметим, что значение `\baselineskip` изменяется декларациями, меняющими размер шрифта — такими, как `\tiny` и `\large`. (Подробности включены только для информации. Ответственность за значение `\baselineskip` несет разработчик стиля документа. Если же Вам на самом деле надо изменить расстояние между базовыми линиями — например, чтобы получить машинописный документ в два интервала — то меняйте значение `\baselinestretch`.) Команда прочная и перед ней не надо ставить `\protect`.

`\baselinestretch` Значением команды является вещественное число, по умолчанию равное 1. Реальное расстояние между базовыми линиями двух последовательных строк абзаца равно значению `\baselineskip`, умноженному на значение `\baselinestretch`.

`\batchmode` Один из примитивов, которые управляют уровнем взаимодействия во время работы Т_ЕX’а. Если первой командой входного файла является эта команда (даже перед `\documentstyle`), то на терминал не выводится никакой информации, а Т_ЕX продолжает работу даже после того, как он встретил ошибку. Протокол работы записывается в файл с расширением `.log`.

`.bbl` Расширение имени файла, который пишется или перезаписывается при работе В_ВТ_ЕX’а и содержит команды форматирования, которые затем используются для получения библиографии в выходном документе. Если Вам не нравится что-либо в результирующем документе, этот файл можно редактировать.

`\begin{env}` Начинает командные скобки *env*. Эти командные скобки должны заканчиваться командой `\end{env}`. Хрупкая.

`\belowdisplayshortskip` Параметр эластичной длины, равный величине дополнительной вертикального пробела, помещаемого под “короткой” выключенной формулой, если в команде `\documentstyle` не выбрана опция `fleqn`. Что такое “короткая” формула, объясняется в описании команды `\abovedisplayshortskip`,

Результат	Команда	Название
Жили-были старик ...	<code>{\bf Жили-были старик ...}</code>	жирный
<i>Жили-были старик ...</i>	<code>{\it Жили-были старик ...}</code>	курсив
Жили-были старик ...	<code>{\rm Жили-были старик ...}</code>	романский
ЖИЛИ-БЫЛИ СТАРИК ...	<code>{\sc ЖИЛИ-БЫЛИ СТАРИК ...}</code>	капитель
Жили-были старик ...	<code>{\sf Жили-были старик ...}</code>	рубленный
<i>Жили-были старик ...</i>	<code>{\sl Жили-были старик ...}</code>	наклонный
Жили-были старик ...	<code>{\tt Жили-были старик ...}</code>	машинописный

Результат	Вход
Я ненавижу вас.	<code>{\bf Я {\em ненавижу\}/} вас.}</code>
<i>Я ненавижу вас.</i>	<code>{\it Я\ / {\em ненавижу} вас.}</code>
Я ненавижу вас.	<code>{\rm Я {\em ненавижу\}/} вас.}</code>
Я ненавижу ВАС.	<code>{\sc Я {\em ненавижу\}/} вас.}</code>
Я ненавижу вас.	<code>{\sf Я {\em ненавижу\}/} вас.}</code>
<i>Я ненавижу вас.</i>	<code>{\sl Я\ / {\em ненавижу} вас.}</code>
Я ненавижу вас.	<code>{\tt Я {\em ненавижу\}/} вас.}</code>

Рис. 9. Декларации, изменяющие шрифт.

а также на рис.2, стр.13. Команда примитивного Т_ЕX’а, которая является прочной, следовательно, перед ней не надо ставить `\protect`.

`\belowdisplayskip` Параметр эластичной длины, равный величине дополнительного пробела, помещаемого под “длинной” выключенной формулой, если в команде `\documentstyle` не выбрана опция `fleqn`. Что такое “длинная” формула объясняется в описании команды `\abovedisplayshortskip` и на рис.2, стр.13. Команда примитивного Т_ЕX’а, которая является прочной, следовательно, перед ней не надо ставить `\protect`.

`\beta` Производит ординарный символ β , но только в математической моде. Прочная.

`\bf` Декларация, которая меняет шрифт на **жирный** — см. рис.9 на стр.28. Команда может быть *локальной*, т.е. ограниченной одной группой:

это `{\bf замечание}` интересно это `замечание` интересно или *глобальной*, если писать `\bf` В этом случае жирным шрифтом будет выделяться весь последующий текст до тех пор, пока не встретится какой-либо другой переключатель шрифтов (например, `\rm`, `\it` или `\sl`). Прочная.

`.bib` Расширение имени файла, содержащего один или несколько элементов библиографической базы данных В_ИT_ЕX’а.

`\bibindent` Параметр жесткой длины — доступный, только если в команде `\documentstyle` выбрана опция `openbib` — равный величине отступа во второй и всех последующих строках элементов, получаемых командными скобками `thebibliography`. Прочная команда, поэтому перед ней не надо ставить `\protect`.

`\bibitem[text]{key}` Команда, которая может встретиться только внутри командных скобок `thebibliography`. Аргумент *key* состоит из букв, цифр и знаков препинания, кроме запятой. Он используется в команде `\cite` для ссылок на работу в библиографии. Если отсутствует аргумент *text*, то \LaTeX генерирует числовую метку, заключенную в квадратные скобки, например, [31]. В противном случае (если *text* присутствует) меткой является [*text*]. Заметим, что *text* — если присутствует — является подвижным аргументом. Хрупкая.

`\bibliography{bib-file-list}` Когда во входном файле встречается `\cite{key}`, \TeX просматривает файлы, перечисленные в списке файлов *bib-file-list*, чтобы найти там элемент с ключом *key*. Если такой элемент найден, то библиография будет содержать элемент, полученный из данных, которые содержатся в `bib`-файле. В *bib-file-list* помещаются только основные имена файлов, причем они разделяются запятыми.

`\bibliographystyle{bib-style}` Используется вместе с \TeX ’ом и определяет, как будет выглядеть библиография в выходном результате. Стандартными опциями для аргумента *bib-style* являются `abbrv`, `alpha`, `plain` и `unsrt`. На Вашем компьютере могут быть и другие опции, например, `agsm`, `dcu` или `kluwer`, которые используются вместе с опцией `harvard` команды `\documentstyle`.

`\big` Команда `plain` \TeX а. Слегка увеличивает растяжимый символ. Только в математической моде. Синтаксис: `\big` *растяжимый символ* (круглая скобка, квадратная скобка, фигурная скобка, вертикальная черта и т.д.) Может быть в трех различных вариантах, а именно: `\bigl`, `\bigr` и `\bigs`. Команда `\bigl` используется с открывающим ограничителем, а команда `\bigr` — с закрывающим (‘l’ для левой скобки и ‘r’ для правой):

$$\text{\$f\bigl(x+(y+z)\bigr)\$} \dots\dots\dots f(x + (y + z))$$

Эти нюансы весьма существенны. Обращайте на них внимание, иначе пробелы расставятся неправильно. Вариант команды `\bigs` слегка увеличивает ограничитель и немного промежутков вокруг него:

$$\text{\$(x\in A(n)|y\in B(m))\$} \dots\dots\dots (x \in A(n)|y \in B(m))$$

$$\text{\$\big(x\in A(n)\big|y\in B(m)\big)\$} \dots\dots\dots (x \in A(n)|y \in B(m))$$

$$\text{\$\bigl(x\in A(n)\bigr|y\in B(m)\bigr)\$} \dots\dots\dots (x \in A(n) | y \in B(m))$$

Итак, рекомендации предельно просты: `\big` — для увеличения растяжимого символа без модификации окружающих его интервалов, `\bigl` для открывающего ограничителя, `\bigr` для закрывающего и `\bigs`, когда нужно увеличить интервал с обеих сторон от символа.

`\bigoplus` Производит символ большого оператора \bigoplus или \bigoplus , но только в математической моде:

$$\text{\$}\displaystyle E=\text{\bigoplus}_{i\in I}E_i\text{\$} \dots\dots\dots E = \bigoplus_{i\in I} E_i$$

В выключенных формулах этот символ будет чуть большего размера, чем в формулах в тексте. Если в выключенных формулах этот символ имеет нижний и верхний индексы, то они печатаются под и над ним в виде пределов. Прочная.

`\bigotimes` Производит символ большого оператора \bigotimes или \bigotimes , но только в математической моде:

$$\text{\$}\bigvee\limits^n E=\text{\bigl(}\text{\textstyle}\text{\bigotimes}^n E\text{\bigr)}/S_n(E)\text{\$} \dots\dots\dots \bigvee^n E = (\bigotimes^n E)/S_n(E)$$

В выключенных формулах этот символ будет чуть большего размера, чем в формулах в тексте. Если в выключенных формулах этот символ имеет нижний и верхний индексы, то они печатаются под и над ним в виде пределов. Прочная.

`\bigskip` Производит вертикальный пробел, определяемый `\bigskipamount` — она определена как `\vspace{\bigskipamount}`. Заметим, что это определение отличается от определения этой команды в plain TeX'e. Хрупкая.

`\bigskipamount` Эластичная длина, равная 12pt plus 4pt minus 4pt.

`\bigsqcup` Производит символ большого оператора \bigsqcup или \bigsqcup , но только в математической моде. В выключенных формулах этот символ будет чуть большего размера, чем в формулах в тексте. Если в выключенных формулах этот символ имеет нижний и верхний индексы, то они печатаются под и над ним в виде пределов. Прочная.

`\bigtriangledown` Производит символ бинарного оператора \bigtriangledown , но только в математической моде. Прочная.

`\bigtriangleup` Производит символ бинарного оператора \bigtriangleup , но только в математической моде. Прочная.

`\biguplus` Производит символ большого оператора \biguplus или \biguplus , но только в математической моде. В выключенных формулах этот символ будет чуть большего размера, чем в формулах в тексте. Если в выключенных формулах этот символ имеет нижний и верхний индексы, то они печатаются под и над ним в виде пределов. Прочная.

`\bigvee` Производит символ большого оператора \bigvee или \bigvee , но только в математической моде. В выключенных формулах этот символ будет чуть большего размера, чем в формулах в тексте. Если в выключенных формулах этот

символ имеет нижний и верхний индексы, то они печатаются под и над ним в виде пределов. Прочная.

`\bigwedge` Производит символ большого оператора \bigwedge или \bigwedge , но только в математической моде. В выключенных формулах этот символ будет чуть большего размера, чем в формулах в тексте. Если в выключенных формулах этот символ имеет нижний и верхний индексы, то они печатаются под и над ним в виде пределов. Прочная.

`.blg` Расширение имени файла, который создается или перезаписывается при работе ВивТ_EX'а. Он содержит всю ту информацию, которая появляется на терминале при обработке входного файла ВивТ_EX'ом.

`\bmod` Производит символ бинарного оператора (\bmod), но только в математической моде. Прочная.

`\boldmath` Декларация, которая сама не может быть использована в математической моде, но приводит к тому, что большинство символов, которые встречаются в математической моде, следующей за этой декларацией (или находящихся в ее области действия), печатаются жирным шрифтом. Жирным шрифтом не печатаются только: все символы в индексном стиле и стиле повторного индекса, символы `+` `:` `;` `!` `?` `(` `)` `[` `]` а также символы, которые имеют переменный размер (как \sum и f), и большие ограничители, полученные командами `\left` и `\right`. Хрупкая.

`book` Одно из значений, которому может быть равен аргумент *doc-style* команды `\documentstyle`. Стиль документа `book` очень похож на `report` (см. его описание), хотя некоторые принятые в нем соглашения отличаются. В команде `\documentstyle` по умолчанию принята опция `twoside`, которая подходит для печати на обеих сторонах листа. Изменить это нельзя. По умолчанию принят стиль страницы `headings`, то есть, заголовки нечетных страниц содержат название главы и номер страницы, а четных — название секции и номер страницы. Это можно изменить, используя в команде `\pagestyle` опцию `myheadings` и связанные с ней команды, опцию `plain` или `empty`.

Если действует декларация `\raggedbottom`, то высота текста на различных страницах может слегка различаться. По умолчанию, однако, действует декларация `\flushbottom` и высота текста на всех страницах одинакова. Если присутствует команда `\maketitle` — обычно сразу после начала командных скобок `document` — то создается отдельный титульный лист. Как и в статьях, перед этим должны стоять обе команды `\author` и `\title` и может стоять команда `\date`; в аргументах этих команд могут встретиться команды `\thanks`. В стиле `book` нельзя использовать командные скобки `abstract`. Если для нумерации уравнений используются командные скобки `eqnarray`, то уравнения нумеруются внутри каждой главы последовательно, начиная с X.1, где X — номер главы, далее X.2, и т.д. Если в документе встречаются рисунки, они также нумеруются в виде X.1, X.2, ..., где X — номер главы.

команда рубрикации	номер уровня	в оглав- лении	нумеру- ется
<code>\part</code>	-1	да	да
<code>\chapter</code>	0	да	да
<code>\section</code>	1	да	да
<code>\subsection</code>	2	да	да
<code>\subsubsection</code>	3	нет	нет
<code>\paragraph</code>	4	нет	нет
<code>\subparagraph</code>	5	нет	нет

Рис. 10. Команды рубрикации стилей документа `book` и `report`.

То же можно сказать и про нумерацию таблиц ($X.1, X.2, \dots$, где X — номер главы). Заметим, что и уравнение, и рисунок, и таблица, могут иметь один и тот же номер, скажем, $X.1$ и что если командные скобки `eqnarray`, рисунок или таблица встретятся перед командой `\chapter`, то X будет равен 0. См. также описание стиля `article` и рис.5 на стр.33.

`booktitle` Имя поля элемента библиографической базы данных ВивTeX'a. Используется для названия работы, на часть которой делается ссылка.

`\bordematrix` Команда plain TeX'a для *окаймленной* матрицы. Только в математической моде. Имеет синтаксис:

```
\bordematrix{form11 & form12 & ... & form1n \cr
              form21 & form22 & ... & form2n \cr
              \vdots
              formm1 & formm2 & ... & formmn \cr}
```

Похожа на `\pmatrix`, но вдоль верхнего ряда и вдоль левого столбца получаемой матрицы размещаются метки:

$$\begin{array}{l}
\begin{array}{l}
\text{\$A=\bordematrix\{ \\
& p & & q & \backslash\text{cr} \\
p & \& I_p & \& 0 & \backslash\text{cr} \\
q & \& 0 & \& J_q & \backslash\text{cr} \\
\text{\}}\text{\$}
\end{array}
\end{array}
\left|
\begin{array}{l}
\begin{array}{cc}
p & q \\
I_p & O \\
O & J_q
\end{array}
\end{array}
\right.
A = \begin{array}{c} p \\ q \end{array} \begin{pmatrix} p & q \\ I_p & O \\ O & J_q \end{pmatrix}$$

Матрица $n \times n$ задается как матрица $(n+1) \times (n+1)$, в которой коэффициент $(1, 1)$ отсутствует. TeX сам решает, как ему разместить круглые скобки.

`\bot` Производит ординарный символ \perp , но только в математической моде.

$$\text{\$(E+F)\^{\bot}=E\^{\bot}\cap F\^{\bot}\$} \dots\dots\dots (E + F)^\perp = E^\perp \cap F^\perp$$

Имеются также команды `\perp`, `\vdash` и `\dashv`, рисующие, соответственно, значки \perp , \vdash , и \dashv . Прочная.

`\bottomfraction` Значением команды является вещественное число от 0 до 1. Оно указывает, какая часть внизу каждой страницы текста может быть занята плавающими вставками. Его можно изменить командами `\def` или `\renewcommand`. По умолчанию равно 0.3. Если выбрана опция стиля документа `twocolumn`, то команда действует только на вставки шириной в одну колонку; аналогичной команды для вставок в две колонки нет.

`bottomnumber` Счетчик, равный максимальному числу плавающих вставок, то есть, таблиц или рисунков, которые могут появиться внизу каждой текстовой страницы. По умолчанию равен 1. Если выбрана опция стиля документа `twocolumn`, значение счетчика действует только на вставки шириной в одну колонку. Аналогичной команды для вставок в две колонки нет.

`\bowtie` Производит символ бинарного отношения \bowtie , но только в математической моде. Прочная.

`\Box` Производит ординарный символ \square , но только в математической моде. Напомним, что в plain T_EX'e такой команды нет. Прочная.

`bp` Ключевое слово T_EX'a для *большого пункта* (*big point*), единицы длины, которая удовлетворяет равенству: 1 дюйм=72 bp. (Приблизительно, 1 bp= 0.0139in= 0.35mm= 1.004pt).

`{form1 \brace form2}` Производит похожее на дробь расположение символов, где *form₁* находится над *form₂*, дробной черты нет, а вся конструкция заключена в фигурные скобки. Команда plain T_EX'a. Используется только в математической моде. Синтаксис сходен с `\over`:

$$\begin{aligned} & \$p+\{n \brace k\}$ \dots\dots\dots p + \left\{ \begin{matrix} n \\ k \end{matrix} \right\} \\ & \$\displaystyle p+\{n \brace k\}$ \dots\dots\dots p + \left\{ \begin{matrix} n \\ k \end{matrix} \right\} \end{aligned}$$

`\bracevert` Доступна только в математической моде, где должна стоять либо после `\left`, либо после `\right` и где дает ограничитель, показанный на рис.5, стр.21 и рис.6, стр.22.

`{form1 \brack form2}` Производит похожее на дробь расположение символов, где *form₁* находится над *form₂*, дробной черты нет, а вся конструкция заключена в квадратные скобки. Команда plain T_EX'a. Синтаксис сходен с `\over`.

$$\begin{aligned} & \$k+\{n \brack k\}$ \dots\dots\dots k + \left[\begin{matrix} n \\ k \end{matrix} \right] \\ & \$\displaystyle\{n \brack k\}+k$ \dots\dots\dots \left[\begin{matrix} n \\ k \end{matrix} \right] + k \end{aligned}$$

`\breve` Рисует над последующим символом знак “качели” (\breve). Только в математической моде.

$$\$ \breve a$, \$ \breve x$, \$ t \backslash, \breve \{ \}$ \dots\dots\dots \breve{a}, \breve{x}, t^{\breve}$$

Прочная.

`\buildrel` Команда plain Т_EXа для размещения какого-либо математического символа над другим символом или их последовательностью. Только в математической моде:

`$x \buildrel def \over{=} 1$` $x \stackrel{def}{=} 1$
`$x\buildrel\mbox{\scriptsize слабо}`
`\over{\longrightarrow} 0$` $x \xrightarrow{\text{слабо}} 0$
`$T\buildrel\mbox{\scriptsize равномерно}\over`
`{\hbox to 2cm{\rightarrowfill}} 1$` $T \xrightarrow{\text{равномерно}} 1$
 Сравните также с командой Л^AT_EXа `\stackrel` — см. ее описание.

`\bullet` Производит символ бинарного оператора \bullet , но только в математической моде. Чтобы получить ‘ \circ ’, используйте `\circ`. (Не забывайте указывать математическую моду!) Прочная.

С

`\cchar` или `\c{char}` Производит цедиллу (\circ) под следующим за ней одним символом `char` в LR или абзацной моде:

`hame\c con, FRAN\c CON, \c S` $\text{hame}\circ\text{con}$, $\text{FRAN}\circ\text{CON}$, S

`\cal` Позволяет получать каллиграфические буквы. Работает только в математической моде и только с заглавными буквами:

`$\cal A$, $\{\cal H}-\{\cal X}$` \mathcal{A} , $\mathcal{H} - \mathcal{X}$

Не забывайте ставить фигурные скобки, иначе результаты могут оказаться непредсказуемыми:

`$\cal A*b+c*T+x/y-z$` $\mathcal{A} * [+] * \mathcal{T} + \mathcal{X} / \mathcal{Y} - \mathcal{Z}$

Прочная.

`\cap` В математической моде производит символ бинарного оператора \cap :

`$A\cap B)\cap C=A\cap(B\cap C)$` $A \cap B) \cap C = A \cap (B \cap C)$

Прочная.

`\caption{heading}` Команда, которая может встретиться только внутри командных скобок `figure`, `figure*`, `table` или `table*`. Производит пронумерованную подпись под рисунком или таблицей. Если Вам нужно сослаться на рисунок или таблицу, имеющую подпись, то внутри тела командных скобок, либо в `heading`, либо где-нибудь еще после команды `\caption` надо поставить команду `\label`. Пример использования этих команд можно видеть на рис.3 на стр.17, а результат — на рис. 4 на стр.18. Например, эта ссылка была получена так:

рис.\ref{talein} на стр.\pageref{talein}, ...
 рис.\ref{taleout} на стр.\pageref{taleout}.

Если список рисунков или таблиц получается командой `\listoffigures` или `\listoftables`, то в полученном списке появится текст из *heading*. Заметим, что *heading* является подвижным аргументом, поэтому перед любой хрупкой командой в нем должна стоять команда `\protect`. (Хрупкая).

`\caption[entry]{heading}` То же самое, что команда `\caption{heading}`, но в списках таблиц или рисунков, полученных командой `\listoftables` или `\listoffigures`, появится текст из *entry*. В этом случае подвижным аргументом является *entry*, а не *heading*. Хрупкая.

`\cases` Соединение математических условий с помощью фигурной скобки. Удобная команда Plain TeXa. В данном случае используется несколько мод: в первой колонке — математическая, чего не скажешь про вторую колонку:

$$\begin{array}{l|l} \varphi(x) = \cases{ & \\ 0 & \text{\& для } \$x \leq 0 \$, \cr & e^{-1/x} \text{\& иначе.} \cr } & \varphi(x) = \begin{cases} 0 & \text{для } x \leq 0, \\ e^{-1/x} & \text{иначе.} \end{cases} \end{array}$$

Следовательно, в первой колонке знак `$` не требуется. Напротив, ни в коем случае не следует забывать его при задании математических объектов второй колонки (“для `$X \leq 0$`”). Чтобы слегка раздвинуть строки, введите `\noalign{\smallskip}` непосредственно после `\cr`.

`сс` Ключевое слово TeX’a для единицы длины *цицера*, удовлетворяющей равенству `1 сс=12 dd`. (Приблизительно, `1 сс=0.1777 in=4.51 mm=12.84 pt`.)

`\cc{text}` Команда, которая может встретиться только внутри командных скобок `letter`. Используется для перечисления “carbon copies” — лиц, кому посылаются копии. TeX генерирует ‘сс:’ и затем *text*. Пример использования этой команды можно видеть на рис.16 на стр.79, а результат — на рис. 17 на стр.80. В аргументе *text* для обозначения конца строки могут встретиться команды `\\`.

`\cdot` Производит символ бинарного оператора `·` (точка, окруженная пробелами). Только в математической моде. Служит, например, для обозначения скалярного произведения:

$$\$x \cdot y = x_1 y_1 + \dots + x_n y_n\$ \dots \dots \dots x \cdot y = x_1 y_1 + \dots + x_n y_n$$

Чтобы поставить точку в качестве знака препинания после горизонтальной черты в дроби, используйте команду `\cdotp`, которая следует далее. Прочная.

`\cdotp` Команда, которая ставит точку в формулах не в качестве бинарного оператора, а в качестве знака препинания, и к тому же на уровне горизонтальной черты в дроби. Команда Plain TeXa, которая используется, в частности, в определении команды `\cdots`.

- `\cdots` Производит многоточие (\cdots), состоящее из трех “центрированных” точек, но только в математической моде. Пример можно посмотреть в `\cdot` (без буквы ‘s’ на конце). Вставляется между знаками $+$, $-$, \times , $=$, $>$, \geq , $<$, \leq \subset и \supset . Прочная.
- `center` Командные скобки, которые используются для центрирования текста (или любого бокса). Для того, чтобы показать, где надо начать новую строку, внутри этих командных скобок можно использовать команду `\l`. Внутри этих командных скобок ЛАТЭХ находится в абзацной моде.
- `\centering` Декларация, которая указывает, что текст (или бокс), находящийся в ее области действия, должен центрироваться в теле страницы. Заметим, что команда ТЭХ’a с тем же именем делает нечто другое. Ее оригинальное определение в ЛАТЭХ’e присвоено команде `\@centering`.
- `chapter` Поле элемента библиографической базы данных ВивТЭХ’a, содержащее номер главы, на которую делается ссылка.
- `chapter` Счетчик, который используется для управления нумерацией глав. Его начальное значение равно 0, и он увеличивается командой `\chapter` перед тем, как генерируется номер. Ему можно присвоить новое значение, используя команду `\setcounter`. (Так, например, если Вы хотите, чтобы первая глава имела номер 0, то надо в преамбулу включить команду `\setcounter{chapter}{-1}`.)
- `\chapter[entry]{heading}` Команда рубрикации, которая открывает новую главу. Ее нельзя использовать в стиле документа `article`. Если используется стиль документа `report` или `book`, то главы имеют номер уровня 0. Во всех стилях документа главы по умолчанию нумеруются автоматически. (Это можно изменить, изменяя значение счетчика `secnumdepth`.) Если отсутствует необязательный аргумент `entry`, то в оглавлении (если оно делается) по умолчанию появится `heading`. (Это можно изменить, изменяя значение счетчика `tocdepth`.) Если присутствует `entry`, то в названии главы появится `heading`, а в оглавлении — `entry`. Если `entry` присутствует, то он является подвижным аргументом, а если отсутствует, то подвижным аргументом является `heading`. Хрупкая.
- `\chapter*{heading}` Команда рубрикации, которая открывает новую главу, не имеющую номера и не появляющуюся в оглавлении. Хрупкая.
- `\chari` Выводит символ текущего шрифта, номер кода которого равен i . Если i — восьмеричное число, перед ним должна стоять открывающая кавычка, а если i — шестнадцатеричное число, это перед ним должен стоять знак двойной кавычки. Число 15 означает ‘17 в восьмеричной системе и ”F — в шестнадцатеричной. Все три команды `\char‘017`, `\char 15` и `\char"F` печатают символ текущего шрифта, десятичный код ASCII которого — 15. Это дает лигатуру ‘ffl’ в прямом шрифте и ‘*ffl*’ — в курсивном. В шрифте `\tt` Вы получите ‘)’.

TeX прекрасно понимает, что таблица шрифтов не всегда у Вас перед глазами, поэтому предлагает такое средство: `\char‘\A` печатает символ текущего шрифта, ASCII-код которого такой же, как у A (функция ‘\’ заключается в том, чтобы выдавать нужный код). Например, если текущий шрифт — `\tt`, `\char‘\{` напечатает открывающую скобку ‘{’, а `\char‘\` — обратную косую черту ‘\’ (в романском шрифте нет бэкслэша!). Очевидно, эта хитрость работает только с символами, которые есть на клавиатуре (лигатуры ‘ffi’ на клавиатуре нет). Команда примитивного TeX’a, которую нельзя использовать в математической моде.

`\check` “Галочка”, без которой некоторые математики не представляют себе жизни. Естественно, математическая мода.

`\check x*\check C+D\check{}`\$ $\check{x} * \check{C} + D$

Иногда “галочка” нужна справа от буквы. Как можно заметить, “галочка” в этом случае оказывается слишком близко к ‘D’. Приведем описание макроккоманды, которая прилично справляется с этой проблемой:

`\def\tcheche#1{#1\mkern2.5mu\check{}}`

Макроккоманда `\mkern` — это керн для математической моды. Используются единицы `mu` (математические единицы). Зачем эти сложности? Затем, что имеется четыре стиля: `\displaystyle`, `\textstyle`, `\scriptstyle` и `\scriptscriptstyle`. Значение `mu` зависит от этих стилей. Вот что получается в каждом из этих стилей (синтаксис `\tcheche A+\tcheche B`):

$$\begin{array}{c}
 A^{\sim} + B^{\sim} + C^{\sim} + D^{\sim} + M^{\sim} + X^{\sim} + Y^{\sim} + Z^{\sim} \\
 A^{\sim} + B^{\sim} + C^{\sim} + D^{\sim} + M^{\sim} + X^{\sim} + Y^{\sim} + Z^{\sim} \\
 A^{\sim} + B^{\sim} + C^{\sim} + D^{\sim} + M^{\sim} + X^{\sim} + Y^{\sim} + Z^{\sim} \\
 A^{\sim} + B^{\sim} + C^{\sim} + D^{\sim} + M^{\sim} + X^{\sim} + Y^{\sim} + Z^{\sim}
 \end{array}$$

Прочная.

`\chi` В математической моде производит ординарный символ χ . Прочная.

`{form1 \choose form2}` Производит похожее на дробь расположение символов, где `form1` находится над `form2`, дробной черты нет, а вся конструкция заключена в круглые скобки. Команда `plain TeX’a`. Синтаксис аналогичен команде `\over`, так как определение этой команды использует “дробь с ограничителем” `\atopwithdelims`:

$$\begin{array}{c}
 \mathop{p + \binom{n}{n+k}} \\
 \mathop{\displaystyle p + \binom{n}{n+k}}
 \end{array}$$

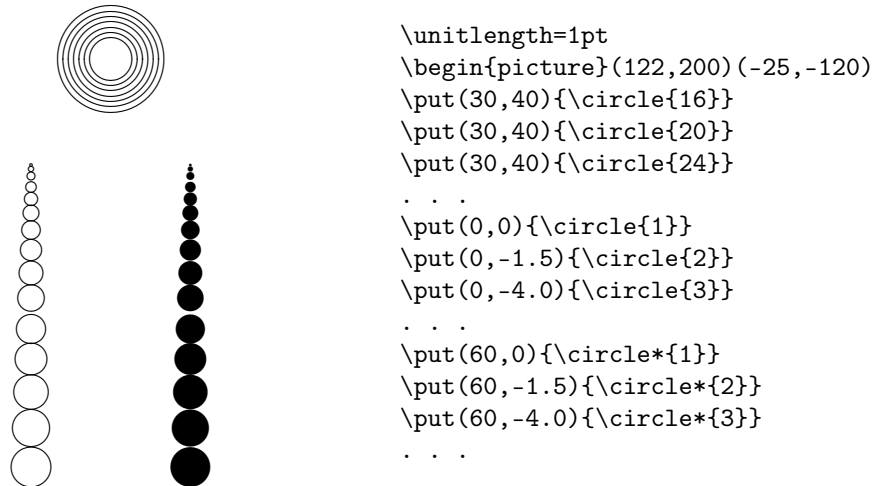


Рис. 11. Окружности и круги, доступные для построения рисунков в командных скобках `picture`.

`\circ` Производит символ бинарного оператора \circ , но только в математической моде.

$$$(f\circ g)'=f'\circ g\times g'$ $(f \circ g)' = f' \circ g \times g'$$$

Если не нравится, что вокруг кружочка остается слишком много места, предлагаем два способа исправления этого недостатка:

$$${u\circ}v$, ${\mathord\circ}v$ $u\circ v, u\circ v$$$

Чтобы нарисовать этот кружочек на месте показателя степени, например, так: 1° , напишите $1\text{\textasciicircum}\circ$. Прочная.

`\circle{d}` Команда, которая может встретиться только в качестве аргумента команды `\put` или `\multiput` внутри командных скобок `picture`. Так, `\put(i,j){\circle{d}}` помещает на картинке окружность диаметром d так, что ее центр находится в точке с координатами (i, j) . Заметим, что \LaTeX может производить только очень ограниченный набор окружностей. Размер самой большой окружности, которую он может нарисовать, равен 40 pt (что приблизительно равно половине дюйма). Также разрешается использовать только очень ограниченное число размеров диаметра. Если указанный размер не совпадает в точности с тем, что имеется в наличии, \LaTeX находит в своем репертуаре окружность, которая ближе всего подходит к указанной. См. также рис.11 на стр.39. Хрупкая.

`\circle*{d}` Команда, аналогичная команде `\circle{d}`, за тем исключением, что производится закрашенный круг или диск. Заметим, что \LaTeX может

производить только очень ограниченный набор дисков. Размер самого большого диска, который он может нарисовать, имеет диаметр, равный 15 пунктов (или около одной пятой дюйма). Также разрешается использовать только очень ограниченное число размеров диаметра. Если указанный размер не совпадает в точности с тем, что имеется в наличии, \LaTeX находит в своем репертуаре диск, который ближе всего подходит к указанному. См. также рис.11 на стр.39. Хрупкая.

`\cite[text]{key-list}` Используется для получения ссылок на библиографию, полученную \TeX ’ом или сделанную вручную. Аргумент *key-list* — это список ключей, разделенных запятыми, которые определяются либо в `\bibfile`, либо в командных скобках `thebibliography`. Например, библиография в конце этого руководства получена командными скобками `thebibliography`:

```
\begin{thebibliography}
\bibitem{...}
. . .
\bibitem{Lamp}
Leslie Lamport, ‘‘{\it \LaTeX\ A Document Preparation System}’’,
Addison-Wesley, 1986.
\bibitem{...}
. . .
\end{thebibliography}
```

и ссылка [2] на книгу Лампорта получается командой `\cite{Lamp}`. Если присутствует *text*, он добавляется к ссылке в виде аннотации. Хрупкая.

`\cleardoublepage` Команда — которую можно использовать только в абзацной моде — прерывает текущий абзац и текущую страницу. Все незаполненное место будет располагаться внизу страницы, а не между абзацами, даже если использовалась декларация `\flushbottom`. Все рисунки и таблицы будут выведены из “памяти”. Если действует двусторонняя печать, то следующий текст начнется с правосторонней страницы, имеющей нечетный номер. Если использовалась опция стиля документа `twocolumn`, то `\cleardoublepage` может привести к тому, что вся правая колонка будет пустой. Хрупкая.

`\clearpage` Команда прерывает текущий абзац и текущую страницу. Ее можно использовать только в абзацной моде. Все незаполненное место будет располагаться внизу страницы, а не между абзацами, даже если использовалась декларация `\flushbottom`. Все рисунки и таблицы, хранящиеся в “памяти”, будут выведены. Если использовалась опция стиля документа `twocolumn`, то `\clearpage` может привести к тому, что вся правая колонка будет пустой. Прочная.

`\cline{i-j}` Команда доступна только внутри командных скобок `array`, `tabular` или `tabular*`. В то время как `\hline` дает горизонтальную черту во всю

ширину таблицы, эта команда проводит горизонтальную прямую только через колонки от i -ой до j -ой, включительно. Как i , так и j обязательны, но могут быть и равными. Должна находиться после команды, которая открывает командные скобки, после команды `\\` или после другой `\cline`. Вот примеры использования этой команды в командных скобках `tabular` и в командных скобках `array`:

гнус	грамм	\$13.65
	каждый	.01
гну	чучело	92.50
лемур		33.33
армадилл	мороженный	8.99

```
\begin{tabular}{||l|lr||} \hline
гнус & грамм & \$13.65 \\
\cline{2-3}
& каждый & .01 \\
\hline
гну & чучело & 92.50 \\
\hline
лемур & & 33.33 \\
\hline
армадилл & мороженный & 8.99 \\
\hline
\end{tabular}
```

$$C = \begin{bmatrix} 1 & 0 & * & 0 & * & * \\ 0 & 1 & * & 0 & * & * \\ 0 & 0 & 0 & 1 & * & * \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

```
$$
\def\temp{\multicolumn{1}{c|}{0}}
C = \left[
\begin{array}{cccccc}
1 & & 0 & & \ast & & 0 & & \ast & & \ast \\
\ast & & & & & & & & & & \\
\temp & & 1 & & \ast & & 0 & & \ast & & \ast \\
\ast & & & & & & & & & & \\
0 & & 0 & & \temp & & 1 & & \ast & & \ast \\
\ast & & & & & & & & & & \\
0 & & 0 & & 0 & & & & 0 & & 0 & & 0 & & 0
\end{array}
\right].
$$
```

`\closing{text}` Команда, которая может встретиться только внутри командных скобок `letter`. Используется для сообщения, которое может быть в конце письма. Например, аргумент `text` может быть равен Искренне Ваш. Пример использования этой команды можно видеть на рис.16 на стр.79, а результат — на рис. 17 на стр.80.

`\clubsuit` Производит ординарный символ ♣, но только в математической моде. Прочная.

`cm` Ключевое слово TeX'a для сантиметра, единицы длины, которая удовлетворяет равенству: 1 дюйм=2.54 см. (Приблизительно, 1 cm=0.3937 in=28.45 pt.)

`\colon` Команда plain TeX'a для двоеточия (:) внутри математической формулы: Двоеточие : в математической моде не считается знаком препинания. Оно рассматривается как символ бинарного отношения. Если Вам в математической моде нужно двоеточие как знак препинания, то его надо получать командой `\colon`. Обратите внимание на различие между двумя следующими формулами:

$f: X \rightarrow Y$	<code>\$f \colon X \to Y\$</code>
$f : X \rightarrow Y$	<code>\$f : X \to Y\$</code>

`\columnsep` Параметр жесткой длины, значение которого равно ширине пробела между двумя колонками текста, если в команде `\documentstyle` задана опция `twocolumn`. Прочная команда, поэтому перед ней не надо помещать команду `\protect`.

`\columnseprule` Параметр жесткой длины, значение которого равно толщине вертикальной прямой, разделяющей две колонки текста, если в команде `\documentstyle` задана опция `twocolumn`. По умолчанию его значение равно 0 дюймам, то есть, между колонками нет видимой линии. Прочная команда, поэтому перед ней не надо ставить `\protect`.

`\cong` Производит символ бинарного отношения \cong , но только в математической моде. Имеется также команда `\simeq`, которая выдает \simeq . Прочная.

`\coprod` Может встретиться только в математической моде, где производит символ большого оператора \coprod или \coprod . Только в математической моде. Не путать с бинарной операцией `\amalg`:

`$$\amalg$`, `$$\coprod$`, `$$\displaystyle\coprod$` \coprod , \coprod , \coprod .

В выключенных формулах этот символ будет чуть большего размера, чем в формулах в тексте. Если в выключенной формуле этот символ имеет нижний и верхний индексы, то они печатаются под и над ним в виде пределов. Прочная.

`\copyright` Во всех модах производит символ копирайт ©. Прочная.

`\cos` Производит обозначение тригонометрической функции (cos), но только в математической моде.

`$$\cos^2x+\sin^2x=1$` $\cos^2 x + \sin^2 x = 1$

Нижние и верхние индексы у этого символа никогда не появляются в виде пределов. Прочная.

`\cosh` Производит обозначение тригонометрической функции (\cosh), но только в математической моде.

$$\text{\$}\cosh^2 t - \sinh^2 t = 1\text{\$} \dots\dots\dots \cosh^2 t - \sinh^2 t = 1$$

Если `\cosh` вам не по душе, используйте другое определение:

```
\def\ch{\mathop{\rm ch}\nolimits}}
```

Нижние и верхние индексы у этого символа никогда не появляются в виде пределов. Прочная.

`\cot` Производит обозначение тригонометрической функции (\cot), но только в математической моде. Если Вы предпочитаете обозначение `cotg`, посмотрите на две строки выше, где определяется макрокоманда `\ch`.

$$\text{\$}\cot(x) = \cos(x) / \sin(x)\text{\$} \dots\dots\dots \cot(x) = \cos(x) / \sin(x)$$

Нижние и верхние индексы у этого символа никогда не появляются в виде пределов. Прочная.

`\coth` Производит обозначение тригонометрической функции (\coth), но только в математической моде.

$$\text{\$}\coth x = \cosh x / \sinh x\text{\$} \dots\dots\dots \coth x = \cosh x / \sinh x$$

Сравните это с записью котангенса, где используются круглые скобки. Какая из них кажется Вам более читабельной? Нижние и верхние индексы у этого символа никогда не появляются в виде пределов. Прочная.

`\cr` Команда примитивного Т_EX'a, используется для обозначения конца строк в аргументах команд `\bordermatrix`, `\cases`, `\displaylines`, `\matrix`, `\pmatrix` и др. — см. их описание.

`\csc` Производит обозначение тригонометрической функции (\csc), но только в математической моде. Нижние и верхние индексы у этого символа никогда не появляются в виде пределов. Прочная.

`\cup` Производит символ бинарного оператора \cup , но только в математической моде.

$$\text{\$}(A \cup B) \cap C = (A \cap C) \cup (B \cap C)\text{\$}$$

$$\text{\$}\cup(B \cap C)\text{\$} \dots\dots\dots (A \cup B) \cap C = (A \cap C) \cup (B \cap C)$$

Большой символ объединения задается командой `\bigcup`:

$$\text{\$}\cup\text{\$, \$}\bigcup\text{\$, \$}\displaystyle\bigcup\text{\$} \dots\dots\dots \cup, \cup, \bigcup$$

Макрокоманда пересечения будет, естественно, выглядеть как `\bigcap`. Прочная.

d

`\d_`*char* или `\d{char}` Производит точечный акцент (◌) под следующим одним символом *char* в LR или абзацной моде.

`kr\d sna`, `KR\d SNA` `krṣna`, `KRṢNA`
Прочная.

`\dagger` Производит символ “кинжал” † во всех модах. Прочная.

`\dagger` Производит символ бинарного оператора †, но только в математической моде. Прочная.

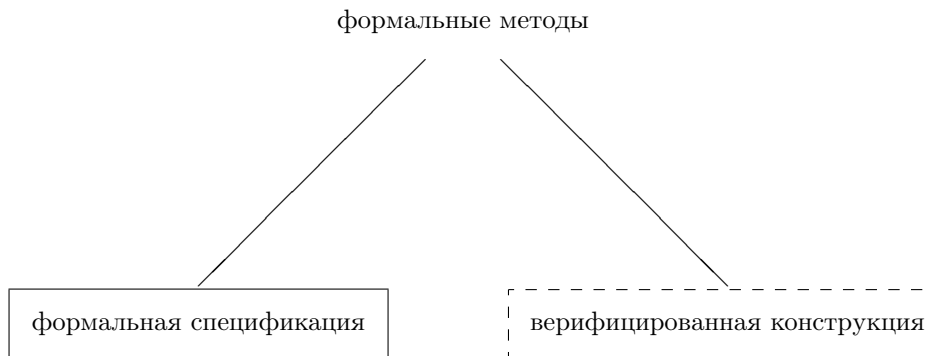
`\dashbox{h}(i,j) [pos]{объект рисунка}` Может появиться только внутри командных скобок `picture`. Производит бокс, вокруг которого пунктиром нарисована рамка. Аргументы команды имеют то же значение, что и в команде `\makebox` (см. ее описание), за исключением аргумента *h*, равного длине тире и пробелов в пунктирной линии, которой нарисована рамка. Действие этой команды показано на рис.12, стр.45. Хрущкая.

`\dashv` Производит символ бинарного отношения ⊣, но только в математической моде. Если нужно получить зеркальное отражение (⊢), дается команда `\vdash`. Прочная.

`\date{text}` (1) Используется, чтобы задать *text*, который будет использоваться в качестве даты документа. Если она опущена, то в качестве даты документа будет использоваться дата обработки L^AT_EX'ом входного файла. Для того, чтобы сделать разбиение строк, внутри *text* можно использовать команды `\`, а также одну или несколько команд `\thanks`, делающих сноски с метками, ширина которых считается равной нулю. Если команда `\thanks` не оканчивает строку, то чтобы вставить пробел, перед ней должна находиться команда `\`. (2) Также может встретиться в стиле документа `letter`, чтобы получить дату по своему выбору. В этом случае в тексте *text* могут встретиться команды `\`, но нельзя использовать команды `\thanks`.

`\dblfloatpagefraction` Действует, только если выбрана опция стиля документа `twocolumn`. Значением является вещественное число между 0 и 1, указывающее минимальную величину области страницы, которая может быть занята плавающими вставками шириной в две колонки на странице, состоящей только из плавающих вставок. Например, если оно равно 0.6, то не менее 60% страницы должно быть занято плавающими вставками шириной в две колонки. Значение этой команды можно изменить командами `\def` или `\renewcommand`. По умолчанию оно равно 0.5.

`\dblfloatsep` Если в команде `\documentstyle` задана опция стиля документа `twocolumn`, то этот параметр эластичной длины действует на размещение



```

\begin{center}
%
\unitlength=1mm
\begin{picture}(110,50)(0,5)
%
\put(55,45){\makebox(0,0){формальные методы}}
\put(20,5){\makebox(0,0)%
{\framebox(50,9){формальная спецификация}}}
\put(90,5){\makebox(0,0){\dashbox{1.5}(58,9)%
{верифицированная конструкция}}}
%
\put(50,40){\line(-1,-1){30}}
\put(60,40){\line(1,-1){30}}
%
\end{picture}
%
\end{center}
  
```

Рис. 12. Пример использования команд `\makebox`, `\dashbox` и `\framebox` в командных скобках `picture`.

плавающих вставок двухколоночной ширины. Он равен величине вертикального пробела, который помещается между двумя плавающими вставками на странице, содержащей как текст, так и вставки. Прочная команда, перед которой не надо ставить команду `\protect`.

`\dbltextfloatsep` Параметр эластичной длины, который действует, только если выбрана опция стиля документа `twocolumn`. Он равен величине вертикального пробела, вставляемого между двухколоночной плавающей вставкой и текстом над и под ней. Прочная команда, перед которой не надо ставить `\protect`.

`\dbltopfraction` Действует, только если выбрана опция стиля документа `twocolumn`. Это вещественное число между 0 и 1, указывающее, какая часть вверху каждой текстовой страницы может быть использована для двухколоночных плавающих вставок. По умолчанию значение равно 0.7 и может быть изменено командами `\def` и `\renewcommand`.

`dbltopnumber` Счетчик, который действует, только если выбрана опция стиля документа `twocolumn`. Равен максимальному числу плавающих вставок, которые могут появиться вверху каждой текстовой страницы. По умолчанию равен 2.

`dd` Ключевое слово Т_EX'a для обозначения *пункта Дудо* (*didot point*), единицы длины, которая удовлетворяет равенству $1157 \text{ dd} = 1238 \text{ pt}$. (Приблизительно, $1 \text{ dd} = 0.0148 \text{ in} = 0.376 \text{ mm} = 1.07 \text{ pt}$.)

`\ddag` Производит символ ‡ во всех модах. Прочная.

`\ddagger` Производит символ бинарного оператора ‡, но только в математической моде. Прочная.

`\ddot` Производит акцент в математической моде. Необходима для тех, кто использует для производных ньютоновские обозначения:

$$\text{\$}\ddot{x}(t)=f(x,t)\text{\$} \dots\dots\dots \ddot{x}(t) = f(x,t)$$

Прочная.

`\ddots` Производит диагональное многоточие \ddots , состоящее из трех расположенных по диагонали точек, но только в математической моде. Прочная.

`\def cmd#1#2...#i{def}` Определяет команду *cmd* — которая должна начинаться с бэкслэша — как команду, эквивалентную *def*. Определяемая команда может иметь до 9 аргументов. Когда употребляется такая команда с параметрами, как `cmd{arg1}{arg2}...{argi}`, она заменяется на *def*, где вместо #*j* (для $1 \leq j \leq i$) подставляется *arg_i*. Следует быть внимательным и не оставлять в списке аргументов никаких пробелов — ни перед, ни между, ни после #*j*. В этом руководстве можно встретить множество примеров использования команды `\def`. Подробное обсуждение этой команды можно найти в главе 20 книги Кнут (1993). Команда примитивного Т_EX'a.

`\deg` Оператор, который пишет в формулах “deg” прямым шрифтом. Только в математической моде:

```
\deg(FG)=\deg F+\deg G .....  $\deg(FG) = \deg F + \deg G$ 
```

Если Вам больше нравятся градусы, составьте другую макрокоманду:

```
\def\degres{^\circ}
```

```
угол в 138\degres ..... угол в 138°
```

Нижние и верхние индексы у символа никогда не появляются в виде пределов. Прочная.

`\delta` Производит ординарный символ δ , но только в математической моде. Для получения ординарного символа ∂ используйте `\partial`. Прочная.

`\Delta` Производит ординарный символ Δ , но только в математической моде. Прочная.

`description` Командные скобки для получения глоссариев. Команда `\item[text]` внутри них печатает *text* жирным шрифтом. Например, команды

```
\begin{description}
\item[филер]
Поставщик таких “важных” услуг, как наблюдение и слежка.
\item[киллер]
Агент, обязанность которого убивать указанных людей.
\item[медвежатник]
Вор, специалист по взлому сейфов.
\end{description}
```

дают следующий мини-глоссарий детективного жаргона:

филер Поставщик таких “важных” услуг, как наблюдение и слежка.

киллер Агент, обязанность которого убивать указанных людей.

медвежатник Вор, специалист по взлому сейфов.

Заметим, что хотя определяемые слова задаются здесь как необязательные аргументы команд `\item`, но если их опустить, результат, получаемый \LaTeX ’ом, будет выглядеть очень странно. По умолчанию необязательный аргумент команды `\item` печатается в выходном документе жирным шрифтом, но это можно изменить, если внутри квадратных скобок поставить изменяющую стиль шрифта декларацию, например, так:

```
\begin{description}
\item[\tt chapter]
Счетчик, используемый \LaTeX’ом для нумерации глав.
\item[ $\{x\}X$ ]

```

```

Используется в комбинаторной логике для представления
скобочной абстракции.
\item[\it пенальти]
Одиннадцатиметровый штрафной удар в футболе за нарушение
правил в штрафной площадке.
\end{description}

```

Эти команды дают следующий результат:

`chapter` Счетчик, используемый ЛАТ_EX'ом для нумерации глав.

`[x]X` Используется в комбинаторной логике для представления скобочной абстракции.

`пенальти` Одиннадцатиметровый штрафной удар в футболе за нарушение правил в штрафной площадке.

Отметим присутствие в `\item[$\{[x]\}X$]` фигурных скобок. Они нужны, чтобы ЛАТ_EX не перепутал квадратные скобки, заключающие необязательный аргумент, с квадратными скобками, которые надо напечатать в выходном документе.

`\det` Оператор `det` (для сокращенного обозначения детерминанта). Только в математической моде:

Если `$\det A \neq 0$` , матрица `A^{-1}` существует.
 Если `$\det A \neq 0$` , матрица `A^{-1}` существует.

Для печати самого детерминанта матрицы используйте команду

```
\left|\matrix{...}\right|.
```

Если символ имеет нижний и верхний индексы, то в выключенных формулах они располагаются под и над ним — в виде пределов. Прочная.

`\diamond` Производит символ бинарного оператора \diamond , но только в математической моде.

закон `$x \diamond y = xy / (x + y)$` закон `$x \diamond y = xy / (x + y)$`

Прочная.

`\Diamond` Производит символ бинарного оператора \Diamond , но только в математической моде. Заметим, что в plain Т_EX'е этой команды нет. Прочная.

`\diamondsuit` Производит ординарный символ \diamond , но только в математической моде. Прочная.

`\dim` Производит оператор (`dim`) (для размерности). Только в математической моде:

```
 $\dim(U+V) = \dim U + \dim V - \dim U \cap V$  .....
```


..... $\dim(U + V) = \dim U + \dim V - \dim U \cap V$

Нижние и верхние индексы у символа никогда не появляются в виде пределов. Прочная.

`\displaylimits` Команда `cmd\displaylimits_form1~form2` в выключенном стиле (в математической моде) помещает `form1` в виде нижнего предела под символом, который производит `cmd`, а `form2` помещается над этим символом в виде верхнего предела. Восстанавливает нормальные соглашения о пределах больших операторов (\sum , \int , ...), которые могли быть Вами изменены командами `\limits` и `\nolimits`. Команда примитивного Т_ЕX'а.

`\displaylines{form1\cr form2\cr ...form_n\cr}` Каждая из `form_i` помещается на отдельной строке и центрируется на ней. Только в выделенной математической моде:

```
$$\displaylines{
x_1+\cdots+x_n=y_1+\cdots+y_m,\cr
A=B+C. \cr
}$$
```

Таким образом мы получим две отцентрированные строки:

$$x_1 + \cdots + x_n = y_1 + \cdots + y_m,$$
$$A = B + C.$$

Синтаксис такой же, как в таблицах без преамбулы с одной колонкой. Не забывайте, что каждая строка заканчивается `\cr`. Довольно частая ошибка, которая приводит в смущение многих начинающих пользователей — ставить знак пунктуации после `\cr`. Т_ЕX справедливо предполагает, что начинается новая строка. Он не встречает `\cr`, который закрывает эту строку, но, поскольку существует механизм, который исправляет эту ошибку (`\crr`), Т_ЕX не жалуется. Единственный осязаемый результат возникает в том случае, если знак пунктуации оказывается размещенным на другой строке, в центре, как если бы это была новая формула. Грубой ошибкой, встречающейся довольно часто, считается пропуск фигурной скобки, закрывающей запись макрокоманды, т.е. примерно так:

```
$$\displaylines{
...
... \cr}$$
```

В конце программы написан `\cr`, но закрывающая фигурная скобка отсутствует. После множества протестов Т_ЕX'а Вы не получите ... ничего! Команда примитивного Т_ЕX'а.

`\displaymath` Командные скобки для набора математической формулы на отдельной строке. Имеются и более сжатые формы: `\[form\]` и `$$form$$`,

причем последняя конструкция не затрагивается опцией `fleqn` в команде `\documentstyle`. Внутри этих командных скобок \LaTeX работает в математической моде и в выключенном стиле.

`\displaystyle` Декларация, которую можно использовать только в математической моде. Указывает \TeX 'у набирать формулу в выключенном стиле, т.е. в стиле, который по умолчанию принят для набора расположенной на отдельной строке формулы: пробелы по обе стороны от формулы шире, чем в обычной математической моде (`\textstyle`). Есть разница в размерах символов и в положении индексов. См. пример с `\displaystyle` (между двумя одиночными долларами мы находимся в режиме `\textstyle`, чтобы задать режим `\displaystyle`, нужно запрашивать его явно):

`\displaystyle\sum_{n=0}`
`^{\infty}\{x_n\over 1+x_n^2\}` $\sum_{n=0}^{n=\infty} \frac{x_n}{1+x_n^2}$

Вот та же самая формула в `\textstyle`:

`\sum_{n=0}^{\infty}\{x_n\over 1+x_n^2\}` $\sum_{n=0}^{n=\infty} \frac{x_n}{1+x_n^2}$

Разумеется, можно в любом месте задавать `\displaystyle`:

дробь `\{a\over b\}` дробь $\frac{a}{b}$
 дробь `\displaystyle\{a\over b\}` дробь $\frac{a}{b}$
 дробь `\{\displaystyle a\over b\}` дробь $\frac{a}{b}$
 дробь `\{a\over\displaystyle b\}` дробь $\frac{a}{b}$

Команда примитивного \TeX 'а. Прочная.

`\div` Производит символ бинарного оператора \div , но только в математической моде:

`\div 4=3` $15 \div 4 = 3$

Прочная.

`\divide` Делит одно число на другое. Часть языка программирования \TeX . Команда `\divide\dimen117 by 12` заменяет содержимое `\dimen117` на результат его деления на 12, т.е. в современном языке программирования надо было бы писать `\dimen117:=\dimen117 div 12`.

`.doc` Расширение имени файла, содержащего те же команды, что и соответствующий `sty`-файл, но с дополнительными комментариями и пояснениями.

`document` Самые внешние командные скобки для всех входных файлов. Каждый входной файл \LaTeX 'а имеет следующую структуру:

```

\documentstyle [opt-list] {doc-style}
  dec-seq
\begin{document}
  text
\end{document}

```

Часть входного файла Л^AT_EX'a, которая располагается между командой `\documentstyle` и открывающей командной скобкой `document`, называется *преамбулой*. Здесь она представлена в виде *dec-seq* и состоит из возможно пустой последовательности деклараций, которые влияют на окончательный вид входного документа.

`\documentstyle [opt-list] {doc-style}` Команда, обязательная в каждом входном файле Л^AT_EX'a. Задаёт стиль документа. Стандартными стилями являются `article`, `report`, `book` и `letter`, но в Вашей системе могут быть и другие. Можно задать только один *doc-style*, и по команде `\documentstyle` будет читаться файл *doc-style.sty*.

Стандартными опциями являются `11pt`, `12pt`, `draft`, `fleqn`, `leqno`, `openbib`, `titlepage`, `twocolumn` и `twoside`.⁵ Если задается более одной опции, они должны разделяться запятыми и между ними нельзя вставлять пробелы. Для каждой опции *opt*, присутствующей в *opt-list*, Л^AT_EX либо выполняет команду `\ds@opt` (если она существует), либо читает файл *opt.sty*. Перед командой `\documentstyle` может встретиться только небольшое количество команд; например, команды `\batchmode`, `\errorstopmode`, `\nonstopmode` и `\scrollmode`. Пример этой команды можно видеть на рис.16 на стр.79, а результат — на рис.17 на стр.80.

`\dot` Для изображения точки над символом, следующим за командой. Только в математической моде. Незаменяема при изображении ньютоновских производных:

`\dot x(t)=f\bigl(x(t),t\bigl)r$ $\dot{x}(t) = f(x(t), t)$`

Обратите внимание на использование команд plain Т_EXа `\bigl` и `\bigr`, которые увеличивают размеры ограничителей (в данном случае, круглых скобок). Для получения двух точек над каким-либо символом используется команда `\ddot`. Прочная.

`\doteq` Производит символ бинарного отношения $\dot{=}$, но только в математической моде. Прочная.

`\dotfill` Производит ряд точек — как, например, в оглавлениях — который заполняет все отведенное им место, как это показано на рис.13, стр.52. При наборе примеров команд в данном руководстве она также часто использовалась. Прочная.

⁵Для подготовки текстов на русском языке используется русифицированный Л^AT_EX, который называется опцией `russian`.







	<code>\makebox[2in]{\hrulefill}</code>
	<code>\makebox[2in]{\downbracefill}</code>
	<code>\makebox[2in]{\upbracefill}</code>
	<code>\makebox[2in]{\dotfill}</code>
	<code>\makebox[2in]{\leftarrowfill}</code>
	<code>\makebox[2in]{\rightarrowfill}</code>

Рис. 13. Шесть fill-команд.

`\doublerulesep` Параметр жесткой длины, равный расстоянию, разделяющему либо две вертикальные прямые, которые производятся выражением `||` в преамбуле командных скобок `array`, `tabular` и `tabular*`, либо две горизонтальные прямые, получаемые двумя последовательными командами `\hrule`. Прочная команда, поэтому перед ней не надо помещать `\protect`.

`\downarrow` Производит символ бинарного отношения \downarrow , но только в математической моде. Однако после команды `\left` или `\right` она дает соответствующий ограничитель. Для получения вертикальной стрелки высотой 12 мм, например, задайте:

```
\left\downarrow\vbbox to 6mm{}\right.
```

См. также рис.5 на стр.21 и рис.6 на стр.22. Прочная.

`\Downarrow` Производит символ бинарного отношения \Downarrow , но только в математической моде. Однако после команды `\left` или `\right` она дает соответствующий ограничитель. Замечания в описании предыдущей команды. Прочная.

`\downbracefill` Производит горизонтальную фигурную скобку, направленную острием вверх, которая растягивается на все отведенное ей горизонтальное пространство. Графически это показано на рис.13, стр.52.

`draft` Возможная опция команды `\documentstyle`. Если она задана, то случаи, когда текст залезает на правое поле, будут на этом поле помечены черным прямоугольником.

`\ds@opt` Если опция `opt` появляется в списке опций команды `\documentstyle`, то выполняется команда `\ds@opt` — если она определена — в противном случае выполняются команды из стилевого файла `opt.sty`.

`.dvi` Расширение имени независимого от устройства (device independent) файла, создаваемого при обработке L^AT_EX'ом соответствующего входного файла (с расширением `tex`).

е

`edition` (Имя поля В_IT_EX'а.) Указывает, на какое издание делается ссылка. Например, 7 или “Тридцать девятое”. Должно задаваться либо числом, либо числительным, написанным с прописной буквы.

`editor` (Имя поля В_IT_EX'а.) Содержит имя редактора книги (или имена, если редакторов несколько) или имя редактора части книги, такой как труды конференции (или имена, если редакторов несколько).

`\ell` Производит ординарный символ ℓ , но только в математической моде. Сравните следующие примеры и выберите то, что Вас больше устраивает:

`$l-l*x_1^2+y_1^\alpha$` $l - l * x_1^2 + y_1^\alpha$
`$_\ell-\ell*x_\ell^2+y_\ell^\alpha$` $\ell - \ell * x_\ell^2 + y_\ell^\alpha$

Две другие латинские буквы в этих формулах пишутся в несколько модифицированном виде: `\imath` (i) и `\jmath` (j). Это позволяет использовать их вместе со стрелками (или с чем-либо еще): в результате `$_\vec{\imath}$` и `$_\vec{\jmath}$` получается \vec{i} и \vec{j} . Прочная.

`\em` Зависимая от шрифта единица длины. Когда-то она была равна ширине заглавной буквы М, но сейчас может иметь и другое значение. Обычно используется для горизонтальных размеров. Она также равна ширине одного `\quad`, двойной ширине `\enskip` или `\enspace` и половине `\qqquad`.

`\em` Изменяет текущий шрифт печати. В то время как команда `\it` всегда задает курсив, а команда `\rm` — романский шрифт, действие команды `\em` зависит от контекста, в котором она встречается. Например, если действовала `\rm`, то `\em` эквивалентна `\it`, а если действовала `\it`, то `\em` эквивалентна `\rm` — см. рис.9 на стр.28. Прочная.

`empty` Опция стиля страницы в декларациях `\pagestyle` и `\thispagestyle`. Заголовков и основание страницы оставляются пустыми. Хотя номера страниц и не появляются в выходном результате, Л_AT_EX продолжает их присваивать. Текущая страница напечатана без номера и бегущего заголовка, поскольку была дана команда `\thispagestyle{empty}`.

`\emptyset` Производит ординарный символ \emptyset , но только в математической моде. Не путать со скандинавской буквой `\O` (\mathcal{O}), которая имеет более округлую форму. Прочная.

`\encl{text}` Встречается только внутри командных скобок `letter`. Используется для описи вложений. Л_AT_EX генерирует ‘encl:’, а затем помещает `text`. Внутри аргумента `text` для указания на начало новой строки можно использовать одну или несколько команд `\\`.

`\end{env}` Команда для окончания командных скобок `env`. Хрупкая.

`\enskip` Во всех модах производит горизонтальный пробел, ширина которого равна `\hskip .5em\relax` текущего шрифта. Команда `\relax` служит для отделения `\hskip` от последующего текста в том случае, когда он начинается с “plus”, “Plus”, “minus” или “Minus”. Т_ЭX’у в этом случае может показаться, что написано `\hskip .5em plus...`, и он будет горько жаловаться на отсутствие размера после `plus`. Это команда plain Т_ЭX’а, и все подробности приведены здесь только для общего образования.

`\enspace` Во всех модах производит горизонтальный пробел, ширина которого равна половине `em`. А противоположность `\enskip`, в том месте, где встретилась эта команда, не может произойти разрыв строки. Команда plain Т_ЭX’а.

`enumerate` Командные скобки для получения нумерованных списков. Нумерованные списки в Л^AT_ЭX’е выглядят так:

- 1) Докажите следующие формулы из логики высказываний:
 - а) $Q \Rightarrow R \vdash (P \vee Q) \Rightarrow (P \vee R)$.
 - б) $\vdash ((P \Rightarrow Q) \Rightarrow P) \Rightarrow P$.
- 2) Докажите следующие формулы из логики предикатов:
 - а) $(\forall x)(Px \Rightarrow Qx) \vdash (\forall x)Px \Rightarrow (\forall x)Qx$.
 - б) $(\exists x)(\forall y)Pxy \vdash (\forall y)(\exists x)Pxy$.

Этот список был получен следующими командами:

```
\begin{enumerate}
\item
Докажите следующие формулы из логики высказываний:
%
\begin{enumerate}
\item
 $\$Q \Rightarrow R \vdash (P \vee Q) \Rightarrow (P \vee R)\$$ .
\item
\label{PEIRCE}
 $\$ \vdash ((P \Rightarrow Q) \Rightarrow P) \Rightarrow P \$$ .
\end{enumerate}
%
\item
Докажите следующие формулы из логики предикатов:
%
\begin{enumerate}
\item
```

```

 $(\forall x) (Px \rightarrow Qx)$ 
 $\vdash (\forall x) Px \rightarrow (\forall x) Qx$ .
\item
 $(\exists x)(\forall y) Pxy \vdash (\forall y)(\exists x) Pxy$ .
\end{enumerate}
%
\end{enumerate}

```

В списках этого вида разрешены только четыре уровня вложенности. Пункты верхнего уровня помечаются 1., 2. и т.д., следующего уровня — (a), (b) и т.д., на следующем уровне метками будут i., ii. и т.д., а на четвертом уровне появятся метки A., B. и т.д. Метки генерируются автоматически, для чего используются четыре счетчика: `enumi`, `enumii`, `enumiii` и `enumiv`.

Команда `\item` командных скобок `enumerate` может иметь необязательный аргумент. Так, например, команды

```

\begin{enumerate}
\item[(1)]
 $\rightarrow R \vdash (P \vee Q) \rightarrow (P \vee R)$ .
\item[(2)]
 $\vdash ((P \rightarrow Q) \rightarrow P) \rightarrow P$ .
\end{enumerate}

```

дают следующий нумерованный список:

- (1) $Q \Rightarrow R \vdash (P \vee Q) \Rightarrow (P \vee R)$.
- (2) $\vdash ((P \Rightarrow Q) \Rightarrow P) \Rightarrow P$.

Если в команде `\item` задается необязательный аргумент, то соответствующий счетчик не увеличивается.

`enumi` Счетчик для отслеживания меток в командных скобках `enumerate`, которые не вложены в другие командные скобки `enumerate`.

`enumii` Счетчик для отслеживания меток в командных скобках `enumerate`, которые вложены в одну пару других командных скобок `enumerate`.

`enumiii` Счетчик для отслеживания меток в командных скобках `enumerate`, которые вложены в две пары других командных скобок `enumerate`.

`enumiv` Счетчик для отслеживания меток в командных скобках `enumerate`, которые вложены в три пары других командных скобок `enumerate`.

`\epsilon` Производит ординарный символ ϵ , но только в математической моде. Чтобы получить ординарный символ ε , надо использовать `\varepsilon`, а для бинарного отношения \in — `\in`.

`eqnarray` (Массив уравнений.) Командные скобки для получения нескольких выключенных уравнений, каждое из которых имеет номер. (Чтобы подавить нумерацию, надо использовать командные скобки `eqnarray*`.) Внутри них \LaTeX находится в математической моде. Например, команды

```
\begin{eqnarray}
(x + y + z)^2 & = & (x + y + z) (x + y + z), \label{BB} \\
& = & x^2 + 2xy + y^2 \nonumber \\
& & \quad \mbox{} + 2yz + z^2 + 2zx. \label{CC}
\end{eqnarray}
```

дают следующий помеченный набор выключенных уравнений:

$$(x + y + z)^2 = (x + y + z)(x + y + z), \quad (1)$$

$$= x^2 + 2xy + y^2 + 2yz + z^2 + 2zx. \quad (2)$$

Команда `\nonumber` подавляет автоматическое получение метки на той строке, на которой она встречается.

`eqnarray*` Командные скобки для получения нескольких непрономерованных выключенных уравнений. Результат — как при использовании командных скобок `array` с преамбулой `rc1` с тем исключением, что выражения в первой и третьей колонках набираются в выключенном стиле, в то время как выражение во второй колонке набирается в текстовом стиле. К тому же в командных скобках `eqnarray` нельзя использовать команду `\multicolumn`. Конец ряда указывается командным символом `\\`, а элементы каждой из трех колонок разделяются амперсандами `&`. В конце последней строки командный символ конца строки не нужен. Если же его все-таки поставить, в выходной результат будет добавлена пустая строка. Например, команды

```
\begin{eqnarray*}
{[x]} E & = & {\bf K} E, \\
{[x]} x & = & {\bf I}, \\
{[x]} E x & = & E, \\
{[x]} E X & = & {\bf B} E ([x] X), \\
{[x]} X E & = & {\bf C} ([x] X) E, \\
{[x]} X Y & = & {\bf S} ([x] X) ([x] Y).
\end{eqnarray*}
```

дают следующий алгоритм:

$$\begin{aligned} [x]E &= \mathbf{K}E, \\ [x]x &= \mathbf{I}, \\ [x]Ex &= E, \end{aligned}$$

$$\begin{aligned} [x]EX &= \mathbf{BE}([x]X), \\ [x]XE &= \mathbf{C}([x]X)E, \\ [x]XY &= \mathbf{S}([x]X)([x]Y). \end{aligned}$$

Следует отметить правильно расставленные знаки препинания. То, что вы пишете математический текст, вовсе не означает, что надо забыть все правила пунктуации, которые вы когда-то знали.

`\eqno` Команда примитивного Т_EX'а, доступная только в математической моде и в выключенном стиле, где производит метку уравнения в правой части тела страницы.

`$$e^{i\pi}=-1\eqno{\rm (ЭЙЛЕР)}$$`

$$e^{i\pi} = -1 \quad ()$$

Чтобы поставить номер слева от формулы, используется `\leqno`:

`$$e^{i\pi}=-1\leqno{\rm (ЭЙЛЕР)}$$`

$$() \quad e^{i\pi} = -1$$

Вы заметили `\rm`, обязательный из-за того, что в математической моде происходит смена шрифтов? Внимание: хотя в последнем варианте номера печатаются слева, в команде тем не менее они должны быть записаны справа от формулы. И последняя рекомендация: Вы не можете использовать макрокоманду `\eqno` в `\displaylines`. Иначе говоря, `\eqno` позволяет нумеровать только *одну изолированную отцентрированную строку*.

`equation` Счетчик, используемый командными скобками `equation` и `eqnarray`, чтобы отслеживать, как помечаются выключенные формулы.

`equation` Командные скобки, производящие выключенную математическую формулу, числовая метка которой генерируется автоматически. В стиле документа `article` формулы нумеруются последовательно во всем документе, начиная с (1), а метка состоит просто из одного номера. В стилях документа `report` и `book` формулы нумеруются последовательно внутри каждой главы, начиная с (X.1), где X — это номер главы, а все метки состоят из двух чисел. Так, помеченная формула

$$(x + y)(x - y) = x^2 - y^2. \quad (3)$$

была получена следующими командами:

```
\begin{equation}
(x + y)(x - y) = x^2 - y^2.
\label{AA}
\end{equation}
```

Функция команды `\label` — облегчить ссылку на эту формулу из других частей написанного Вами текста.

`\equiv` Производит символ бинарного отношения \equiv , но только в математической моде. Противоположное отношение $\not\equiv$ задается с помощью `\not\equiv`. Прочная.

`\errorstopmode` Команда примитивного TeX'a, после которой TeX, когда наталкивается на ошибку, запрашивает ответ пользователя. По умолчанию принята именно эта мода работы. Одна из немногих команд, которые во входном файле могут стоять перед командой `\documentstyle`.

`\eta` Производит ординарный символ η , но только в математической моде. Прочная.

`\evensidemargin` Параметр жесткой длины, действующий на внешний вид четных страниц выходного результата. На левосторонних страницах — если задана двусторонняя печать — расстояние между левой границей листа бумаги и левой границей тела страницы равно сумме значения `\evensidemargin` и одного дюйма. (См. рис.23 на стр.137.) Прочная команда, перед которой не надо ставить `\protect`.

`ex` Зависящая от шрифта единица длины, которая изначально была равна высоте строчной буквы 'x', но сейчас может и не быть таковой. Обычно используется для вертикальных измерений.

`\exists` Производит ординарный символ \exists , но только в математической моде. Для логиков, а также для любителей “современной математики”: Например, такое математическое определение

Функция f непрерывна по x , если $\forall \varepsilon \in \mathbf{R}_*^+ \exists \nu \in \mathbf{R}_*^+$ такое, что $\forall h \in \mathbf{R}$ $h < \nu, |h| < \nu$ выполняется $|f(x + h) - f(x)| < \varepsilon$

было получено следующими командами:

```
{\sl Функция  $f$  непрерывна по  $x$ , если
 $\forall \varepsilon \in \mathbf{R}_*^+ \exists \nu \in \mathbf{R}_*^+$  такое, что
 $\forall h \in \mathbf{R} \quad |h| < \nu, |h| < \nu$ 
выполняется  $|f(x + h) - f(x)| < \varepsilon$ 
```

Прочная.

`\exp` Производит обозначение элементарной функции (\exp), но только в математической моде. Нижние и верхние индексы у нее никогда не появляются в виде пределов.

`\exp(x+y)=\exp x\exp y` $\exp(x + y) = \exp x \exp y$

`\extracolsep{len}` Параметр длины, который может встретиться только внутри @-выражения в преамбулах командных скобок `array`, `tabular` и `tabular*`. Помещает пробел величиной *len* слева от всех следующих колонок, который не подавляется присутствием других @-выражений.

f

`\fbox{text}` Сокращенная версия команды `\framebox`. Производит \boxed{text} во всех модах, но не внутри командных скобок `picture`, причем *text* обрабатывается в LR моде. Толщина образующих рамку линий определяется значением параметра длины `\fboxrule`, а ширина пробела, отделяющего *text* от рамки, задается параметром длины `\fboxsep`. Ширина результирующего бокса равна ширине бокса, полученного при обработке текста *text*, плюс удвоенная сумма `\fboxrule` и `\fboxsep`. Прочная.

`\fboxrule` Параметр жесткой длины, равный толщине горизонтальных и вертикальных “рамочных” линий, которые получаются командами `\fbox` или `\framebox`, но не внутри командных скобок `picture`. Толщину линий внутри командных скобок `picture` задают другие команды.

`\fboxsep` Параметр жесткой длины, равный величине пробела, который отделяет бокс, получаемый при обработке аргумента *text* командой `\fbox` или `\framebox`, от горизонтальных и вертикальных прямых, образующих окружающую его “рамку”, но не внутри командных скобок `picture`. Внутри командных скобок `picture` такой пробел не оставляется.

`figure` Счетчик для нумерации плавающих вставок, которые создаются командными скобками `figure` и `figure*`. Автоматически увеличивается только тогда, когда внутри этих командных скобок имеется команда `\caption`.

`figure` Командные скобки, которые производят плавающую вставку — см. рис.3 на стр.17 и рис. 4 на стр.18. Если выбрана опция стиля документа `twocolumn`, то плавающие вставки имеют ширину только одной колонки и аргумент *pos* не может принимать значение `b` и `h`. В противном случае они ведут себя так же, как командные скобки `figure*` (см. ниже).

`figure*` Командные скобки, которые производят плавающую вставку. Если внутри них имеется команда `\caption`, то автоматически генерируется слово ‘Figure’⁶ и числовая метка. Если выбрана опция стиля документа `twocolumn`, то производимая вставка имеет ширину в две колонки. Их синтаксис такой:

```
\begin{figure*}[pos] text \end{figure*}
```

⁶В русифицированной версии — ‘Рис.’

Аргумент *text* обрабатывается в абзацной моде и при этом получается абзацный бокс шириной `\textwidth`. Необязательный аргумент *pos* — это последовательность из от одной до четырех букв, выбранных из **b**, **h**, **p** и **t**. Он действует на позиционирование плавающей вставки следующим образом:

- b** вставка может появиться внизу текстовой страницы;
- h** вставка в выходном результате появляется в том же положении относительно текста вокруг нее, что и во входном файле;
- p** вставка может появиться на странице, составленной только из вставок;
- t** вставка может появиться сверху текстовой страницы.

По умолчанию значение *pos* равно `tbp`.

`\fill` Команда эластичной длины, которая может “бесконечно” растягиваться.

`\flat` Производит ординарный символ `b`, но только в математической моде. Прочная.

`fleqn` Опция стиля документа: все выключенные уравнения, которые встречаются между командами `\[` и `\]`, а также внутри командных скобок `displaymath`, `equation`, `eqnarray` и `eqnarray*`, не центрируются, как это принято по умолчанию, а печатаются с отступом от левого поля, равным `\mathindent`. Заметим, что эта опция не действует на уравнения внутри пар из двойных знаков долларов. Они продолжают центрироваться, даже если выбрана опция `fleqn`.

`\floatpagefraction` Значение команды является вещественным числом между 0 и 1. Указывает минимальную часть страницы, которая должна быть занята плавающими вставками, если на странице располагаются только вставки. Например, если оно равно 0.6, то не менее 60% такой страницы должно быть занято плавающими вставками. По умолчанию равно 0.5, но его можно изменить командами `\def` или `\renewcommand`. Если выбрана опция стиля документа `twocolumn`, то действует только на вставки шириной в одну колонку; для плавающих вставок шириной в две колонки см. команду `\dblfloatpagefraction`.

`\floatsep` Параметр эластичной длины, равный величине вертикального пробела между плавающими вставками, которые встречаются на одной и той же текстовой странице. Если выбрана опция стиля документа `twocolumn`, то эта команда действует только на плавающие вставки шириной в одну колонку; для вставок шириной в две колонки см. `\dblfloatsep`. Прочная команда, перед которой не надо ставить `\protect`.

`\flushbottom` Декларация, делающая высоту тела всех страниц одинаковой, для чего между абзацами, если надо, вставляются дополнительные пробелы.

`flushleft` Командные скобки для получения абзацев, которые справа не выровнены, а имеют неровный правый край.

`flushright` Командные скобки для получения абзацев, которые слева не выровнены, а имеют неровный левый край.

`\fnsymbol{ctr}` Здесь *ctr* — счетчик, значение которого должно находиться между 1 и 9, включительно. Команда может использоваться только в математической моде, где, в зависимости от счетчика *ctr*, производит один из следующих девяти символов: *, †, ‡, §, ¶, |, **, †† и ‡‡. Например, команды:

```
\newcounter{stone}\setcounter{stone}{4}\fnsymbol{stone}$
```

дают §. Прочная.

`\footheight` Параметр жесткой длины, действующий на внешний вид страницы результата. Равен высоте бокса, содержащего текст в основании страницы. См. рис.23 на стр.137. Прочная команда, поэтому перед ней не надо ставить `\protect`.

`footnote` Счетчик для нумерации сносок. В стилях документов `book` и `report` устанавливается в нуль в начале каждой главы, а в стиле документа `article` устанавливается в нуль только в начале документа. Увеличивается автоматически командами `\footnote` и `\footnotetext` перед их использованием.

`\footnote[i]{text}` Может использоваться либо в абзацной моде, либо внутри командных скобок `minipage` для получения сносок. В абзацной моде, если необязательный числовой аргумент *i* отсутствует, счетчик `footnote` увеличивается и используется как номер сноски. Он появляется в виде верхнего индекса в том месте тела страницы, где встретилась команда `\footnote`, а также в основании страницы, где печатается *text*. Если *i* — которое должно быть положительным целым числом — присутствует, то счетчик `footnote` остается неизменным, а *i* используется в качестве числовой метки сноски¹⁹⁹⁵. Внутри командных скобок `minipage` сноски помечаются строчными буквами и помещаются не внизу страницы, а под текстом, генерируемым этими командными скобками (см. пример в описании командных скобок `minipage`). В этом случае также, если присутствует аргумент *i* (который должен быть положительным целым числом), то в качестве метки сноски будет использоваться соответствующая строчная буква. Заметим, что команда отличается от `\footnote plain` TeX'a. Хрупкая.

`\footnotemark[i]` Используется в любой моде для получения метки сноски, причем метка появляется в теле страницы в виде верхнего индекса, и больше нигде. Если *i* отсутствует, то счетчик `footnote` увеличивается на единицу и это значение используется как метка сноски. Если же *i*, которое должно быть положительным целым числом, присутствует, то счетчик `footnote` остается неизменным, а в качестве метки сноски используется *i*. Команда обычно используется вместе с командой `\footnotetext`. Хрупкая.

¹⁹⁹⁵Сноска, которую Вы сейчас читаете, была получена командой `\footnote[1995]{Сноска, которую Вы сейчас читаете, ...}`.

Результат	Команды
Жили-были старик ...	<code>\tiny\temp</code>
Жили-были старик ...	<code>\scriptsize\temp</code>
Жили-были старик ...	<code>\footnotesize\temp</code>
Жили-были старик ...	<code>\small\temp</code>
Жили-были старик ...	<code>\normalsize\temp</code>
Жили-были старик ...	<code>\large\temp</code>
Жили-были старик ...	<code>\Large\temp</code>
Жили-были старик ...	<code>\LARGE\temp</code>
Жили-были старик ...	<code>\huge\temp</code>
Жили-были старик ...	<code>\Huge\temp</code>

Рис. 14. Размеры шрифтов.

`\footnoterule` Обычно сноска на странице отделяется от основного текста горизонтальной чертой протяженностью в треть ширины страницы. Это обеспечивает команда `\footnoterule`. Ее определение в \LaTeX е выглядит следующим образом:

```
\def\footnoterule{\kern -3pt
\hrule width .4\columnwidth \kern 2.6pt}
```

Это отличается от соответствующего определения в `plain TeX`. Ширину (`width`) этой черты можно легко изменять.

`\footnotesep` Параметр жесткой длины, значение которого минус текущее значение пробела, разделяющего базовые линии, дает величину пробела между сносками, а также сразу под горизонтальной чертой, генерируемой `\footnoterule`. Прочная команда, перед которой не надо ставить `\protect`.

`\footnotesize` Изменяет размер шрифта, а в этом размере выбирает романский шрифт. В математической моде использовать нельзя. Обычно выбирается размер, чуть больший, чем `\scriptsize` и чуть меньший, чем `\small`. См. рис.14 на стр.62. Хрупкая.

`\footnotetext [i]{text}` Может использоваться в любой моде. Ведет себя так же, как и `\footnote`, за тем исключением, что в теле страницы, на которой имеется сноска, метка сноски не появляется. Обычно используется с командой `\footnotemark`. Хрупкая.

`\footskip` Параметр жесткой длины, действующий на внешний вид каждой страницы результата. Равен расстоянию между нижней границей тела страницы

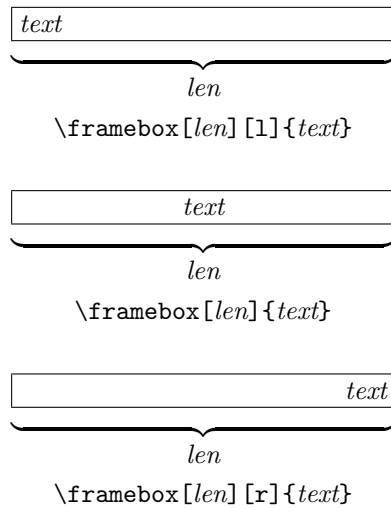


Рис. 15. Действие аргумента *pos* в команде `\framebox`.

и нижней границей ее основания. См. рис.23 на стр.137. Прочная команда, поэтому перед ней никогда не надо ставить `\protect`.

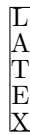
`\forall` Производит ординарный символ \forall , но только в математической моде. См. `\exists`. Прочная.

`\frac{top}{bot}` Команда, которую можно использовать только в математической моде для получения дроби с *top* над дробной чертой и *bot* — под ней:

$$\frac{n!}{i!(n-1)!} \dots \frac{n!}{i!(n-1)!}$$

Прочная.

`\frame{объект рисунка}` Можно использовать только внутри командных скобок `picture`. Помещает вокруг аргумента *объект рисунка* рамку, составленную из вертикальных и горизонтальных прямых, не отделяя ее пробелами. Например, внутри командных скобок `picture` команда `\put(0,0){\frame{\shortstack{L\\A\\T\\E\\X}}` дает следующий результат:



Точкой привязки является нижний левый угол результирующего бокса. Хрупкая.

`\framebox[len][pos]{text}` Вне командных скобок `picture` обрабатывает аргумент *text* в LR моде, а затем “обрамляет” результат, как показано на рис.15, стр. 63. Хрупкая.

`\framebox(i, j)[pos]{объект рисунка}` В этой форме команда может встретиться только внутри командных скобок `picture`. Параметры имеют то же самое значение, что и в описанной ниже команде `\makebox`. Единственное отличие этой команды от `\makebox` — то, что она производит текст в “рамке” из вертикальных и горизонтальных прямых. Действие этой команды показано на рис.12, стр.45.⁷ Хрупкая.

`\frenchspacing` Равномерно распределяет пробелы по всей строке. Чтобы понять эту макрокоманду, нужно знать, что в англо-американской полиграфии после точки, вопросительного и восклицательного знаков ставят двойной пробел. С другой стороны, после запятой и точки с запятой они вообще не используют пробелов. Французы, напротив, предпочитают, чтобы все пробелы были примерно одинаковыми. Следовательно, Вы должны выбирать (`\frenchspacing` или `\nonfrenchspacing`?) в зависимости от того, на каком языке вы пишете. В русском языке обычно используется `\nonfrenchspacing` (к тому же принятый в формате `plain` по умолчанию), а пробелы после запятых добавляются вручную. Хрупкая.

`\frown` Производит символ бинарного отношения \frown , но только в математической моде. Прочная.

`\fussy` Влияет на разбиение строк. Под ее действием — а по умолчанию это так — текст иногда может залезать на правое поле, причем об этом на терминал будет выдаваться сообщение. Если она действует в конце абзаца, то действует и на весь абзац.

g

`\gamma` Производит ординарный символ γ , но только в математической моде. Прочная.

`\Gamma` Производит ординарный символ Γ , но только в математической моде. Прочная.

`\gcd` Производит обозначение “наибольший общий делитель” (`gcd`), но только в математической моде. В выключенном стиле нижние и верхние индексы будут печататься под и над этим обозначением — в виде пределов.

⁷Заметим, что внутри командных скобок `picture` параметр длины `\fboxsep` не действует, так что между получаемым боксом и рамкой никакого места не оставляется.

`\ge` (*greater or equal*) Для получения \geq . Только в математической моде. Можно также использовать `\geq`. Отношение \leq задается командами `\le` (*lower or equal* — меньше либо равно) или `\leq`. Знаки строго неравенства ‘<’ и ‘>’ есть на клавиатуре. Прочная.

`\geq` Производит символ бинарного отношения \geq , но только в математической моде. См. `\ge`. Прочная.

`\gets` Производит символ бинарного отношения \leftarrow , но только в математической моде. Действует так же, как команда `\leftarrow`. Прочная.

`\gg` Производит символ бинарного отношения \gg , но только в математической моде. Противоположное отношение \ll задается `\ll`. Не путать с угловыми скобками `\langle` и `\rangle` (более “открытыми”):

```
\ll\kern 1cm\gg$ .....<< >>
\langle\!\langle\kern 1cm\rangle\!\rangle$ .....<< >>
```

Здесь для задания промежутка величиной 1 см использована команда примитивного Т_ЕXа `\kern`. Прочная.

`.glo` Расширение имени файла, который записывается, если во входном файле имеется команда `\makeglossary` (и не имеется команды `\nofiles`). Файл содержит команды `\glossaryentry{text}{p}` для каждой команды `\glossary{text}`, которая встречается во входном файле (и ничего более).

`\global` Если команду поместить перед командой `\def` или перед присваиванием, то декларация становится глобальной, то есть, ее значение выходит за рамки фигурных скобок, задающих ее область действия. Команда примитивного Т_ЕX’а.

`\glossary{text}` Приводит к тому, что в файл с расширением `glo` (если он пишется) записывается команда `\glossaryentry{text}{p}`. Аргумент `text` может включать в себя любые символы (но все фигурные скобки, включая `\{` и `\}`, должны быть парными) за исключением случая, когда команда `\glossary` встречается в аргументе другой команды — тогда `text` может состоять только из букв, цифр и знаков пунктуации.

`\glossaryentry{text}{p}` Может встретиться только в `glo`-файле, куда записывается Л^AT_ЕX’ом автоматически, когда во входном файле встречается команда `\glossary{text}` (только если в преамбуле входного файла имеется команда `\makeglossary` и нет команды `\nofiles`). Число `p` — это номер страницы выходного результата, где встретилась команда `\glossary{text}`.

`\grave` Производит акцент, но только в математической моде.

```
\grave a$, \grave +$, \grave X$ .....à, ÷, X̂
Прочная.
```

h

`\Hchar` или `\H{char}` Производит длинный венгерский умлаут (ó) над одним следующим символом *char* в LR или абзацной моде. Прочная.

`\hat` Производит акцент в математической моде. Ставит над следующим за командой символом знак “шляпка” (в математической моде). Чтобы нарисовать “шляпку” не в математической моде, используйте команду `\^`.

`$$\hat x$`, `$$\hat +$`, `$$\hat q$` \hat{x} , $\hat{+}$, \hat{q}

Прочная.

`\hbar` Производит ординарный символ \hbar , но только в математической моде. Прочная.

`\hbox{text}` Делает горизонтальный бокс; компоненты аргумента *text* помещаются горизонтально один за другим. Говоря по-простому, любой объект TeX’a можно представить в виде бусины с горизонтальным отверстием. Это отверстие может быть сдвинуто *вниз* (случай `\vbox`), находиться в *центре* (случай `\vcenter`) или *наверху* (случай `\vtop`). Разместить несколько боксов в одном `\hbox` означает пропустить через все отверстия одну нить. Команда примитивного TeX’a.

`\headheight` Параметр жесткой длины, действующий на внешний вид каждой страницы выходного результата. Его значение — это высота бокса, содержащего текст заголовка страницы. См. рис.23 на стр.137. Прочная команда, перед которой никогда не надо помещать `\protect`.

`headings` Опция стиля страницы в декларациях `\pagestyle` и `\thispagestyle`. Информация, которая помещается в заголовке страницы, определяется стилем документа (обычно это заголовок единицы рубрикации), а основание страницы оставляется пустым. Такой стиль страницы по умолчанию принят для стиля документа `book` и демонстрируется на этой странице (при отсутствии заголовков единиц рубрикации). Подробнее см. в описании стилей `book` и `report`.

`\headsep` Параметр жесткой длины, действующий на внешний вид страницы выходного результата. Он равен расстоянию между нижней границей заголовка страницы и верхней границей ее тела. См. рис.23 на стр.137. Прочная команда, поэтому перед ней никогда не надо ставить `\protect`.

`\heartsuit` Производит ординарный символ \heartsuit , но только в математической моде. Прочная.

`\hfill` Просто сокращение для `\hspace{\fill}`. Прочная.

- `\hline` Доступна только внутри командных скобок `array`, `tabular` и `tabular*`. Производит горизонтальную черту через всю ширину результирующей таблицы и должна располагаться или после команды, которая открывает командные скобки, или после команды `\\`. Примеры этой команды можно найти в описаниях команд `\cline` и `\multicolumn`, а также командных скобок `tabular`.
- `\hom` Производит обозначение (`hom`), но только в математической моде. Нижний и верхний индексы (если они есть) никогда не располагаются под и над ним в виде пределов. Прочная.
- `\hookleftarrow` Производит символ бинарного отношения \leftarrow , но только в математической моде. Прочная.
- `\hookrightarrow` Производит символ бинарного отношения \rightarrow , но только в математической моде. Прочная.
- `howpublished` (Поле имени ВивТ_ЭX'a.) Является необязательным для двух видов публикации и указывает, как был издан обсуждаемый источник. Текст должен начинаться с прописной буквы.
- `\hrulefill` Производит горизонтальную черту, заполняющую все отведенное ей место. Как это делается, показано на рис.13, стр.52. Прочная.
- `\hspace{len}` Производит горизонтальный пробел величиной `len`, который исчезает, если на нем происходит разрыв строки. Прочная.
- `\hspace*{len}` Производит горизонтальный пробел величиной `len`, который не исчезает, если на нем происходит разрыв строки. Прочная.
- `\huge` Декларация, которая изменяет размер шрифта и в этом размере выбирает **романский шрифт**. Ее нельзя использовать в математической моде. Обычно этот размер больше, чем `\LARGE` и меньше, чем `\Huge`. См. рис.14 на стр.62. Хрупкая.
- `\Huge` Декларация, которая изменяет размер шрифта и в этом размере выбирает **романский шрифт**. Ее нельзя использовать в математической моде. Обычно это максимально возможный размер шрифта, который крупнее, чем `\huge`. См. рис.14 на стр.62. Хрупкая.
- `\hyphenation{список слов}` Глобальная декларация, которая позволяет добавлять новое слово и способ его разбиения на слоги в специальную таблицу. Если Вы юрист и печатаете какой-нибудь труд по конституционному праву, Вы захотите научить Т_ЭX соответствующим переносам. Поместите в начале своего файла команду:
- ```
\hyphenation{кон-сти-ту-ци-он-ный кон-сти-ту-ция}
```

Таким способом Вы указываете, как правильно разбивать эти слова для переноса с одной строки на другую. Разумеется, этот список новых слов можно увеличивать. Прочная.

## i

- `\i` Производит бесточечную букву *i* в абзацной и LR модах. Это позволяет поставить над ним другие знаки (`\^i`, `\"i`). Прочная.
- `.idx` Расширение имени файла, который записывается, если в преамбуле входного файла встречается команда `\makeindex` (и нет команды `\nofiles`). Содержит только команды `\indexentry{text}{p}` для каждой команды `\index{text}` из входного файла (и ничего более).
- `\iff` (If and only if — тогда и только тогда.) Производит в математической моде символ  $\iff$ . Действует так же, как команды `\; \Longleftarrow`; то есть, дает символ  $\iff$  с дополнительными толстыми пробелами по обе стороны. Прочная.
- `\Im` Мнимая часть комплексного числа  $\Im$ . Только в математической моде. Правда, эта готическая буква  $\Im$  часто приводит к недоумению (кто в наше время знаком с готическим шрифтом?). Можно помочь зрительному восприятию этой  $\Im$ , написав после нее букву ‘m’:  

$$\Im z, \Im m, z \dots \Im z, \Im z$$
 Обратите внимание на использование математического мини-пробела `\.`. Если Вы часто используете эту конструкцию, лучше сделать из нее одну макрокоманду. Пробелы будут автоматически вставляться при верстке страницы:  

$$\def\IM{\mathop{\Im m}\nolimits}$$

$$\IM z, \IM(z+z') \dots \Im z, \Im(z+z')$$
 Действительная часть задается с помощью `\Re`, при этом получается готическая  $\Re$ . И в этом случае  $\Re$  производит лучшее впечатление: теперь вам ничего другого не остается, как определить макрокоманду `\RE`. Обратите внимание на употребление (обязательное) заглавных букв, чтобы не смешивать новые макрокоманды с макрокомандами `\Im` и `\Re` формата plain. Прочная.
- `\imath` Производит ординарный символ *i*, но только в математической моде. Прочная.
- `in` Ключевое слово Т<sub>E</sub>X’а для единицы измерения дюйм (inch), которая удовлетворяет равенству: 1 in=72.27 pt=25.4 mm.

`\in` Производит символ бинарного отношения  $\in$ , но только в математической моде. Прочная.

`\include{file}` Можно использовать в соединении с командой `\includeonly{file-list}` для получения только части большого документа, содержимое которого разделено на несколько входных файлов. Заметим, что результат обработки файла *file* — если вообще появится — всегда начинается с новой страницы, и он будет оборван, как если бы в конце файла *file.tex* стояла команда `\clearpage`. Хрупкая.

`\includeonly{file-list}` Может встретиться только в преамбуле входного файла. Аргумент *file-list* представляет собой список из нуля или более имен файлов без расширений. Если в списке более одного файла, то их имена разделяются запятыми (под *file* имеется в виду *file.tex*). Если в *file-list* встречается *file*, то команда `\include{file}` не должна находиться в теле входного файла. В выходном результате, получаемом ЛАТЭХ'ом, появится текст только из файлов, содержащихся в списке файлов *file-list*. Хрупкая.

`\indent` Приводит к тому, что следующая строка результирующего документа, который получается ЛАТЭХ'ом, печатается с отступом, равным текущему значению параметра длины `\parindent`. Прочная.

`\index{text}` Приводит к тому, что в файл с расширением `idx` (если он записывается) пишется команда `\indexentry{text}{p}`. Аргумент *text* может включать в себя любые символы (но все фигурные скобки, включая `\{` и `\}`, должны быть парными), за исключением случая, когда команда `\index` входит в аргумент другой команды. Тогда *text* может состоять только из букв, цифр и знаков пунктуации. Хрупкая.

`\indexentry{text}{p}` Может встретиться только в `idx`-файле. ЛАТЭХ записывает ее туда автоматически, когда встречает во входном файле команду `\index{text}`. (Это происходит, только если в преамбуле входного файла имеется команда `\makeindex` и нет команды `\nofiles`.) Число *p* указывает номер той страницы, на которую в выходном результате попадает команда `\index{text}`.

`\indexspace` Команда, которая используется внутри командных скобок `theindex` для получения дополнительного вертикального пробела обычно для отделения элементов, начинающихся с различных букв алфавита.

`\inf` Производит обозначение (inf), но только в математической моде:

$$\begin{aligned} \$m=\inf_{t\in T}t\$ & \dots\dots\dots m = \inf_{t\in T} t \\ \$\displaystyle m=\inf_{t\in T}t\$ & \dots\dots\dots m = \inf_{t\in T} t \end{aligned}$$

В выключенном стиле нижний и верхний индексы размещаются под и над ним в виде пределов. Можно изменять поведение индексов в `\textstyle` и `\displaystyle`, задавая команды `\inf\limits_{t\in T}` или `\inf\nolimits_{t\in T}`. Прочная.

`\infty` Производит ординарный символ  $\infty$ , но только в математической моде. Прочная.

`\input{file}` Действует так, как если бы в выходной файл было включено содержимое файла *file*. Если *file* не имеет расширения, то включается *file.tex*. Фигурные скобки необязательны.

`institution` (Имя поля ВивTeX'a.) Содержит название организации — например, “Programming Research Group” — в которой выпущен представленный технический отчет.

`\int` Производит символ большого оператора  $\int$  и  $\int$ , но только в математической моде. В выключенных формулах этот символ больше, чем в формулах в тексте. Кроме того, в выключенных формулах нижние и верхние индексы печатаются под и над ним в виде пределов. Знаки `_` (индекс) и `^` (показатель степени) служат для обозначения пределов.

$$\text{\$}\displaystyle\Gamma(z)=\int_0^{\infty} e^{-t}t^{z-1}dt\text{\$} \dots\dots\dots \\ \dots\dots\dots\Gamma(z) = \int_0^{\infty} e^{-t}t^{z-1}dt$$

Если Вам тесно на строке, используйте конструкцию `\int\limits`, которая позволяет “индексам” размещаться над и под знаком интеграла:

$$\text{\$}\displaystyle\Gamma(z)=\int\limits_0^{\infty} e^{-t}t^{z-1}dt\text{\$} \dots\dots\dots\Gamma(z) = \int_0^{\infty} e^{-t}t^{z-1}dt$$

Прочная.

`\intertextsep` Параметр эластичной длины, равный величине вертикального пробела, помещаемого над и под плавающей вставкой, если она размещается в середине текстовой страницы (когда выбрана опция *h*). Прочная команда, поэтому перед ней не надо ставить `\protect`.

`\iota` Производит ординарный символ  $\iota$ , но только в математической моде. Для того, чтобы получить бесточечную курсивную букву  $\iota$  — нужную, например, если ее надо использовать с акцентом — применяйте команду `\imath`. Прочная.

`\it` Изменяет шрифт на *курсив* — см. рис.9 на стр.28. Прочная.

`\item[text]` Встречается внутри командных скобок `enumerate`, `description`, `list`, `itemize`, `theindex` и `trivlist` (см. описание и примеры соответствующих командных скобок). Хотя ее действие слегка и различается в этих командных скобках, она всегда указывает на начало новой порции информации. Отличается от команды `plain` TeX'a с тем же именем. Хрупкая.

`\itemindent` Команда жесткой длины, действующая на внешний вид командных скобок `list`, как это показано на рис.18, стр.81. Прочная команда, перед которой не надо ставить `\protect`.

`\itemize` Командные скобки, которые используются для получения помеченных списков, метки пунктов которых не меняются при переходе от одной команды `\item` к другой. Например, список

- Вычислительная математика — это наука об алгоритмах.
- Язык программирования — это способ записи алгоритмов.
- Изучение языков программирования основано на изучении логических систем.

был получен следующими командами:

```
\begin{itemize}
\item
Вычислительная математика --- это наука об алгоритмах.
\item
Язык программирования --- это способ записи алгоритмов.
\item
Изучение языков программирования основано на
изучении логических систем.
\end{itemize}
```

Допускается только четыре уровня вложенности. На этих уровнях метками, принятыми по умолчанию, являются символы `•` (`\bullet`), `—` (`\bf --`), `*` (`\ast`) и `·` (`\cdot`). Эти символы “хранятся” в четырех командах: `\labelitemi`, `\labelitemii`, `\labelitemiii` и `\labelitemiv`, соответственно. Так, например, если Вы хотите в качестве метки вместо `•` использовать `♠`, Вам надо переопределить команду `\labelitemi` одной из следующих эквивалентных деклараций:

```
\def\labelitemi{\spadesuit}
\renewcommand{\labelitemi}{\spadesuit}
```

- ♠ Вычислительная математика — это наука об алгоритмах.
- ♠ Язык программирования — это способ записи алгоритмов.
- ♠ Изучение языков программирования основано на изучении логических систем.

Можно получать различные метки и на одном уровне вложенности. Для этого надо внутри командных скобок `\itemize` задать команде `\item` дополнительный аргумент. Например, следующий список:

- ♥ Вычислительная математика — это наука об алгоритмах.
- ◇ Язык программирования — это способ записи алгоритмов.
- ♣ Изучения языков программирования основано на изучении логических систем.

был получен такими командами:

```

\begin{itemize}
\item[{\heartsuit}]
Вычислительная математика --- это наука об алгоритмах.
\item[{\diamondsuit}]
Язык программирования --- это способ записи алгоритмов.
\item[{\clubsuit}]
Изучение языков программирования основан на изучении
логических систем.
\end{itemize}

```

`\itemsep` Параметр эластичной длины, действующий на внешний вид командных скобок `list`, как это показано на рис.18, стр.81. Прочная команда, перед которой не надо ставить `\protect`.

## j

`\j` Производит бесточечную букву j в абзацной и LR модах. Прочная.

`\jmath` Производит ординарный символ  $j$ , но только в математической моде. Прочная.

`\jobname` Дает базовое имя основному обрабатываемому входному файлу.

`\Join` Производит символ бинарного отношения  $\bowtie$ , но только в математической моде. В plain TeX'e такой команды нет. Прочная.

`\jot` Параметр жесткой длины, равный величине дополнительного пробела, который вставляется между рядами в командных скобках `eqnarray` и `eqnarray*`.

`journal` (Имя поля BibTeX'a.) Название журнала, в котором опубликована упоминаемая статья. В Вашей системе могут использоваться различные сокращения — проконсультируйтесь у специалистов.

## k



`\kappa` Производит греческую букву  $\kappa$ , но только в математической моде. Прочная.

`\ker` Производит символ (ker), но только в математической моде:

для  $\$x \in \ker u\$$  справедливо  $\$u(x)=0\$$  .....  
..... для  $x \in \ker u$  справедливо  $u(x) = 0$   
Если используется с нижним и верхним индексом, они никогда не располагаются под и над этим символом в виде пределов.

`\kern` Команда примитивного Т<sub>E</sub>Xа, которая дает керн, т.е. неизменяемый пробел, который нельзя ни сжать, ни растянуть. Пример приводится при описании команды `\gg`. В математической моде имеется команда `\mkern`.

`key` (Имя поля В<sub>I</sub>T<sub>E</sub>X'а.) Не путать с *key*, который используется в команде `\cite`! Используется для сортировки элементов, когда отсутствуют все другие поля для сортировки.

`\kill` Может встретиться только внутри командных скобок `tabbing`, где обозначает, что ряд, который она заканчивает, не появляется в выходном результате, но значение всех позиций табуляции запоминается. Основная идея показана на следующем простом примере:

|        |                                          |
|--------|------------------------------------------|
|        | <code>\begin{tabbing}</code>             |
|        | <code>123\=456\=789\=\kill</code>        |
| один   | <code>\&gt;    один  \\\</code>          |
| два    | <code>\&gt;    два   \\\</code>          |
| три    | <code>\&gt;\&gt;   три  \\\</code>       |
| четыре | <code>\&gt;\&gt;   четыре \\\</code>     |
| пять   | <code>\&gt;\&gt;\&gt;  пять  \\\</code>  |
| шесть  | <code>\&gt;\&gt;\&gt;  шесть  \\\</code> |
| семь   | <code>\&gt;\&gt;\&gt;  семь   \\\</code> |
| восемь | <code>\&gt;\&gt;\&gt;  восемь \\\</code> |
| девять | <code>\&gt;    девять \\\</code>         |
| десять | <code>\&gt;    десять</code>             |
|        | <code>\end{tabbing}</code>               |

Здесь первая строка внутри командных скобок `tabbing` устанавливает позиции табуляции — так же, как позиции табуляции на пишущей машинке — а команда `\kill` просто говорит L<sup>A</sup>T<sub>E</sub>X'у не производить никакого выходного результата, соответствующего этой строке. Команды `\=` устанавливают позиции табуляции. Заметим, что вне командных скобок `tabbing` команда `\=` делает акцент макрон над следующим за ней символом. Если Вам нужен акцент макрон внутри командных скобок `tabbing`, следует использовать команду `\a=`. Также заметим, что в шрифтах Computer modern, разработанных Кнудом, все цифры имеют ширину, равную половине em. Команда

`\>` перемещает начало следующего за ней текста на очередную позицию табуляции. Команда `\kill` также задает *next-tab-stop* равным *left-margin-tab* и начинает новую строку.

## l

`\l` Доступна только в абзацной и LR модах для получения польской строчной перечеркнутой l (l̄). Прочная.

`\L` Доступна только в абзацной и LR модах для получения польской прописной перечеркнутой L (L̄). Прочная.

`\label{key}` Используется для организации перекрестных ссылок. Связывает *key* — который может состоять только из букв, цифр и знаков пунктуации — с текущим значением `\ref` и `\pageref`. Например, команда `рис.\ref{taleout}` на стр.`\pageref{taleout}` производит ссылку “рис.4 на стр.18”, поскольку, как это видно из `рис.3` на стр.17, при получении этого рисунка использовалась команда `\label{taleout}`. Хрупкая, но перед ней, когда она встречается в аргументах команд рубрикации или команды `\caption`, не надо ставить `\protect`.

`\labelitemi` Содержит символ метки пункта внешнего или самого высокого уровня вложенности командных скобок `itemize`. По умолчанию в стилях документа `article`, `report`, `book` и `letter` он равен `•` (`\bullet`). “Значение” можно изменить декларациями `\def` или `\renewcommand`. См. описание `itemize`.

`\labelitemii` Содержит символ метки пункта командных скобок `itemize`, которые встречаются внутри командных скобок `itemize`. По умолчанию в стилях документа `article`, `report`, `book` и `letter` используется символ `–` (`\bf –`). “Значение” можно изменить декларациями `\def` или `\renewcommand`. См. описание `itemize`.

`\labelitemiii` Содержит символ метки пункта командных скобок `itemize`, которые содержатся внутри областей действия двух других командных скобок `itemize`. По умолчанию в стилях документа `article`, `report`, `book` и `letter` используется символ `*` (`\ast`). “Значение” можно изменить декларациями `\def` или `\renewcommand`. См. описание `itemize`.

`\labelitemiv` Содержит символ метки пункта командных скобок `itemize`, которые содержатся внутри областей действия трех других командных скобок `itemize`. По умолчанию в стилях документа `article`, `report`, `book` и `letter` используется символ `·` (`\cdot`). “Значение” можно изменить декларациями `\def` или `\renewcommand`. См. описание `itemize`.

`\labelsep` Параметр жесткой длины, действующий на внешний вид командных скобок `list`, как это показано на рис.18, стр.81. Прочная команда, перед которой не надо ставить `\protect`.

`\lambda` Производит ординарный символ  $\lambda$ , но только в математической моде. Прочная.

`\Lambda` Производит ординарный символ  $\Lambda$ , но только в математической моде. Прочная.

`\land` (logical and — логическое “и”) Производит символ бинарной операции  $\wedge$ , но только в математической моде. Служит для задания векторного произведения:

```
\def\#1{\overrightarrow{\#1}} $(\u\land\v)\land\w=
(\v\cdot\w)\, \u-(\u\cdot\w)\, \v$
```

$$(\vec{u} \wedge \vec{v}) \wedge \vec{w} = (\vec{v} \cdot \vec{w}) \vec{u} - (\vec{u} \cdot \vec{w}) \vec{v}$$

Прочная.

`\langle` Производит открывающий символ  $\langle$ , но только в математической моде. Закрывающая угловая скобка задается командой `\rangle`. Чтобы поставить по две скобки рядом, пишем `\langle\langle` и `\rangle\rangle`.

$$|\langle\langle A, B \rangle\rangle|^2 \leq |\langle\langle A, A^+ \rangle\rangle| \times |\langle\langle B^+, B \rangle\rangle|$$

Не путайте угловую скобку  $\langle$  с более “острым” знаком неравенства ‘ $<$ ’. Прочная.

`\large` Изменяет размер шрифта и в этом размере задает романский шрифт. Нельзя использовать в математической моде. Выбранный размер больше, чем `\normalsize` и меньше, чем `\Large`. См. рис.14 на стр.62. Хрупкая.

`\Large` Изменяет размер шрифта и в этом размере задает романский шрифт. Нельзя использовать в математической моде. Выбранный шрифт больше, чем `\large`, но меньше, чем `\LARGE`. См. рис.14 на стр.62. Хрупкая.

`\LARGE` Изменяет размер шрифта и в этом размере задает **романский шрифт**. Нельзя использовать в математической моде. Выбранный шрифт больше, чем `\Large`, но меньше, чем `\huge`. См. рис.14 на стр.62. Хрупкая.

`\LaTeX` Производит логограмму  $\text{\LaTeX}$ . Эта логограмма “приклеивается” к следующему слову, поэтому пробел после нее получается так: `\LaTeX`, `\LaTeX{}` или `\LaTeX\`.

`\lbrace` Известно, что фигурные скобки ‘{’ и ‘}’ зарезервированы Т<sub>Е</sub>X’ом для специальных целей. Для получения открывающей скобки, задайте `$$\lbrace$`, или более просто, `$$\{`. Закрывающая скобка получается аналогично: `$$\rbrace$` или `$$\}`. Нельзя задавать фигурную скобку вне математической моды. Величина фигурной скобки может меняться с помощью команд plain Т<sub>Е</sub>Xа серии `\big` или конструкции `\left... \right`. Например, в результате `$$E=\bigr\{x\mid P(x)\bigr\}$$` получится:

$$E = \{x \mid P(x)\}$$

См. также рис.5 на стр.21 и рис.6 на стр.22. Прочная.

`\lceil` Производит открывающий символ  $\lceil$ , но только в математической моде. Величина может меняться, как и у `\lbrace`:

`$$\lceil x\rceil$` --- это наименьшее целое  $n \geq x$  .....  
 .....  $\lceil x \rceil$  — это наименьшее целое  $n \geq x$

Прочная.

`\ldots` Во всех модах производит многоточие (...), состоящие из трех “низких” точек.

`$$x=(x_1,\ldots,x_n)$` .....  $x = (x_1, \dots, x_n)$

`$$x=(x_1, \dots, x_n)$` .....  $x = (x_1, \dots, x_n)$

Разница заметна, не так ли? Если при наборе математики нужны *отцентрированные* точки (т.е., на высоте знака ‘=’ или ‘+’, как, например, в формуле  $S_n = x_1 + \dots + x_n$ ), используйте `\cdots`. Прочная.

`\le` Производит символ бинарного отношения  $\leq$ , но только в математической моде. Действует так же, как команда `\leq`. Прочная.

`\leadsto` Производит символ бинарного отношения  $\leadsto$ , но только в математической моде. В plain Т<sub>Е</sub>X’е такой команды нет. Прочная.

`\leftdelim1 form \rightdelim2` Конструкция разрешена только в математической моде и в выключенном стиле. Сначала обрабатывается формула *form*, а затем выбираются ограничители подходящего размера. Обе команды `\left` и `\right` обязательны, но если в выходном результате какой-либо из ограничителей не нужен, вместо соответствующего *delim<sub>1</sub>* или *delim<sub>2</sub>* следует поставить точку. Примеры использования можно найти на рис.5, стр.21 и на рис.6, стр.22, а также в описании команды `\downarrow`. Как `\left`, так и `\right` являются прочными.

`\leftarrow` Производит символ бинарного отношения  $\leftarrow$ , но только в математической моде. Такой же символ можно получить командой `\gets`. Еще более длинную стрелку можно получить с помощью команды `\longleftarrow`. Правый вариант получается через `\rightarrow` и `\longrightarrow`. Вертикальные стрелки получаются командами `\uparrow` и `\downarrow`. Прочная.

- `\Leftarrow` Производит символ бинарного отношения  $\Leftarrow$ , но только в математической моде. Более длинная стрелка получается с помощью `\Longleftarrow`. Правые варианты этих скобок получаются ... впрочем, сообразите сами! Прочная.
- `\leftarrowfill` Производит направленную влево стрелку, которая заполняет все отведенное ей место. Графически это показано на рис.13, стр.52.
- `\lefteqn{form}` Может встретиться только внутри командных скобок `eqnarray` или `eqnarray*`. Аргумент `form` обрабатывается в математической моде и в выключенном стиле, но при этом Т<sub>E</sub>X считает, что она имеет нулевую ширину. Используется для разбиения длинных формул, которые не помещаются на одной строке.
- `\leftharpoondown` Производит символ бинарного отношения  $\leftharpoondown$ , но только в математической моде. Имеется также и команда `\rightharpoondown`, которая рисует  $\rightharpoondown$ . Прочная.
- `\leftharpoonup` Производит символ бинарного отношения  $\leftharpoonup$ , но только в математической моде. Имеется также `\rightharpoonup`, которая рисует  $\rightharpoonup$ . Прочная.
- `\leftmargin` Параметр жесткой длины, действующий на горизонтальное расстояние между левым полем охватывающих командных скобок и левым полем текущих командных скобок `list`, как это показано на рис.18, стр.81. Если командные скобки `list` не находятся в области действия других командных скобок `list`, то по умолчанию параметр `\leftmargin` равен `\leftmargini`; если командные скобки `list` находятся внутри областей действия других командных скобок `list`, то по умолчанию параметр `\leftmargin` равен `\leftmarginii`; ... ; если командные скобки `list` находятся внутри областей действия пяти других командных скобок `list`, то по умолчанию параметр `\leftmargin` равен `\leftmarginiv`; более глубокая вложенность не разрешается. Прочная команда, перед которой не надо ставить `\protect`.
- `\leftmargini` Параметр жесткой длины для задания величины некоторых левых полей, например, в командных скобках `list`. См. команду `\leftmargin`. Прочная команда, поэтому перед ней не надо ставить `\protect`.
- `\leftmarginii` Параметр жесткой длины для задания величины некоторых левых полей, например, в командных скобках `list`. См. команду `\leftmargin`. Прочная команда, поэтому перед ней не надо ставить `\protect`.
- `\leftmarginiii` Параметр жесткой длины для задания ширины некоторых левых полей, например, в командных скобках `list`. См. команду `\leftmargin`. Прочная команда, поэтому перед ней не надо ставить `\protect`.
- `\leftmarginiv` Параметр жесткой длины для задания ширины некоторых левых полей, например, в командных скобках `list`. См. команду `\leftmargin`. Прочная команда, поэтому перед ней не надо ставить `\protect`.



```

\documentstyle[11pt]{letter}
\address{27 Hudson Street,\
 Athens,\
 Wessex,\
 A26 7YU.}
\signature{Джеймс Холмс}
\begin{document}
%
\begin{letter}
\begin{letter}{Dr Albert Grovenor,\
 ‘‘Appleblossom’’,\
 Whittington Green,\
 Hampshire.}
\opening{Дорогой Альберт!}
Спасибо за Ваше письмо от 16 мая 1991
и приложенный образец. Как Вы и просили, я провел полный
анализ, но результаты оказались настолько экстраординарными,
что я могу передать их Вам только лично.
\closing{Искренне Ваш,}
\ps{PS: Я буду в отпуске до 22-го.}
\cc{Миссис Дикенс \ \ Проф.~Купер}
\end{letter}
%
\end{document}

```

Рис. 16. Пример использования командных скобок `letter`.

Заметим, что здесь  $text_1$  является подвижным аргументом, поэтому перед любой хрупкой командой, содержащейся в нем, надо ставить `\protect`. Этот аргумент используется для задания адреса получателя письма. Внутри аргумента  $text_1$  могут встретиться команды `\`, которые приказывают начинать новую строку. Заметим, что  $text_1$  — обязательный аргумент. Его пропуск приведет к сообщению об ошибке. Сами командные скобки `letter` являются подвижным аргументом.

`\lfloor` Производит открывающий символ  $\lfloor$ , но только в математической моде.

Пример:

```


$$\lfloor x \rfloor = E(x) \text{ (целая часть)}$$

$$\lfloor x \rfloor = E(x) \text{ (целая часть)}$$


```

Прочная.

`\lg` Производит обозначение элементарной функции логарифм (`lg`), но только в математической моде. Индексы у нее никогда не появляются в виде пределов. Прочная.

27 Hudson Street,  
Athens,  
Wessex,  
A26 7YY.

10 июля 1992 г.

Dr Albert Grovenor,  
“Appleblossom”,  
Whittington Green,  
Hemlock.

Дорогой Альберт!

Большое спасибо за Ваше письмо от 30 июня 1992 г. и  
приложенный образец. Как Вы и просили, я провел полный  
анализ, но результаты оказались настолько экстраординар-  
ными, что я могу передать их Вам только лично.

Искренне Ваш,

Джеймс Холмс

PS: Я буду в отпуске до 22-ого.  
cc: Миссис Дикенс  
Проф. Купер

Рис. 17. Пример письма, отформатированного Л<sup>A</sup>T<sub>E</sub>X'ом.



|          |                                                                                 |          |                             |
|----------|---------------------------------------------------------------------------------|----------|-----------------------------|
| <i>a</i> | <code>\topsep + \parskip</code> or <code>\topsep + \parskip + \partopsep</code> | <i>f</i> | <code>\leftmargin</code>    |
| <i>b</i> | <code>\itemsep + \parsep</code>                                                 | <i>g</i> | <code>\listparindent</code> |
| <i>c</i> | <code>\parsep</code>                                                            | <i>h</i> | <code>\rightmargin</code>   |
| <i>d</i> | <code>\labelwidth</code>                                                        | <i>i</i> | <code>\parindent</code>     |
| <i>e</i> | <code>\labelsep</code>                                                          |          |                             |

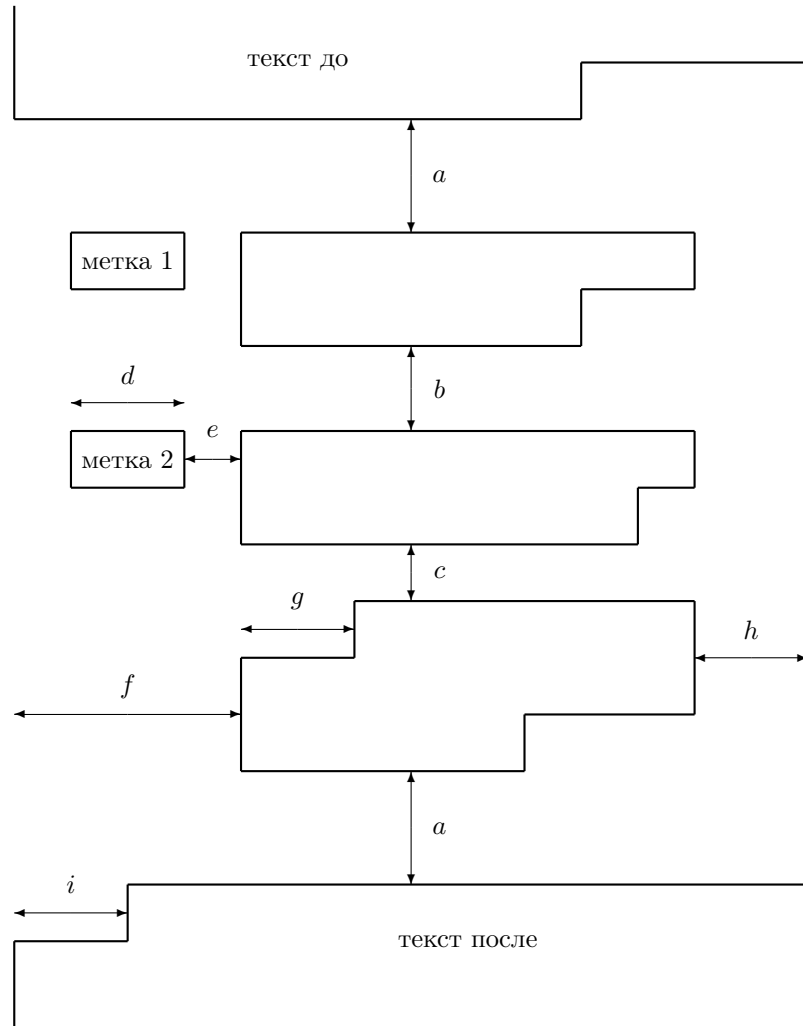


Рис. 18. Параметры командных скобок `list`.

`\lgroup` Доступна только в математической моде, где производит ограничитель, похожий на большую левую круглую скобку; однако перед ней обязательно должна стоять команда `\left` или `\right` (см. рис.5 на стр.21 и рис.6 на стр.22).

`\lhd` Производит символ бинарного оператора  $\triangleleft$ , но только в математической моде. Заметим, что в plain TeX'e такой команды нет. Прочная.

`\lim` Производит обозначение предела ( $\lim$ ), но только в математической моде. В выключенных формулах нижний и верхний индексы печатаются под и над ним в виде пределов.

`\lim_{x\rightarrow 0}{\sin x\over x}=1` .....  $\lim_{x\rightarrow 0} \frac{\sin x}{x} = 1$   
`\displaystyle \lim_{x\rightarrow 0}`

`{\sin x\over x}=1` .....  $\lim_{x\rightarrow 0} \frac{\sin x}{x} = 1$

Можно изменять поведение верхнего и нижнего индексов, используя варианты `\limits` или `\nolimits`. Прочная.

`\liminf` Производит обозначение ( $\liminf$ ), но только в математической моде. Если этот символ встречается в выключенных формулах, нижний и верхний индексы печатаются под и над ним в виде пределов.

`\liminf_{x\rightarrow 0}f(x)` .....  $\liminf_{x\rightarrow 0} f(x)$

Некоторые предпочитают оператор `\lim` с таким определением:

`\def\liminf{\mathop{\underline{\rm lim}}}`

`\liminf_{x\rightarrow 0}f(x)` .....  $\underline{\lim}_{x\rightarrow 0} f(x)$

`\displaystyle\liminf_{x\rightarrow 0}f(x)` .....  $\underline{\lim}_{x\rightarrow 0} f(x)$

Определение `\limsup` аналогично. Достаточно заменить `\underline` на `\overline` в предыдущем определении `\liminf`:

`\def\limsup{\mathop{\overline{\rm lim}}}`

`\limsup_{x\rightarrow 0}f(x)` .....  $\overline{\lim}_{x\rightarrow 0} f(x)$

`\displaystyle\limsup_{x\rightarrow 0}f(x)` .....  $\overline{\lim}_{x\rightarrow 0} f(x)$

Прочная.

`\limits` Команда `cmd\limits_{form_1}^{form_2}` в математической моде действует так, что формула  $form_1$  появляется в виде нижнего предела символа, производимого командой  $cmd$ , а формула  $form_2$  — в виде его верхнего предела. Должна сопровождаться оператором (макрокомандой, составленной с помощью `\mathop`). Только в математической моде. Введем определение:

`\def\Som{\mathop{\bf S}}`

Формула `\Som_{i=0}^{i=n}a_i` выглядит по-разному в зависимости от того, печатается она в тексте (между двумя простыми долларами, то есть в

`\textstyle`), или на отдельной строке (между двумя двойными долларами, в `\displaystyle`):

$$\begin{aligned} \$\text{\Som}_{i=0}^{i=n}a_i\$ & \dots\dots\dots \mathbf{S}_{i=0}^{i=n} a_i \\ \$\$ \text{\Som}_{i=0}^{i=n}a_i \$\$ & \dots\dots\dots \mathbf{S}_{i=0}^{i=n} a_i \end{aligned}$$

Поведение индексов зависит, следовательно, от используемого стиля. В `\textstyle` индексы помещаются сбоку от **S**, а в `\displaystyle` они располагаются над и под **S**.

Если макрокоманда `\limits` следует за оператором, индексы всегда размещаются над и под этим оператором, как в `\displaystyle`:

$$\begin{aligned} \def\Tom{\mathop{\bf T}\limits} \\ \$\text{\Tom}_{i=0}^{i=n}a_i\$ & \dots\dots\dots \mathbf{T}_{i=0}^{i=n} a_i \\ \$\displaystyle\text{\Tom}_{i=0}^{i=n}a_i\$ & \dots\dots\dots \mathbf{T}_{i=0}^{i=n} a_i \end{aligned}$$

Если определение оператора использует `\limits`, а Вы хотите нейтрализовать эту `\limits`, добавьте к этому оператору `\nolimits`:

$$\begin{aligned} \$\text{\Tom\nolimits}_{i=0}^{i=n}a_i\$ & \dots\dots\dots \mathbf{T}_{i=0}^{i=n} a_i \\ \$\displaystyle\text{\Tom\nolimits}_{i=0}^{i=n}a_i\$ & \dots\dots\dots \mathbf{T}_{i=0}^{i=n} a_i \end{aligned}$$

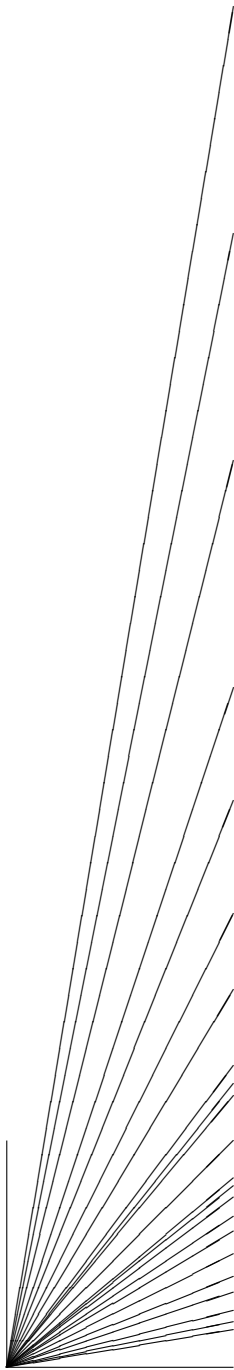
Если же определение оператора использует `\nolimits` (например, `\int`), можно нейтрализовать `\nolimits`, добавив к этому оператору `\limits`. (Примеры см. в `\int`). Команда примитивного `TeX` а.

`\limsup` Производит обозначение ( $\lim \sup$ ), но только в математической моде. В выключенных формулах нижний и верхний индексы печатаются под и над ним в виде пределов. См. `\liminf`. Прочная.

`\line` Может встретиться только в качестве аргумента команд `\put` и `\multiput` внутри командных скобок `picture`. Команда:

$$\text{\put}(i, j)\{\text{\line}(p, q)\{l\}$$

рисует отрезок прямой, который начинается в точке  $(i, j)$  и проекция которого на ось  $x$  равна  $l$  единиц. (Единственным исключением являются вертикальные отрезки, в этом случае  $l$  равно действительной длине получаемой линии.) Наклон прямой задается парой  $(p, q)$ , то есть,  $p$  единиц в  $x$ -направлении на каждые  $q$  единиц в  $y$ -направлении. Как  $p$ , так и  $q$  должны быть целыми числами между  $-6$  и  $+6$  включительно, без общего делителя. Полный набор наклонов прямых с положительными  $p$  и  $q$ , допустимых в `TeX`, показан на рис.19, стр.84. Заметим, что команда `TeX`'а `\line` не имеет ничего общего с командой `plain TeX`'а с таким же именем. Хрупкая.



```

\unitlength=1pt
\begin{picture}(165,900)(0,-10)
\put(0,0){\line(0,1){150}}
\put(0,0){\line(1,0){150}}
\put(0,0){\line(6,1){150}}
\put(0,0){\line(5,1){150}}
\put(0,0){\line(4,1){150}}
\put(0,0){\line(3,1){150}}
\put(0,0){\line(2,1){150}}
\put(0,0){\line(1,1){150}}
\put(0,0){\line(5,2){150}}
\put(0,0){\line(3,2){150}}
\put(0,0){\line(1,2){150}}
\put(0,0){\line(5,3){150}}
\put(0,0){\line(4,3){150}}
\put(0,0){\line(1,3){150}}
\put(0,0){\line(5,4){150}}
\put(0,0){\line(3,4){150}}
\put(0,0){\line(1,4){150}}
\put(0,0){\line(6,5){150}}
\put(0,0){\line(4,5){150}}
\put(0,0){\line(3,5){150}}
\put(0,0){\line(2,5){150}}
\put(0,0){\line(1,5){150}}
\put(0,0){\line(1,6){150}}
\put(0,0){\line(5,6){150}}
\end{picture}

```

Рис. 19. Наклоны прямых линий, допустимые в командных скобках `picture` — см. команду `\line`.

`\linebreak[i]` Необязательный числовой аргумент *i* может быть равен 0, 1, 2, 3 или 4. Если он отсутствует или равен 4, то команда `\linebreak` обозначает положение конца строки. Результат выравнивается справа — если только правое выравнивание не подавляется какими-нибудь другими командами или декларациями — причем слово непосредственно перед командой `\linebreak` оказывается в крайне правой позиции той строки, на которой оно оказалось. (Это может привести к предупреждающему сообщению о недозаполненном горизонтальном боксе.) Если числовой аргумент *i* равен 0, то `TeX` хотя и может окончить строку в этой точке, но присутствие команды не принуждает его к этому, правда, и не запрещает делать это. Если числовое значение *i* равно 1, 2 или 3, то это рекомендует `TeX`'у делать разрыв строки в этой точке, причем чем больше значение *i*, тем настойчивее рекомендация. Хрупкая.

`\linethickness{len}` Действует на толщину только вертикальных и горизонтальных прямых в командных скобках `picture` — делает ее равной *len*.

`\linewidth` Параметр жесткой длины — который нельзя изменять — равен текущей ширине строк. Изменяется, когда используются некоторые командные скобки (такие как `quotation`). Прочная команда, перед которой никогда не надо ставить `\protect`.

`list` Командные скобки широкого назначения для получения списков информации. Пункты списка могут быть помечены различными способами. Организацией и внешним видом списка управляют несколько параметров; см. рис.18 на стр.81. (Параметр длины `\parindent` также включен в эту схему, поскольку может влиять на выбор значений некоторых других параметров.) Синтаксис командных скобок `list`:

```
\begin{list}{text1}{dec-list} text2 \end{list}
```

где *text*<sub>1</sub> — то, что будет генерироваться командой `\item`, когда она не имеет необязательного аргумента, *dec-list* — последовательность присваиваний значений некоторым параметрам длины, которые показаны на рис.18, стр.81. Если какому-нибудь параметру длины не присваивается новое значение таким образом, оно задается присваиванием в одном из `\@listi`, ..., `\@listv` или `\@listvi` (выбор зависит от уровня вложенности), которые имелись перед *dec-list*. Далее, *text*<sub>2</sub> — информация, которая помещается в список; в ней могут встретиться одна или несколько команд `\item`. Заметим, что чтобы метки генерировались автоматически и в возрастающей последовательности, можно в *dec-list* поместить команду `\usecounter`.

`\listoffigures` Производит список рисунков в том месте, где она встретилась во входном файле. Чтобы получить правильный список рисунков, надо пропустить `LaTeX` как минимум дважды. Пишет или перезаписывает файл с расширением `lof` (если нет команды `\nofiles`).

`\listoftables` Производит список таблиц в том месте, где она встретила в исходном файле. Чтобы получить правильный список таблиц, надо пропустить `\listoftables` как минимум дважды. Пишет или перезаписывает файл с расширением `lot` (если нет команды `\nofiles`).

`\listparindent` Параметр жесткой длины, который действует на внешний вид командных скобок `list` (как это показано на рис.18, стр.81). Прочная команда, перед которой не надо ставить команду `\protect`.

`\ll` Производит символ бинарного отношения  $\ll$ , но только в математической моде. Правый вариант ( $\gg$ ) называется, естественно, `\gg`. Прочная.

`\llap` Команда plain TeX'a, записывает свой аргумент на предыдущем тексте. Иными словами, эта команда позволяет вносить запись слева от курсора, не переходя на это новое место. Если набрать что-нибудь вроде `OOO\llap{\$| | | | \$} MMM`, TeX сначала изобразит "ООО MMM", а затем "||||" *поверх* "ООО", а в результате получится "ОФФ MMM". Это используется очень часто. Аналогичная команда, позволяющая вносить запись справа от курсора, не передвигаясь на новое место, называется `\rlap`. Две этих макрокоманды имеют по одной переменной, следовательно, сразу после командного слова должна следовать группа. Для примера попробуем изобразить следующее отношение:

```
\def\toto{\mathrel{\vbox{\hsize=9pt\hrule\kern1pt
\centerline{\$ \circ \$}\kern.6pt\hrule}}}
```

`\$P\toto Q\$` .....  $P \overline{Q}$

Можно получить и противоположное отношение, перечеркнув `\toto` кривой чертой. Для этого нужно собрать вместе `\toto\llap{/\kern2pt}`, превратив их в одно отношение. Некоторые усовершенствования позволят получить окончательный вариант макрокоманды:

```
\def\nototo{\setbox1=\hbox{\$ \toto \$\llap{\raise1pt\hbox{\big/}}
\kern 2pt}} \mathrel{\box1}}
```

`\$P\nototo Q\$` .....  $P \not\overline{Q}$

Для записи на левом поле, как здесь, задайте:

```
\llap{\$ \diamond \$ \quad} Начало строки...
```

- ◇ Начало строки.....
- Команда `\llap` используется и внутри аргумента команды `\displaylines`, чтобы поместить метки в крайне правое положение.

`\lmoustache` Доступна только в математической моде, где производит необычный ограничитель, показанный на рис.5, стр.21 и на рис.6, стр.22. Заметим, что перед этой командой должна стоять команда `\left` или `\right`. Соответствующий правый ограничитель — `\rmoustache`:

`\$ \left \lmoustache формула \right \rmoustache \$` ..... ( формула )

- `\ln` Производит обозначение элементарной функции ( $\ln$ ), но только в математической моде. Если имеются индексы, они никогда не помещаются под и над ним в виде пределов.
- `\lnot` (logical not — логическое “нет”.) Производит ординарный символ  $\neg$ , но только в математической моде. Действует, как команда `\neg`. Прочная.
- `\load{size}{style}` Загружает шрифт, размер которого задается аргументом *size*, а стиль — аргументом *style*; где *size* — это `\tiny`, `\scriptsize`, `\footnotesize`, `\small`, `\normalsize`, `\large`, `\Large`, `\LARGE`, `\huge` или `\Huge`; а *style* — это `\bf`, `\it`, `\sc`, `\sf`, `\sl` или `\tt`.
- `.lof` Расширение файла, который пишется или перезаписывается, только если входной файл содержит команду `\listoffigures` (и не содержит команду `\nofiles`). Такой lof-файл содержит информацию, необходимую для получения в документе списка рисунков. Информация поступает от команд `\caption`, которые встречаются во входном файле внутри командных скобок `figure` и `figure*`. Список рисунков производит команда `\listoffigures`, если во время обработки входного файла существует lof-файл.
- `.log` Расширение имени файла, который записывается при работе ЛАТ<sub>E</sub>X’а и содержит всю ту информацию, которая при обработке входного файла появляется на Вашем терминале, а также некоторую другую информацию.
- `\log` Производит обозначение элементарной функции ( $\log$ ), но только в математической моде. Если имеются индексы, они никогда не появляются под и над нею в виде пределов. Прочная.
- `\longleftarrow` Производит символ бинарного отношения  $\longleftarrow$ , но только в математической моде. Прочная.
- `\Llongleftarrow` Производит символ бинарного отношения  $\Llongleftarrow$ , но только в математической моде. Прочная.
- `\longrightarrow` Производит символ бинарного отношения  $\longrightarrow$ , но только в математической моде. Прочная.
- `\Llongrightarrow` Производит символ бинарного отношения  $\Llongrightarrow$ , но только в математической моде. Прочная.
- `\longmapsto` Производит символ бинарного отношения  $\longmapsto$ , но только в математической моде. Прочная.
- `\longrightrightarrow` Производит символ бинарного отношения  $\longrightarrow$ , но только в математической моде. Прочная.
- `\Llongrightrightarrow` Производит символ бинарного отношения  $\Longrightarrow$ , но только в математической моде. Прочная.

`\lor` (logical or — логическое “или”.) Производит символ бинарной операции  $\vee$ , но только в математической моде. Действует так же, как команда `\vee`.  
Например:

$$\$a\lor(b\land c)=(a\lor b)\land(a\lor c)\$ \quad a \vee (b \wedge c) = (a \vee b) \wedge (a \vee c)$$

Прочная.

`.lot` Расширение файла, который создается или перезаписывается, только если входной файл содержит команду `\listoftables` (и не содержит команду `\nofiles`). Такой `lot`-файл содержит информацию, необходимую для получения в документе списка таблиц. Информация поступает от команд `\caption`, которые встречаются во входном файле внутри командных скобок `table` или `table*`. Список таблиц производит команда `\listoftables`, если во время обработки входного файла существует `lot`-файл.

## m

`\makebox[len][pos]{text}` Вне командных скобок `picture` аналогично `\framebox` (см. описание этой команды), но без “рамки”. Хрупкая.

`\makebox(i,j)[pos]{объект рисунка}` В такой форме команда может встретиться только внутри командных скобок `picture`. Это команда  $\text{\TeX}$ ’а, которая создает бокс, содержащий текст или символы, заключенные в фигурные скобки. Аргумент  $(i, j)$  задает размер получаемого бокса, т.е. получается бокс шириной  $i$  и высотой  $j$  с точкой привязки в его левом нижнем углу. Используется та единица измерения, которая была задана последним присваиванием значения параметру длины `\unitlength`. Необязательный аргумент *pos* задает положение аргумента *объект рисунка* внутри бокса, получаемого командой `\makebox`. Если аргумент *pos* опущен, то *объект рисунка* центрируется внутри получаемого бокса. Это можно изменить, давая аргументу *pos* одно- или двухбуквенное значение. Буквы, которые можно для этого использовать, выбираются из следующих четырех возможностей: `l`, `r`, `t` и `b`. Если в аргументе присутствует только одна из этих букв, то она действует следующим образом:

- `l` *объект рисунка* в получаемом боксе занимает крайне левое положение и находится на полпути между верхом и низом бокса;
- `r` *объект рисунка* в получаемом боксе занимает крайне правое положение и находится на полпути между верхом и низом бокса;
- `t` *объект рисунка* находится вверху получаемого бокса и на полпути между левой и правой границей этого бокса;
- `b` *объект рисунка* находится внизу получаемого бокса и на полпути между левой и правой границей этого бокса.



Для двухбуквенного значения аргумента *pos* есть четыре возможности:

- `tl` объект рисунка находится в левом верхнем углу получаемого бокса;
- `tr` объект рисунка находится в правом верхнем углу получаемого бокса;
- `br` объект рисунка находится в правом нижнем углу получаемого бокса;
- `bl` объект рисунка находится в левом нижнем углу получаемого бокса.

У команды `\makebox` имеются две родственные команды, а именно, команды `\framebox` и `\dashbox`. Команда `\framebox` имеет те же аргументы, что и команда `\makebox`. Единственное отличие между ними состоит в том, что команда `\framebox` производит текст в нарисованной вокруг него прямоугольной рамке. См. рис.12, стр.45.

`\makeglossary` Может находиться только в преамбуле входного файла. Указывает записывать файл с расширением `glo`, который содержит команды `\glossaryentry`, генерируемые из команд `\glossary`, встречающихся во входном файле. Декларация `\nofiles` в преамбуле подавляет запись (или перезапись) файла `glo`.

`\makeindex` Может помещаться только в преамбуле входного файла. Указывает записывать файл с расширением `idx`, который содержит команды `\indexentry`, генерируемые командами `\index` из входного файла. Декларация `\nofiles` в преамбуле подавляет запись (или перезапись) файла `idx`.

`\makelabel{text}` Генерирует метку, которую производит в командных скобках команда `\item`, используя аргумент *text*. Если она не переопределена командой `\@list...`, ее определение по умолчанию размещает метку на одном уровне прямо рядом с правым краем ее бокса. Ее определение можно изменить командой `\renewcommand`.

`\makelabels` Команда не имеет аргумента и может встретиться только в преамбуле при стиле документа `letter`. Указывает, что после того, как будут прогенерированы все письма, на отдельных страницах надо выдать список всех адресов получателей. По желанию эти адреса можно скопировать на наклеивающие этикетки.

`\maketitle` Производит заглавие в стилях документа `article`, `report` и `book`. В последних двух стилях заглавие размещается на отдельной странице, в то время как в стиле документа `article` оно размещается вверху первой страницы результата. На той же самой странице может появиться текст аннотации или его часть. Команда `\maketitle` — если она вообще имеется — может помещаться внутри командных скобок `document`. Если она есть, перед ней должны находиться декларации `\title` и `\author` и может находиться декларация `\date`. См. рис.7 на стр.24.

`\mapsto` Производит символ бинарного отношения  $\mapsto$ , но только в математической моде. Прочная.

- `\marginpar[text1]{text2}` Производит заметки на полях. Так, заметка на полях этой страницы была получена командой `\marginpar{Внимание!}`. Если отсутствует необязательный аргумент *text*<sub>1</sub>, то в качестве заметки на полях появится *text*<sub>2</sub>. Если необязательный аргумент *text*<sub>1</sub> присутствует, то на правом поле заметка будет печататься в виде *text*<sub>2</sub>, а на левом поле — в виде *text*<sub>1</sub>. Хрупкая.
- `\marginparpush` Параметр жесткой длины; действует на внешний вид заметки на полях на странице результата. Равен минимальному расстоянию, разделяющему две последовательные заметки на полях. Прочная команда, поэтому перед ней не надо ставить `\protect`.
- `\marginparsep` Параметр жесткой длины; действует на внешний вид заметки на полях на странице результата. Равен расстоянию между внешней границей тела страницы и внутренней границей заметок на полях. (При односторонней печати по умолчанию внешней границей тела страницы является правая граница, а внутренней границей заметки на полях — левая. При двусторонней печати по умолчанию внешней границей тела страницы для четных страниц является правая граница, а для нечетных страниц — левая граница. Внутренней границей заметок на полях для нечетных страниц будет левая граница, а для четных страниц — правая граница. Это можно изменить декларацией `\reversemarginpar`.) Прочная команда, перед которой никогда не надо ставить `\protect`.
- `\marginparwidth` Параметр жесткой длины, действующий на внешний вид заметок на полях на странице результата. Равен ширине абзацного бокса, содержащего заметку на полях. Прочная команда, перед которой никогда не надо ставить `\protect`.
- `\markboth{text1}{text2}` Используется вместе с опцией `myheadings` декларацией `\pagestyle` или `\thispagestyle`. При двусторонней печати помещает *text*<sub>1</sub> в бегущий заголовок четных страниц, а *text*<sub>2</sub> — в бегущий заголовок нечетных страниц. Хрупкая.
- `\markright{text}` Используется с опцией `myheadings` деклараций `\pagestyle` и `\thispagestyle`. При односторонней печати помещает *text* в бегущий заголовок всех страниц, кроме первой. Хрупкая.
- `math` Командные скобки для набора математических формул, которые печатаются не на отдельных строках, то есть, для набора формул, включенных в текст абзаца. Для этой цели имеются и более короткие формы `\(form\)` и `$form$`. Внутри командных скобок `math` Л<sup>A</sup>T<sub>E</sub>X находится в математической моде и формулы набираются в текстовом стиле.
- `\mathindent` Параметр жесткой длины, равный расстоянию от левой границы тела страницы до левой границы выключенной формулы, если выбрана опция стиля документа `fleqn`. Прочная команда, перед которой никогда не должна стоять `\protect`.

`\mathop` Команда примитивного Т<sub>Е</sub>Xа для создания математического *оператора*: пробелы до и после этого оператора адаптируются к контексту, верхний и нижний индексы печатаются сбоку в `\textstyle` и “над-под” в `\displaystyle`.  
Например:

```
\def\min{\mathop{\rm min}}
```

Для модификации поведения индексов используются команды `\limits` и `\nolimits`. Для создания нового имени функции, пишется, например:

```
\def\airy{\mathop{\rm Airy}\nolimits}
```

См. также `\limits`.

`\mathord` Команда примитивного Т<sub>Е</sub>Xа для создания ординарного символа. Например, `\circ` является отношением. Если написать `$u\circ v$`, это дает  $u \circ v$ . Некоторые считают, что вокруг кружка слишком широкие пробелы. Это можно исправить, превратив отношение в ординарный символ: `$u\mathord\circ v$` дает  $u \circ v$ . Этот же результат можно получить проще: `$u{\circ}v$`.

`\mathstrut` Команда plain Т<sub>Е</sub>X’а, доступная только в математической моде, где производит невидимый вертикальный отрезок, высота которого равна высоте круглой скобки, а ширина нулю. Хотя она и называется `\math...`, ею можно пользоваться и вне математики. Не смешивать `\mathstrut` с другой невидимой чертой, `\strut`, которая больше. Покажем их видимыми (`\mathstrut` окружена круглыми скобками):

```
(\ \hbox{\mathstrut\vrule}\) \hbox{\strut\vrule} (|) |
```

`\matrix` Синтаксис этой команды plain Т<sub>Е</sub>X’а такой:

```
\matrix{form11 & form12 & ... & form1n \cr
 form21 & form22 & ... & form2n \cr
 \vdots \\
 formm1 & formm2 & ... & formmn \cr}
```

Можно использовать только в математической моде, где она производит матрицу без каких-либо ограничивающих скобок. Например:

|                                                                              |                                                                               |
|------------------------------------------------------------------------------|-------------------------------------------------------------------------------|
| $  \begin{matrix}  \alpha & \beta \\  a & b_1 + \dots + b_n  \end{matrix}  $ | $  A = \begin{matrix} a & b_1 + \dots + b_n \\ \alpha & \beta \end{matrix}  $ |
|------------------------------------------------------------------------------|-------------------------------------------------------------------------------|

Матрица в круглых скобках задается командой `\pmatrix` (‘p’ обозначает *parentheses*). Вообще говоря, конструкция `\left ... \right` позволяет окружать матрицу любыми ограничителями. Детерминант матрицы печатается так:

```
$$\left|\matrix{...}\right|$$
```

Внимание, макрокоманда `\det` пишет только “det” романским шрифтом в тексте. Система уравнений кодируется так:

$$\det \left( \begin{matrix} \dots \end{matrix} \right) \Sigma$$

Обратите внимание на *нулевой ограничитель* (`\right.`) и “нумерацию слева” `\legno(\Sigma)`, которая дает  $(\Sigma)$ .

`\max` Производит обозначение максимума (max), но только в математической моде. В выключенной формуле нижний и верхний индексы этого символа располагаются под и над ним в виде пределов.

$$\max_{i \in I} x_i \dots\dots\dots M = \max_{i \in I} x_i$$

$$\displaystyle \max_{i \in I} x_i \dots\dots\dots M = \max_{i \in I} x_i$$

Для модификации поведения индексов см. `\limits`. Прочная.

`\mbox{text}` Можно использовать только в математической моде. Аргумент *text* обрабатывается в LR моде и никогда не будет разбиваться между строками. Любая декларация, которая действует в то время, когда встретилась команда `\mbox`, действует и при обработке текста *text*, если L<sup>A</sup>T<sub>E</sub>X находится в абзацной или LR моде. Если он находится в математической моде, то обработка текста производится под влиянием тех деклараций, которые действовали, когда L<sup>A</sup>T<sub>E</sub>X вошел в эту математическую моду. Например, `\it` когда `$17 > \mbox{возраст}` дает *когда 17 > возраст*. Прочная.

`\medskip` Делает вертикальный пробел, заданный параметром `\medskipamount`. Определена как `\vspace{\medskipamount}`. Заметим, что это отличается от ее определения в plain T<sub>E</sub>X'e. Хрупкая.

`\medskipamount` Параметр эластичной длины, равный 6pt plus 2pt minus 2pt.

`\mho` Производит ординарный символ  $\mathcal{O}$ , но только в математической моде. Заметим, что в plain T<sub>E</sub>X'e такой команды нет. Прочная.

`\mid` Производит символ бинарного отношения  $|$ , но только в математической моде.

$$\{t \mid 0 < t \leq 1\} \dots\dots\dots I = \{t \mid 0 < t \leq 1\}$$

$$\{t \mid 0 < t \leq 1\} \dots\dots\dots I = \{t \mid 0 < t \leq 1\}$$

Прочная.

`\min` Производит обозначение минимума (min), но только в математической моде. В выключенной формуле нижний и верхний индексы этого символа располагаются под и над ним в виде пределов. Прочная.

`minipage` Командные скобки для получения абзацного бокса, то есть, бокса, содержимое которого обрабатывается в абзацной моде. Их синтаксис:

$$\begin{minipage}[pos]{len} text \end{minipage}$$

где *len* — это ширина получаемого бокса, а необязательный параметр *pos* может быть равен *b* или *t*. Опция *b* делает базовую линию нижней строки получаемого бокса частью базовой линии текущей строки текста, а опция *t* делает частью базовой линии текущей строки текста базовую линию верхней строки получаемого бокса. Например, следующие команды

```
\begin{center}
\begin{minipage}{4in}
Хотя путешествие по времени и невозможно, писатели-фантасты,
начиная с Герберта Уэллса, были захвачены этой идеей,
плодотворность которой в который раз была доказана недавним
успехом серии фильмов {\it ‘‘Назад в будущее’’} и
{\it ‘‘Терминатор’’}.\footnote{Дополнительную информацию
о путешествиях по времени можно найти в книгах Лемма
‘‘Рассказ о путешествии по времени’’ и Гарднера ‘‘Путешествие
по времени’’}
Сюжет, который имеет многочисленные воплощения, включает
путешественника по времени, занимающегося историей искусств и
изучающего знаменитого художника (в других версиях
скульптора), который \ldots
\end{minipage}
\end{center}
```

дают такой результат:

Хотя путешествие по времени и невозможно, писатели-фантасты, начиная с Герберта Уэллса, были захвачены этой идеей, плодотворность которой в который раз была доказана недавним успехом серии фильмов *‘‘Назад в будущее’’* и *‘‘Терминатор’’*.<sup>a</sup>

Сюжет, который имеет многочисленные воплощения, включает путешественника по времени, занимающегося историей искусств и изучающего знаменитого художника (в других версиях скульптора), который . . .

---

<sup>a</sup>Дополнительную информацию о путешествиях по времени можно найти в книгах Лемма ‘‘Рассказ о путешествии по времени’’ и Гарднера ‘‘Путешествие по времени’’.

Заметим, что внутри командных скобок `minipage` абзацы набираются без абзацного отступа, но пробел между абзацами делается чуть больше, чем между обычными строками. Сноски внутри `minipage` помечаются не цифрами, а строчными буквами.

`minus len` Ключевое слово Т<sub>E</sub>X'a для задания значения эластичной длины; *len* равна величине, на которую может сжиматься естественная компонента длины.

`\mit` Изменяет стиль печати на математический курсив. Может использоваться только в математической моде.

`\mkern len` Команда примитивного Т<sub>E</sub>X'a, доступная только в математической моде. Генерирует пробел величины *len*, где *len* задана в математических единицах (`\mu`). Вид пробела (горизонтальный или вертикальный) зависит от контекста, в котором встречается команда. `\mkern` выражается в единицах `mu` (*math unit* — математические единицы). Пробел (`\space` или `\` ) равен примерно `6mu`, а самый маленький пробел, определенный в математической моде — это мини-пробел, равный `3mu`. Нам его определение послужит примером синтаксиса:

```
\def\,{\mkern 3mu}
```

`\mkern` делает горизонтальный пробел, вертикального аналога не имеется. Для примера использования см. `\circ`.

`mm` Ключевое слово Т<sub>E</sub>X'a для обозначения *миллиметра*, единицы длины, удовлетворяющей равенству  $10\text{ mm} = 1\text{ cm}$ . (Приблизительно,  $1\text{ mm} = 0.03937\text{ in} = 2.854\text{ pt}$ .)

`\models` Производит символ бинарного отношения  $\models$ , но только в математической моде. Прочная.

`month` (Имя поля ВивТ<sub>E</sub>X'a.) Можно использовать `jan`, `feb`, `mar`, `apr`, `may`, `jun`, `jul`, `aug`, `sep`, `oct`, `nov` или `dec`.

`\mp` Производит символ бинарного оператора  $\mp$ , но только в математической моде. Прочная.

`mpfootnote` Счетчик для нумерации сносок внутри командных скобок `minipage`, который устанавливается в нуль при открытии этих командных скобок. Автоматически увеличивается командами `\footnote` и `\footnotetext` перед их использованием. Хотя его значение должно быть положительным целым числом, в выходном результате он появляется в виде соответствующей строчной буквы.

`mu` Ключевое слово Т<sub>E</sub>X'a для обозначения *математической единицы*, зависящей от шрифта единицы длины, которую можно использовать только в математической моде, причем  $18\text{ mu} = 1\text{ em}$ , где `em` взято из шрифта семейства 2 в текущем стиле. (См. `\mkern`.)

`\mu` Производит ординарный символ  $\mu$ , но только в математической моде. Не путайте с единицей математической длины `mu`, которая рассматривалась выше. Прочная.

`\multicolumn{i}{pre}{text}` Используется, чтобы в рядах командных скобок `tabular`, `tabular*` или `array` расположить материал способом, который отличается от указанного в преамбуле. Аргумент  $i$  — это положительное целое число, которое указывает, на сколько колонок действует эта команда;  $pre$  аналогичен преамбулам командных скобок `tabular` или `array`, за тем исключением, что в нем разрешается меньше выражений, а  $text$  — информация, которая размещается. Команда `\multicolumn` должна стоять либо в начале ряда, либо сразу после амперсанда. Например, следующая таблица

| правила вывода            |                          |
|---------------------------|--------------------------|
| собственные               | несобственные            |
| $\wedge$ -включение       |                          |
| $\wedge$ -исключение      |                          |
| $\vee$ -включение         | $\vee$ -исключение       |
| $\Rightarrow$ -исключение | $\Rightarrow$ -включение |
| $\neg$ -исключение        | $\neg$ -включение        |
| $\neg\neg$ -исключение    |                          |
| $\#\text{-}$ исключение   |                          |
| $\forall$ -исключение     | $\forall$ -включение     |
| $\exists$ -включение      | $\exists$ -исключение    |

была получена такими командами:

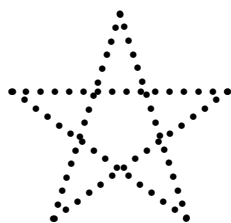
```

\begin{center}
\begin{tabular}{|c|c|} \hline
\multicolumn{2}{|c|}{правила вывода} \\ \hline\hline
собственные & несобственные \\ \hline\hline
 \wedge -включение & \\
 \wedge -исключение & \\
 \vee -включение & \vee -исключение \\
 \Rightarrow -исключение & \Rightarrow -включение \\
 \neg -исключение & \neg -включение \\
 $\neg\neg$ -исключение & \\
 $\#\text{-}$ исключение & \\
 \forall -исключение & \forall -включение \\
 \exists -включение & \exists -исключение \\ \hline
\end{tabular}
\end{center}

```

`\multiply` Умножение на целое число. Прimitивная команда  $\TeX$ а. Примеры:

```
\multiply\count123 by 2, \multiply\dimen7 by 6.
```



```

\unitlength=1mm
\newcommand{\mpt}{\tiny \bullet}
\begin{picture}(35,35)(-17.5,-35)
\multiput(0,0)(0.59,-1.8){16}{\mpt}
\multiput(8.8,-27.1)(-1.53,1.12){16}{\mpt}
\multiput(-14.3,-10.3)(1.9,0){16}{\mpt}
\multiput(14.3,-10.3)(-1.53,-1.12){16}{\mpt}
\multiput(-8.8,-27.1)(0.59,1.8){16}{\mpt}
\end{picture}

```

Рис. 20. Пример использования макрокоманды `\multiput` в командных скобках `picture`

Для сложения (или вычитания) используется `\advance`, а для деления — `\divide`.

`\multiput(i,j)(m,n){t}{объект рисунка}` Может стоять только внутри командных скобок `picture`, где эквивалентна следующим  $t$  командам:

```

\put(i,j){объект рисунка}
\put(i+m,j+n){объект рисунка}
\put(i+2m,j+2n){объект рисунка}
⋮
\put(i+(t-1)m,j+(t-1)n){объект рисунка}

```

Л<sup>A</sup>T<sub>E</sub>X не располагает большим объемом памяти, поэтому разрешено не более 100 таких повторений. Пример использования этой команды можно увидеть на рис.20, стр.96.

`myheadings` Опция стиля страницы в декларациях `\pagestyle` и `\thispagestyle`, которая позволяет самому подготовить информацию для бегущего заголовка страницы. (Основание страницы остается пустым.) Как получить информацию заголовка, подскажут команды `\markboth` и `\markright`. Обратите внимание на оформление текущей страницы. Это происходит потому, что во входном файле этого руководства здесь стоят команды

```

\thispagestyle{myheadings}
\markboth{Заголовок четных страниц}{Заголовок нечетных страниц}

```

## п

`\nabla` В математической моде производит ординарный символ  $\nabla$ . Прочная.

`\natural` В математической моде производит ординарный символ  $\natural$ . Прочная.



- `\ne` Производит символ бинарного отношения  $\neq$ , но только в математической моде. Тот же символ можно получить и командой `\neq`. И обе команды эквивалентны `\not=`. Прочная.
- `\nearrow` (*north east arrow*). Производит символ бинарного отношения  $\nearrow$ , но только в математической моде. Прочная.
- `\neg` Производит ординарный символ  $\neg$ , но только в математической моде. Тот же символ можно получить командой `\lnot` (логическое нет). Прочная.
- `\neq` Производит символ бинарного отношения  $\neq$ , но только в математической моде. Точно такой же символ можно получить и командой `\ne`. Обе эти команды эквивалентны `\not=`. Прочная.
- `\newcommand{cmd}[i]{def}` Неглобальная декларация, определяющая *cmd* — которая к этому времени не должна существовать — как новую команду; *cmd* начинается с бэкслэша, но не с `\end`. Необязательный параметр *i* должен быть положительным целым числом между 1 и 9; он указывает, сколько аргументов имеет определяемая команда. Определение — это *def*. Если в *def* встречается формальный параметр  $\#j$ , то при использовании команды *cmd* он заменяется ее *j*-ым реальным параметром. Пример использования этой команды можно посмотреть на рис.20, стр.96. Хрупкая.
- `\newcounter{ctr1}[ctr2]` Глобальная декларация; делает *ctr1* — который должен состоять целиком из букв и не может являться именем уже существующего счетчика — счетчиком, который инициализируется в нуль. Автоматически определяется команда `\thectr1` как `\arabic{ctr1}`. Нельзя использовать в файле, имя которого встречается в качестве параметра команды `\include`. Если присутствует *ctr2* (который может быть именем уже существующего счетчика), то как только *ctr2* увеличивается командой `\stepcounter` или `\refstepcounter`, счетчик *ctr1* устанавливается в нуль. Пример в описании команды `\fnsymbol`. См. также описание `\thectr`.
- `\newenvironment{env}[i]{def1}{def2}` Определяет новые командные скобки с именем *env*, где *env* должно быть именем новых командных скобок, состоящим целиком из букв. Более того, не должно существовать команды с именем `\env`. Необязательный параметр *i* представляет собой целое положительное число между 1 и 9 и указывает, сколько аргументов будут иметь определяемые командные скобки. Параметр *def1* содержит список команд, которые выполняются, когда ЛАТЭХ наталкивается на команду `\begin{env}`, а *def2* содержит список команд, которые выполняются, когда ЛАТЭХ наталкивается на команду `\end{env}`. Фигурные скобки вокруг *def1* и *def2* являются частью синтаксиса этой декларации и не ограничивают область действия любых деклараций из *def1* и *def2*. Если присутствует параметр *i*, то новые командные скобки открываются следующей командой:

```
\begin{env}{arg1}...{arg2}
```

и это то же самое, что выполнить  $def_1$ , в котором формальные параметры  $\#j$  заменены на аргументы  $arg_i$  (где  $1 \leq j \leq i$ ). Заметим, что  $\#j$  не должно встречаться в  $def_2$ . Хрупкая.

`\newfont{cmd}{font}` Провозглашает новое имя команды  $cmd$  именем прочной декларации — доступной только в абзацной или LR моде — которая делает текущим шрифтом шрифт  $font$ :

`\newfont{\bgbf}{cmbx12 scaled\magstep1}`

`{\bf просто жирный шрифт}` ..... **просто жирный шрифт**

`{\bgbf крупный жирный шрифт}` .....

.....

Хрупкая.

`\newlength{cmd}` Делает имя  $cmd$  — которое должно начинаться с бэкслэша и не должно быть до этого определено — командой длины и задает ей значение нуль. Хрупкая.

`\newline` Обрывает строку в конце текущего абзаца. Хрупкая.

`\newpage` Обрывает текущий абзац и текущую страницу. Если задана опция стиля документа `twocolumn`, то она обрывает текущий абзац и текущую колонку. Прочная.

`\newsavebox{cmd}` Объявляет ячейку памяти с именем  $cmd$ , которое должно быть новым именем, начинающимся с бэкслэша. Используется в соединении с командой `\savebox` или `\sbox`. Хрупкая. Пример использования — в описании команды `\savebox`.

`\newtheorem{env1}[env2]{text}[ctr]` Глобальная декларация; определяет новые командные скобки  $env_1$ . Имя  $env_2$  должно состоять только из букв и не быть именем уже существующих командных скобок или счетчика. Может присутствовать не более одного необязательного аргумента  $env_2$  или  $ctr$ . Аргумент  $text$  содержит слово или предложение, которое появится в выходном результате — по умолчанию жирным шрифтом — когда используются командные скобки  $env_1$ . Если отсутствуют  $env_2$  и  $ctr$ , то будет создан счетчик с именем  $env_1$  и каждое использование командных скобок  $env_1$  даст последовательно (начиная с 1) пронумерованный результат. Если присутствует  $ctr$  (который должен быть именем счетчика), то результат каждого командных скобок  $env_1$  будет иметь номер  $i.j$ , где  $i$  — это значение счетчика  $ctr$ , а  $j$  — номер этих командных скобок. Если присутствует  $env_2$  (который должен быть именем командных скобок, созданных ранее декларацией `\newtheorem`), то результаты как  $env_1$ , так и  $env_2$ , будут последовательно нумероваться в одной числовой последовательности. Хрупкая.

`\ni` Производит символ бинарного отношения  $\ni$ , но только в математической моде. Тот же символ можно получить и командой `\owns`. Используется адептами *машинного языка*, которые могут написать (и прочесть) вот это:

$$\mathbf{R}^n \ni x \xrightarrow{f} x + \phi(x)y \in \mathbf{R}^m$$

Формула получена следующими командами:

```


$$\mathbf{R}^n \ni x \xrightarrow{f} x + \phi(x)y \in \mathbf{R}^m$$


```

Прочная.

`\noalign{text}` Команда примитивного Т<sub>Е</sub>X’а, которая помещает *text* между рядами таблиц, полученных Т<sub>Е</sub>X’ом. Ее нельзя использовать, например, в командных скобках Л<sup>A</sup>Т<sub>Е</sub>X’а `array`. Обычно эта команда увеличивает пробел между рядами результата, полученного, например, командой `\cases`. Она может выглядеть так: `\noalign{\smallskip}`.

`\nocite{key-list}` Используется для включения в библиографию, получаемую ВВТ<sub>Е</sub>X’ом, элемента, ссылка на который не делается. Параметр *key-list* — это список ключей, разделенных запятыми, которые определены в `bib`-файле. Соответствующие элементы помещаются в получаемую библиографию. Чтобы получить список всех элементов, которые содержатся в `bib`-файле, надо внутри командных скобок `document` поставить команду `\nocite{*}`.  
Хрупкая.

`\nofiles` Когда Л<sup>A</sup>Т<sub>Е</sub>X обрабатывает входной файл, он создает (или может создать) несколько дополнительных файлов, а именно, файлы `aux`, `glo`, `idx`, `lof`, `lot` и `toc`. (Например, если Ваш оригинальный файл называется *file.tex*, то вспомогательный файл будет называться *file.aux*.) Если Вы включите декларацию `\nofiles` в преамбулу своего входного файла — а это единственное место, куда ее можно поместить — то ни один из вспомогательных файлов записываться не будет. В частности, если какой-нибудь из них уже существует, он не будет перезаписываться.

`\noindent` Обычно первая строка каждого абзаца печатается, слегка отступив от левой границы тела страницы. (Ширина этого отступа задается параметром длины `\parindent`.) Отступ можно подавить командой `\noindent`, что иногда полезно, например, в начале командных скобок `abstract` и `quotation`.  
Прочная.

`\nolinebreak[i]` Необязательный числовой аргумент *i* может быть равен 0, 1, 2, 3 или 4. Если он отсутствует или равен 4, команда `\nolinebreak` ни при каких обстоятельствах не разрешает делать разрыв строки в том месте, где она встретилась. Если числовой аргумент *i* равен 0, то Т<sub>Е</sub>X может окончить строку в этой точке, но присутствие команды не принуждает и не запрещает это. Если числовое значение *i* равно 1, 2 или 3, то это рекомендует Т<sub>Е</sub>X’у

не делать разрыв строки в этой точке, причем чем выше значение  $i$ , тем настойчивее рекомендация. Хрупкая.

`\nolimits` Конструкция `cmd\nolimits_form_1^form_2` в математической моде помещает  $form_1$  на месте нижнего индекса символа, получаемого командой `cmd`, а  $form_2$  — на месте его верхнего индекса. Сравните, например:

$$\begin{aligned} \text{\$}\sum\nolimits_{i = 1}^{i = n} i^3\text{\$} & \dots\dots\dots \sum_{i=1}^{i=n} i^3 \\ \text{\$}\sum_{i = 1}^{i = n} i^3\text{\$} & \dots\dots\dots \sum_{i=1}^{i=n} i^3 \end{aligned}$$

Работает и в текстовом, и в выключенном стиле. Команда примитивного Т<sub>Е</sub>X’а.

`\nonumber` Может встретиться только внутри командных скобок `eqnarray`. Приводит к тому, что у некоторого ряда номер уравнения не печатается. См. описание `eqnarray`.

`\nonfrenchspacing` По умолчанию Т<sub>Е</sub>X после точки, оканчивающей предложение, помещает увеличенный пробел. Декларация `\nonfrenchspacing` возвращает к этому соглашению, если оно было ранее отменено декларацией `\frenchspacing`. См `\frenchspacing`. Хрупкая.

`\nonstopmode` Команда примитивного Т<sub>Е</sub>X’а обеспечивает безостановочную (в случае встретившейся ошибки) обработку входного файла, хотя при крайней необходимости эту обработку можно остановить по требованию пользователя. Одна из нескольких команд, которые во входном файле могут находиться перед командой `\documentstyle`.

`\nopagebreak[i]` Необязательный числовой аргумент  $i$  может быть равен 0, 1, 2, 3 или 4. Если он отсутствует или равен 4, то место входного файла, где встретилась команда `\nopagebreak`, ни при каких обстоятельствах не будет концом страницы. Если числовой аргумент  $i$  равен 0, то Т<sub>Е</sub>X может оканчивать страницу в этом месте, а может и не оканчивать — команда к этому не принуждает, но и не запрещает. Если числовой аргумент  $i$  равен 1, 2 или 3, то это рекомендует Т<sub>Е</sub>X’у не делать разрыв страницы в этой точке, причем чем выше значение  $i$ , тем настойчивее эта рекомендация. Если задана опция стиля документа `twocolumn`, то команда действует на окончание колонки, а не страницы. Хрупкая.

`\normalmarginpar` Делает так, что заметки на полях помещаются так, как это принято по умолчанию. (См. команду `\marginpar`.)

`\normalsize` Изменяет размер шрифта и в этом размере задает романский шрифт. Обычно выбранный размер больше `\small` и меньше `\large`. См. рис.14 на стр.62. Хрупкая.

`\not` Используется только в математической моде, где производит отрицание следующего за ней символа, помещая на нем слэш.

`$x\not=y$, $U\not\subset V$` .....  $x \neq y, U \not\subset V$

См. также `\llap`, чтобы получать такие отношения “вручную”.

`note` (Имя поля ВивТ<sub>E</sub>X’a.) В поле может появиться любая дополнительная информация, которую Вы хотите поместить в свою библиографию. Например, `note="Edited by John Worrall and Elie Zahar"`. Заметим, что первое слово должно начинаться с прописной буквы.

`\nu` В математической моде производит ординарный символ  $\nu$ . Прочная.

`number` (Имя поля ВивТ<sub>E</sub>X’a.) Номер работы, на которую делается ссылка.

`\numberline{sec-unit}{text}` Может встретиться только как аргумент команды `\addcontentsline` (см. ее описание). Хрупкая.

`\nwarrow` (north west arrow) Производит символ бинарного отношения  $\nwarrow$ , но только в математической моде. Прочная.

## О

`\o` Доступна только в абзацной и LR модах для получения строчной скандинавской буквы  $\o$ . Прочная.

`\O` Доступна только в абзацной и LR модах для получения прописной скандинавской буквы  $\O$ . Не путайте с символом пустого множества `\emptyset` ( $\emptyset$ ), который применяется только в математической моде. Прочная.

`\oddsidemargin` Параметр жесткой длины, действующий на внешний вид страницы результата. На правых страницах при двусторонней печати и на всех страницах при односторонней печати расстояние между левой границей листа бумаги и левой границей тела страницы равно сумме значения `\oddsidemargin` и одного дюйма — см. рис.23 на стр.137. Прочная команда, перед которой никогда не надо ставить `\protect`.

`\odot` Производит символ бинарного оператора  $\odot$ , но только в математической моде. Если Вы хотите сделать его более крупным, пишите `\bigodot`:  $\bigodot$ .

Прочная.

`\oe` Доступна только в абзацной и LR модах для получения строчной французской лигатуры ( $\oe$ ):

`\oe uf, b\oe uf, v\oe ux`..... $\oeuf, boeuf, voeux$ .

Аналогом для пары ‘a-e’ является `\ae`: `n\ae vus` дает  $\naevus$ . Прочная.

`\OE` Доступна только в абзацной и LR модах для получения прописной французской лигатуры (Œ):

`L'\OE IL DE CUIVRE ..... L'ŒIL DE CUIVRE`

Прочная.

`\oint` Производит символ большого оператора  $\oint$  и  $\int$ , но только в математической моде. Когда этот символ встречается в выключенной формуле, он имеет больший размер, чем в формуле внутри текста. Кроме того, в выключенном стиле нижний и верхний индексы располагаются под и над ним в виде пределов. Прочная.

`\omega` Производит ординарный символ  $\omega$ , но только в математической моде. Прочная.

`\Omega` Производит ординарный символ  $\Omega$ , но только в математической моде. Прочная.

`\ominus` Производит символ бинарного оператора  $\ominus$ , но только в математической моде. Прочная.

`\onecolumn` Неглобальная декларация, то есть, подчиняется обычным правилам TeX'a относительно ее области действия. Сначала начинает новую страницу, выполняя команду `\clearpage`, а затем продолжает набирать входные данные в одну колонку.

`openbib` Опция команды `\documentstyle`. Обычно, когда библиография производится командными скобками `thebibliography`, вторая и все последующие строки каждого элемента начинаются точно под началом его первой строки (с меткой “на полях”). Если Вы хотите, чтобы вторая и последующие строки печатались с отступом, то используйте эту опцию. Размером отступа управляет параметр жесткой длины `\bibindent`.

`\opening{text}` Может встретиться только внутри командных скобок `letter`. Используется, чтобы генерировать начало письма. Заметим, что если вы хотите, чтобы слово “Уважаемый” появлялось в получаемом письме, надо включить его в `text`. См. рис.16 на стр.79 и рис.17 на стр.80.

`\oplus` Производит символ бинарного оператора  $\oplus$ , но только в математической моде.

`dim(U\oplus V)=dim U+dim V$ ..... dim(U  $\oplus$  V) = dim U + dim V`

Для большого оператора  $\oplus$  имеется макрокоманда `\bigoplus`.

`displaystyle E=\bigoplus_{i\in I}E_i$ .....  $E = \bigoplus_{i \in I} E_i$`

Прочная.

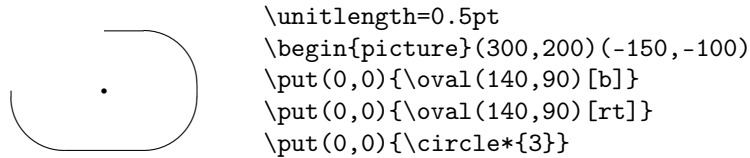


Рис. 21. Использование команды `\oval` для рисования 3/4 рамки с закругленными углами

`organization` (Имя поля ВивТ<sub>E</sub>X'a.) Спонсоры конференции или организация, связанная с этим техническим руководством.

`\oslash` Производит символ бинарного оператора  $\oslash$ , но только в математической моде. Прочная.

`\otimes` Производит символ бинарного оператора  $\otimes$ , но только в математической моде.

`$(E\otimes F)\otimes G=$  
\otimes(F\otimes G)$ .....  $(E \otimes F) \otimes G = E \otimes (F \otimes G)$`

Для большого оператора  $\bigotimes$  используйте макрокоманду `\bigotimes`:

`$(\displaystyle E=\bigotimes_{i \in I} E_i)$` .....  $E = \bigotimes_{i \in I} E_i$

Прочная.

`\oval(p,q)[part]` Может встретиться только в аргументах команд `\put` и `\multiput` внутри командных скобок `picture`. Команда Л<sub>A</sub>T<sub>E</sub>X'a

`\put(i,j){\oval(p,q)[part]}`

рисует удлиненную фигуру с закругленными углами, вписанную в прямоугольник, ширина которого равна  $p$ , а высота —  $q$  единиц, причем центр этой фигуры расположен в точке с координатами  $(i, j)$ . Необязательный аргумент `part` состоит из одно- или двухбуквенного кода, указывающего, какая часть фигуры должна быть нарисована (если этот аргумент отсутствует, то рисуется вся фигура). Однобуквенный код задает половину фигуры. Так:

- l Рисуется левая половина фигуры.
- t Рисуется верхняя половина фигуры.
- r Рисуется правая половина фигуры.
- b Рисуется нижняя половина фигуры.

Двухбуквенный код задает четверть фигуры. Имеют смысл только четыре комбинации:

- `tl` Рисуется левая верхняя четверть фигуры.
- `tr` Рисуется верхняя правая четверть фигуры.
- `br` Рисуется нижняя правая четверть фигуры.
- `bl` Рисуется нижняя левая четверть фигуры.

Пример использования этой команды показан на рис.21, стр.103.

`{form1 \over form2}` Производит дробь с горизонтальной чертой, числителем  $form_1$  и знаменателем  $form_2$ . Команда примитивного Т<sub>Е</sub>X’а. Т<sub>Е</sub>X автоматически центрирует числитель и знаменатель. Не следует использовать дроби с горизонтальной чертой в тексте (в `\textstyle`, а не в `\displaystyle`). Результат (например,  $\frac{1+x^2}{1+a^2+b^2}$ ) является не самым удачным. Не забывайте о читателях с плохим зрением! Лучше используйте косую черту и круглые скобки:  $(1+x^2)/(1+a^2+b^2)$ . Напротив, “маленькие” дроби типа  $\frac{1}{2}x$  и  $\frac{3}{4}y$  в тексте выглядят красиво.

Черта дроби, проведенная командой `\over`, может показаться слишком толстой. Если Вам это не нравится, замените `\over` на вновь определенную команду `\sur: \def\sur{\above.2pt}`.

Пример этой команды можно видеть также на рис.1, стр.10.

`\overbrace{form}` Производит  $\overbrace{form}$ , но только в математической моде. В выключенных формулах верхний индекс помещает метку над фигурной скобкой: `\$e=\bigl[2,\overbrace{1,2,1}^{k=1},\overbrace{1,4,1}^{k=2},\overbrace{1,6,1}^{k=3},\ldots,\overbrace{1,2k,1}^{k=2k},\dots\bigr]`

$$e = [2, \overbrace{1, 2, 1}^{k=1}, \overbrace{1, 4, 1}^{k=2}, \overbrace{1, 6, 1}^{k=3}, \dots, \overbrace{1, 2k, 1}^{k=2k}, \dots]$$

Попробуйте поместить “подпорку” в команду `\overbrace`, написав так: `\overbrace{\strut...}`. Это немного поднимет фигурную скобку и улучшит разборчивость:

$$e = [2, \overbrace{1, 2, 1}^{k=1}, \overbrace{1, 4, 1}^{k=2}, \overbrace{1, 6, 1}^{k=3}, \dots, \overbrace{1, 2k, 1}^{k=2k}, \dots]$$

Команда, помещающая горизонтальную фигурную скобку под своим аргументом, называется `\underbrace`. Она работает в математической моде. Вне математической моды имеется команда `\downbracefill` (которая соответствует `\overbrace`) и команда `\upbracefill` (которая соответствует `\underbrace`) (см. рис.13 на стр.52). Прочная.



`\overleftarrow{form}` Производит  $\overleftarrow{form}$ , но только в математической моде:

`\overleftarrow A=\overleftarrow{f(x,y)}` .....  $\overleftarrow A = \overleftarrow{f(x,y)}$

`\overline{form}` Производит  $\overline{form}$ , но только в математической моде. Для подчеркивания следует использовать команду `\underline`. См. `\bar` для сравнения и некоторых советов. Если Вы используете макрокоманду `\overline` для черты над заглавными буквами, результат будет не совсем удовлетворительный: проведенная черта окажется слегка сдвинутой влево. Чтобы исправить эту ошибку, поместите после `\overline` “отрицательный” пробел `!` и сравните две строки:

$$\overline{A} + \overline{B} + \overline{C} + \overline{M} + \overline{X} + \overline{Y} + \overline{Z}$$

и

$$\overline{A} + \overline{B} + \overline{C} + \overline{M} + \overline{X} + \overline{Y} + \overline{Z} :$$

В первой строке используется команда `\overline{...}`, а во второй — конструкция `\overline{!...}`. Прочная.

`\overrightarrow{form}` Производит  $\overrightarrow{form}$ , но только в математической моде. Идеальная команда для вектора:

`$$\def\vect#1{\overrightarrow{\kern-2pt#1\kern 2pt}}`

`AB^2=AC^2-\,\vect{AB}\cdot\vect{AB}$$`

$$AB^2 = AC^2 - 2\overrightarrow{AB} \cdot \overrightarrow{AB}$$

Коррекция (`\kern-2pt#1\kern 2pt`) дает лучший результат:

`\overrightarrow{AB}` .....  $\overrightarrow{AB}$

`\overrightarrow{\kern-2pt AB\kern 2pt}` .....  $\overrightarrow{AB}$

Не используйте макрокоманду `\vec`: `\vec{AB}` дает  $\vec{AB}$ . `\vec` — это *акцент* для строчных букв:  $(\vec{i}, \vec{j}, \vec{k})$ .

`\owns` Производит символ бинарного отношения  $\owns$ , но только в математической моде. Его можно получить и командой `\ni`. Прочная.

## Р

`p{len}` Может встретиться только в *преамбуле* командных скобок `array` и `tabular`.

Каждый элемент соответствующей ему колонки набирается в абзацном боксе шириной *len*, то есть, как если бы это был аргумент *text* команды `\parbox[t]{len}{text}`. Например, таблица

*утверждение*

*отрицание*

|                 |                                                                                                          |                                                                                                                              |
|-----------------|----------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------|
| <i>утвержд.</i> | Если Джек храпит, то у Макса галлюцинации; но Джек не храпит; <i>поэтому</i> , у Макса нет галлюцинаций. | Это не тот случай когда и Джек храпит, и у Макса галлюцинации; но Джек не храпит; <i>поэтому</i> , у Макса нет галлюцинаций. |
| <i>отриц.</i>   | Либо Джек храпит, либо у Макса галлюцинации; но у Макса нет галлюцинаций; <i>поэтому</i> , Джек храпит.  | Если Джек храпит, то у Макса галлюцинации; но у Макса нет галлюцинаций; <i>поэтому</i> , Джек не храпит.                     |

была получена следующими командами:

```
\begin{center}
\begin{tabular}{cp{2in}p{2in}}
& \multicolumn{1}{c}{\it утверждение}&
\multicolumn{1}{c}{\it отрицание} \\
& & \\
\it утвержд. &
Если Джек храпит, то у Макса галлюцинации; но
Джек не храпит; {\it поэтому}, у Макса нет галлюцинаций. &
Это не тот случай когда и Джек храпит, и у Макса
галлюцинации; но Джек не храпит; {\it поэтому},
у Макса нет галлюцинаций. \\
& & \\
\it отриц. &
Либо Джек храпит, либо у Макса галлюцинации; но у Макса нет
галлюцинаций; {\it поэтому}, Джек храпит. &
Если Джек храпит, то у Макса галлюцинации; но
у Макса нет галлюцинаций; {\it поэтому}, Джек не храпит.
\end{tabular}
\end{center}
\end{table}
```

Так как команда `\\` используется для разделения рядов в командных скобках `array` и `tabular`, внутри *text* она может встретиться только в особых случаях, а именно, внутри таких командных скобок, как `array`, `minipage` или `tabular`, внутри аргумента *text* команды `\parbox` или в области действия — которая должна быть явно обозначена фигурными скобками — декларации `\centering`, `\raggedright` или `\raggedleft`.

`\P` Производит символ “абзац” ¶ во всех модах. Прочная.

`page` Счетчик, который содержит номер *текущей* страницы. В отличие от других счетчиков, увеличивается *после* генерирования номера страницы. Поэтому его начальное значение равно 1, а не нулю.

`\pagebreak[i]` Необязательный числовой аргумент *i* может быть равен 0, 1, 2, 3 или 4. Если он отсутствует или равен 4, команда `\pagebreak` указывает на конец страницы. Выходной результат выравнивается — если другие команды или декларации не отменяют выравнивание — так, что слово, которое стоит перед этой командой, оказывается у правой границы строки, на которой оно расположено. (Это может привести к предупреждающему сообщению о незаполненном горизонтальном боксе.) Если числовой аргумент *i* равен нулю, то  $\TeX$  может окончить страницу в этой точке, но присутствие команды `\pagebreak` не принуждает его к этому, но и не запрещает это. Если числовое значение *i* равно 1, 2 или 3, то команда `\pagebreak` рекомендует  $\TeX$ 'у делать разрыв страницы, причем чем выше значение *i*, тем настойчивее эта рекомендация. При опции стиля документа `twocolumn` эта команда действует на окончание колонки, а не страницы. Хрупкая.

`\pagenumbering{num-style}` Глобальная декларация; указывает, в каком виде номер страницы появится в выходном результате. Параметр *num-style* может быть равен `arabic` (для нумерации арабскими цифрами), `Roman` (для обозначения номеров прописными римскими цифрами), `alph` (для строчных латинских букв) и `Alph` (для прописных латинских букв). По умолчанию значение равно `arabic`. После этой декларации страницы нумеруются, начиная с 1.

Так, например, в больших документах можно фронтальный материал нумеровать римскими цифрами, а основной — арабскими. Это можно получить с помощью команды `\pagenumbering`, что схематически показано ниже:

```
\begin{document}
\pagenumbering{roman} % римская нумерация
% титульный лист
\maketitle % использует \thispagestyle{empty}
% остальной фронтальный материал
. . .
\tableofcontents
% начало основного текста
\cleardoublepage % глава 1 начинается с чет.страницы
\pagenumbering{arabic} % арабская нумерация
\uninclude{chapter1}
\uninclude{chapter2}
. . .
\end{document}
```

Глобальная декларация `\pagenumbering` переопределяет команду `\thepage` в `\num-style{page}`. Хрупкая.

- `\pageref{key}` Используется для организации перекрестных ссылок. Результатом является номер страницы, на которой встретилась соответствующая команда `\label{key}`. Пример см. в описании команды `\label{key}`. Хрупкая.
- `pages` (Имя поля ВивTeX'a.) Диапазон номеров страниц или несколько таких диапазонов или номеров. Например, "679-703", "33-45,60-63" или "3,40-43,70".
- `\pagestyle{page-style}` Декларация, которая подчиняется стандартным правилам TeX'a по поводу областей действия. Определяет внешний вид каждой страницы окончательного выходного документа. Имеется четыре стандартных стиля страницы, а именно, `plain`, `empty`, `headings` и `myheadings`, то есть, `page-style` может быть любым из них. (См. описание этих стилей). Хрупкая.
- `\par` Имеет то же действие, что и пропущенная пустая строка. Прочная.
- `paragraph` Счетчик для управления нумерацией абзацев. Устанавливается в нуль и увеличивается командой `\paragraph` перед генерацией номера. Можно присваивать ему значение командой `\setcounter`. Значение этого счетчика переустанавливается в нуль командами `\chapter`, `\section`, `\subsection` и `\subsubsection`.
- `\paragraph[entry]{heading}` Команда рубрикации, начинающая новый абзац. В стилях документа `article`, `report` и `book` номер уровня абзацев равен 4. По умолчанию в этих стилях документа абзацы автоматически не нумеруются. (Можно изменить, меняя значение счетчика `secnumdepth`.) По умолчанию заголовок `heading` не появляется в оглавлении, если оно производится, но и это можно изменить, меняя значение счетчика `tocdepth`. Если в оглавлении должен появиться заголовок абзаца, то для этого используется `heading`, но если присутствует необязательный аргумент `entry`, в оглавлении появляется он. Если `entry` присутствует, то он является подвижным аргументом, а если отсутствует, то подвижным будет аргумент `heading`. Хрупкая.
- `\paragraph*{heading}` Команда рубрикации, которая открывает новый абзац, но не нумерует его и не заносит в оглавление. Хрупкая.
- `\parallel` Производит символ бинарного отношения `||`, но только в математической моде. Команды `\|` и `\Vert` производят тот же знак, но в качестве ординарного символа, вокруг которого не добавляются специальные пробелы. Прочная.
- `\parbox[pos]{len}{text}` Обрабатывает `text` в абзацной моде и помещает результат в абзацный бокс, ширина которого задается аргументом `len`. Действие команды `\parbox` аналогично действию командных скобок `minipage` и это сходство отражается в их синтаксисе — сравните сами:

```
\begin{minipage}[pos]{len} text \end{minipage}
\parbox[pos]{len}{text}
```

Необязательный аргумент *pos* может быть равен либо **b**, либо **t**. Необязательный аргумент **b** делает базовую линию верхней строки полученного бокса продолжением базовой линии текущей строки текста, а необязательный аргумент **t** делает базовую линию нижней строки полученного бокса продолжением базовой линии текущей строки текста. Например:

|                                                                                                    |              |                                                                                                                                            |
|----------------------------------------------------------------------------------------------------|--------------|--------------------------------------------------------------------------------------------------------------------------------------------|
| <p>Это абзацный<br/>бокс, выровненный по нижней строке.<br/>он выровнен по его верхней строке.</p> | <p>А ЭТО</p> | <pre>\parbox[b]{1in}{Это абзацный бокс, выровненный по нижней строке.} \ А ЭТО \ \parbox[t]{1in}{он выровнен по его верхней строке.}</pre> |
|----------------------------------------------------------------------------------------------------|--------------|--------------------------------------------------------------------------------------------------------------------------------------------|

В аргументе *text* не могут встречаться такие командные скобки, как `center`, `description`, `enumerate`, `flushleft`, `flushright`, `itemize`, `quote`, `quotation`, `thebibliography` и `verse`, любые командные скобки, заданные декларацией `\newtheorem`, командные скобки `tabular`, а также команды, производящие сноски. Хрупкая.

`\parfillskip` Параметр жесткой длины, равный величине горизонтального пробела в последней строке абзацев. Для этой команды есть забавное применение: если Ваш абзац достаточно длинный, попробуйте задать `\parfillskip=0mm`. Так как Т<sub>Е</sub>X больше не имеет права добавлять пробел в конце абзаца, последняя строка, если она достаточно длинная, *должна* будет окончиться на правом поле:

*На краю мертвого марсианского моря раскинулся безмолвный городок. Он был пуст. Ни малейшего движения на улицах. Днем и ночью в универмагах одиноко горели огни. Двери лавок открыты настежь, словно люди обратились в бегство, забыв о ключах. На проволочных рейках у входа в немые закусочные нечитанные, порыжевшие от солнца, шелестели журналы, доставленные месяц назад ракетой с Земли.<sup>8</sup>*

```
{\parfillskip=0mm\sl...\par}
```

Но это работает не во всех случаях! Обратите внимание, что Т<sub>Е</sub>X здесь раздвигает пробелы между словами немного больше, чем обычно. Команда `\par`, которая расположена перед закрывающей фигурной скобкой группы и делает локальной команду `\parfillskip=0mm`, является обязательной. Команда примитивного Т<sub>Е</sub>X'а.

`\parindent` Параметр жесткой длины, равный ширине отступа в начале обычного абзаца. В боксе, полученном командой `\parbox`, равен 0 дюймов. В любом месте значение можно изменить, например, командой `\setlength`. Прочная команда, перед которой никогда не надо ставить `\protect`.

---

<sup>8</sup>Р.Бредбери, *Марсианские хроники*.

`\parsep` Параметр эластичной длины, определяющий вертикальный пробел, который помещается между абзацами одного пункта внутри командных скобок `list`; графически его действие показано на рис.18, стр.81. Прочная команда, перед которой не надо ставить `\protect`.

`\parskip` Параметр эластичной длины, равный дополнительному вертикальному пробелу, вставляемому между последовательными абзацами (кроме обычного пробела `\baselineskip` между двумя последовательными строками внутри абзаца). Заметим, что `\parskip` — это эластичная длина, естественное значение которой равно нулю дюймов. Значение можно изменить — например, командой `\setlength` — но оно всегда должно оставаться эластичным или растяжимым. Прочная команда, перед которой никогда не надо ставить `\protect`.

`part` Счетчик для управления нумерацией частей. Устанавливается в нуль и увеличивается командой `\part` перед тем, как генерируется номер. Значение счетчику можно присвоить командой `\setcounter`.

`\part[entry]{heading}` Команда рубрикации, которая открывает новую часть документа. Можно использовать в стилях документа `article`, `report` и `book`. В стиле документа `article` части имеют номер уровня 0, а в стилях документа `report` и `book` их номер уровня равен -1. Во всех стилях документа по умолчанию части нумеруются автоматически. (Можно изменить, поменяв значение счетчика `secnumdepth`.) Если отсутствует необязательный аргумент `entry`, то по умолчанию в оглавлении, если оно производится, появится `heading`. (Можно изменить, поменяв значение счетчика `tocdepth`.) Если же `entry` присутствует, то он появится в оглавлении, а `heading` появится в теле получаемого документа. Если `entry` присутствует, то он является подвижным аргументом, а если отсутствует, подвижным будет аргумент `heading`. Хрупкая.

`\part*{heading}` Команда рубрикации, которая открывает новую часть, которая не нумеруется и не появляется в оглавлении.

`\partial` Запись частных производных  $\partial$ . Только в математической моде:

$$\begin{aligned} & \mathcal{L}\{\partial^2 f \over \partial x \partial y\} = \\ & \mathcal{L}\{\partial^2 f \over \partial x \partial y\} \dots\dots\dots \frac{\partial^2 f}{\partial x \partial y} = \frac{\partial^2 f}{\partial x \partial y} \end{aligned}$$

Ординарный символ  $\delta$  получается командой `\delta`. Прочная.

`\partopsep` Параметр эластичной длины, равный дополнительному вертикальному пробелу, добавляемому перед и после командных скобок `list`, если перед открывающей командной скобкой есть пустая строка. См. рис.18 на стр.81. Прочная команда, перед которой не надо ставить `\protect`.

`pc` Ключевое слово TeX'a для *пайки* (*pica*), единицы длины, которая удовлетворяет равенству: 1 pc=12 pt. (Приблизительно, 1 pc = 0.166 in = 4.22 mm.)

`\perp` Производит символ бинарного отношения  $\perp$ , но только в математической моде; команда `\bot` производит тот же знак, но в качестве ординарного символа. Прочная.

`\phi` Производит ординарный символ  $\phi$ , но только в математической моде. Ординарный символ  $\varphi$  получается командой `\varphi`. Прочная.

`\Phi` Производит ординарный символ  $\Phi$ , но только в математической моде. Прочная.

`\pi` Производит ординарный символ  $\pi$ , но только в математической моде. Ординарный символ  $\varpi$  получается командой `\varpi`. Прочная.

`\Pi` Производит ординарный символ  $\Pi$ , но только в математической моде. Не путайте с оператором `\prod`:

`$$\Pi, \prod, \displaystyle\prod$` .....  $\Pi, \prod, \prod$

Прочная.

`picture` Командные скобки для получения простых линейных рисунков. Примеры показаны на рис.11 стр.39, рис.19, стр.84 и др.

`plain` Опция стиля страницы для деклараций `\pagestyle` и `\thispagestyle`. Номер страницы помещается в ее основании, а заголовок страницы остается пустым — как на этой странице. Принят по умолчанию в стилях документа `article` и `report`.

`plain` Необязательный аргумент команды `\bibliographystyle`. Полученная библиография сортируется по алфавиту и помечается числовыми метками, например, так: [17].

`plus` *len* Ключевое слово TeX'a, которое вместе с ключевым словом `minus` используется для присваивания значений параметрам эластичной длины; *len* представляет собой величину, на которую может увеличиться естественная длина.

`\pm` Производит символ бинарного оператора  $\pm$ , но только в математической моде. Прочная.

`\pmatrix` Команда plain TeX'a. Синтаксис:

```
\pmatrix{form11 & form12 & ... & form1n \cr
 form21 & form22 & ... & form2n \cr
 \vdots
 formm1 & formm2 & ... & formmn \cr}
```

Похожа на команду `\matrix`, но полученная ею матрица заключается в большие круглые скобки. Матрица может состоять из любых элементов и даже из других матриц;

$$J = \begin{pmatrix} \begin{pmatrix} \lambda & 1 \\ 0 & \lambda \end{pmatrix} & 0 \\ 0 & \begin{pmatrix} \mu & 0 \\ 0 & \mu \end{pmatrix} \end{pmatrix}$$

Эта эффективная матрица кодируется очень просто:

```

 $\mathbb{J} = \begin{pmatrix} \begin{pmatrix} \lambda & 1 \\ 0 & \lambda \end{pmatrix} & 0 \\ 0 & \begin{pmatrix} \mu & 0 \\ 0 & \mu \end{pmatrix} \end{pmatrix}$

```

См. также команду `\matrix`.

`\pmod` Можно использовать только в математической моде, где получается “модульное” выражение в скобках. Например, `\mathbb{m}_1 \equiv \mathbb{m}_2 \pmod{n}` дает  $m_1 \equiv m_2 \pmod{n}$ . Прочная.

`\roptabs` Доступна только внутри командных скобок `tabbing`, где восстанавливает расположение позиций табуляции, сохраненное предыдущей командой `\pushtabs`.

`\pounds` Во всех модах производит знак фунта стерлингов  $\mathcal{L}$ . Прочная.

`\Pr` Производит математическое обозначение (Pr), но только в математической моде. В выключенных формулах нижний и верхний индексы располагаются под и над ним в виде пределов. Прочная.

`\prec` Производит символ бинарного отношения  $\prec$ , но только в математической моде. Прочная.

`\preceq` Производит символ бинарного отношения  $\preceq$ , но только в математической моде. Противоположное отношение ( $\succeq$ ) кодируется `\succeq`. Прочная.

`\prime` Производит ординарный символ  $\prime$ , но только в математической моде. Предназначен для употребления в индексах, поэтому обычно мы встречаем его в уменьшенном размере: `\mathbb{y}_1^\prime` дает  $y'_1$ . Заметим, что `\mathbb{S}'` реализовано как `\mathbb{S}^\prime`. Прочная.

`\prod` Большой оператор  $\prod$  и  $\prod$  (его размер зависит от стиля). Только в математической моде. В выключенных формулах размер символа несколько больше, чем в формулах, включенных в текст. К тому же в выключенных формулах нижний и верхний индексы располагаются под и над этим символом в виде пределов.

`\cal C(n;a) = e^{-a} \prod_{i=1}^n (-a)^{-n_i} \sum_{k \geq 0} \dots`



`\prod_{i=1}^m C_{n_i}^{(a)}(k) \frac{a^k}{k!}`

$$C(n; a) = e^{-a} \prod_{i=1}^m (-a)^{n_i} \sum_{k \geq 0} \prod_{i=1}^m C_{n_i}^{(a)}(k) \frac{a^k}{k!}$$

`\propto` Производит символ бинарного отношения  $\propto$ , но только в математической моде. Не путайте с греческой буквой `\alpha`, которую оно напоминает:

`\$ \propto, \alpha \$` .....  $\propto, \alpha$

Прочная.

`\protect` Должна стоять перед каждой хрупкой командой, которая находится в подвижном аргументе. (Конечно же, прочная.)

`\ps{text}` Может встретиться только в командных скобках `letter`. Используется для получения в письмах постскриптума — см. рис.16 на стр.79 и рис.17 на стр.80. Заметим, что  $\text{\LaTeX}$  не генерирует буквы ‘PS’, поэтому команда `\pps` не нужна.

`\psi` Производит ординарный символ  $\psi$ , но только в математической моде. Прочная.

`\Psi` Производит ординарный символ  $\Psi$ , но только в математической моде. Прочная.

`pt` Ключевое слово  $\text{\TeX}$ ’а для *point* (пункта), единицы длины, которая удовлетворяет равенству 1 in = 72.27 pt. (Приблизительно, 1 pt = 0.0138 in = 0.35 mm.)

`publisher` (Имя поля  $\text{\BibTeX}$ ’а.) Название издательства. Например, "Oxford University Press" или "Springer".

`\pushtabs` Команда, доступная только внутри командных скобок `tabbing`, где запоминает расположение позиций табуляции (которое затем можно восстановить командой `\poptabs`).

`\put(i,j){объект рисунка}` Может встретиться только внутри командных скобок `picture`, где помещает на картинке *объект рисунка* в положение с координатами  $(i,j)$ . Каждый объект рисунка имеет связанную с ним точку привязки и координаты  $(i,j)$  указывают ее точное положение. (Объектом рисунка может быть почти все, что угодно, но некоторые объекты используются чаще других.) Примеры показаны на рис.11 стр.39, рис.19, стр.84 и др.

## q

`\qqquad` Команда plain T<sub>E</sub>X'a; во всех модах производит горизонтальный пробел шириной в два em.

`\quad` Команда plain T<sub>E</sub>X'a; во всех модах производит горизонтальный пробел шириной в один em.

`quotation` Командные скобки для получения цитат; могут встретиться только в абзацной моде. Внутри них L<sup>A</sup>T<sub>E</sub>X находится в абзацной моде. Как и в первых строках абзацев, отступ делается и в первой строке этих командных скобок, что можно подавить командой `\noindent`. Вертикальные пробелы между абзацами имеют обычный размер. Левое и правое поля результирующего бокса расположены на одинаковом расстоянии от левого и правого полей страницы. Например, команды

```
\begin{quotation}
\noindent
Схема типичного художественного фильма: набор трансформаций,
действующих на ряде неустойчивостей с таким результатом, что
достигается некое художественное равновесие. Хит (Heath) пишет:
%
\begin{quotation}
\noindent
Повествовательное действие --- это такой ряд элементов, имеющих
отношение к трансформации, что их последовательность ---
переход от одной трансформации к другой --- определяет
состояние S' , отличное от состояния S \ldots
\end{quotation}

Одна из руководствующих метафор Хита в описании действия
художественного фильма --- это ‘‘верни вещи на место.’’
\end{quotation}
```

производят следующий фрагмент текста, напечатанный с отступами:

Схема типичного художественного фильма такова: набор трансформаций, действующих на ряде неустойчивостей с таким результатом, что достигается некоего вида художественное равновесие. Хит (Heath) пишет:

Повествовательное действие — это такой ряд элементов, имеющих отношение к трансформации, что их последовательность — переход от одной трансформации к другой — определяет состояние  $S'$ , отличное от состояния  $S$  . . .

Одна из руководствующих метафор Хита в описании действия художественного фильма — это “верни вещи на место.”

`quote` Командные скобки, которые используются для коротких цитат и только в абзацной моде. Внутри них L<sup>A</sup>T<sub>E</sub>X находится в абзацной моде. В первой строке нового абзаца абзацный отступ не делается, а между абзацами вставляется дополнительный вертикальный пробел. Левое и правое поле результирующего абзацного бокса находятся на одинаковом расстоянии от левого и правого полей страницы. Например, команды:

```
\begin{quote}
О стиле Жордана (Jordan) говорят, что если у него есть
4 однородные переменные, он их обозначит как a ,
 M_3^{\prime} , ϵ_2 и $\Pi_{1,2}^{\prime}$.
\end{quote}
```

дают следующий результат<sup>9</sup>:

О стиле Жордана (Jordan) говорят, что если у него есть 4 однородные переменные, он их обозначит как  $a$ ,  $M_3^{\prime}$ ,  $\epsilon_2$  и  $\Pi_{1,2}^{\prime}$ .

## Г

`\raggedbottom` Позволяет несколько различаться высоте тела различных страниц, при этом вертикальные пробелы между абзацами остаются постоянными.

`\raggedleft` Используется для получения абзацев, которые слева не выровнены, а имеют неровную левую границу.

`\raggedright` Используется для получения абзацев, которые имеют неровную правую границу.

`\raisebox{ $len_1$ }[ $len_2$ ][ $len_3$ ]{text}` Аргумент *text* обрабатывается в LR моде и результирующий бокс поднимается над текущей базовой линией на расстояние  $len_1$ . Если присутствует  $len_2$ , то полученный бокс продлевается на расстояние  $len_2$  над базовой линией, а если присутствует  $len_3$ , то полученный бокс продлевается на расстояние  $len_3$  под базовой линией. Можно *поднять* или *опустить* текст. Можно `\raise` или `\raisetext`.

Хрупкая.

Для более опытных пользователей: имеется также команда примитивного T<sub>E</sub>Xа `\raise` — бокс можно приподнять командой `\raise 2mm\vbox{...}`

<sup>9</sup>Littlewood, J. E. *Littlewood's Miscellany*

и опустить командой `\raise -3mm\hbox{...}`. Так перемещать можно любой тип боксов. Внимание: `\raise` функционирует только в *горизонтальной* моде, т.е., на строке, в `\hbox` или в таблице.

За командой `\raise` должен обязательно следовать бокс: `\raise 3pt{...}` вызывает сообщение об ошибке.

Посмотрите на интересную формулу: она расположена в квадратных скобках, которые спускаются очень низко. Если по наивности написать `\left[дробь\right]`, получатся квадратные скобки, которые спущены, как надо, но к тому же еще и слишком высоко подняты (что нас не устраивает):

$$\sin z = z \lim_{p \rightarrow \infty} \prod_{k=1}^{p-1} \left[ 1 - \frac{z^2}{4p^2 \tan^2 \frac{k\pi}{2p}} \right]$$

Нам же надо создать квадратные скобки нужной высоты, а потом их опустить:

```
\def\sur{\above .2pt}
\def\crochet#1{\raise -5pt
\hbox{\$ \left#1 \vbox to 22pt{\right.$}}
$$\sin z=z\lim_{p\rightarrow\infty}\prod_{k=1}^{p-1}
\crochet{[1-\frac{z^2}{\displaystyle 4p^2\tan^2\frac{k\pi}{2p}}]}
\crochet{]}$$
```

Поиск хороших параметров потребовал нескольких попыток, но результат получился прекрасный:

$$\sin z = z \lim_{p \rightarrow \infty} \prod_{k=1}^{p-1} \left[ 1 - \frac{z^2}{4p^2 \tan^2 \frac{k\pi}{2p}} \right]$$

В примитивном Т<sub>E</sub>Xе есть также команда `\lower`, опускающая боксы.

`\rangle` Производит закрывающий символ `\rangle`, но только в математической моде.

Не смешивать со знаком неравенства `>`, который более острый. Открывающая угловая скобка `\langle`, конечно же, называется `\langle`. Прочная.

`\rbrace` Производит закрывающий символ `\rbrace`, но только в математической моде.

Так же действует и команда `\}`. См. `\lbrace`. Прочная.

`\rbrack` Закрывающая квадратная скобка `\rbrack`. См. `\lbrack`. Прочная.

`\rceil` Производит закрывающий символ `\rceil`, но только в математической моде.

См. `\lceil`. Прочная.

`\Re` Производит ординарный символ “действительная часть”:  $\Re$ . См. макрокоманду `\Im`. Только в математической моде. Прочная.

`\ref{key}` Используется для получения перекрестных ссылок. Производит в выходном результате значение `\ref`, которое связано с ключом *key* командой `\label{key}`. Примеры в описании команды `\label`. Хрупкая.

`\refstepcounter{ctr}` Значение счетчика *ctr* увеличивается на 1, а значение всех вложенных счетчиков устанавливается в нуль. Более того, текущим значением `\ref` становится текст, генерируемый командой `\thectr`.

`\renewcommand{cmd}[i]{def}` Аналогична `\newcommand` (см. ее описание), за тем исключением, что *cmd* может быть именем уже определенной команды. Хрупкая.

`\renewenvironment{env}[i]{def1}{def2}` Аналогична `\newenvironment` (см. ее описание), за тем исключением, что *env* может быть именем уже существующих командных скобок. Хрупкая.

`report` Может быть аргументом *doc-style* команды `\documentstyle`. Стиль документа `report` (отчет) похож на `article` (см. его описание) и по умолчанию в нем, в основном, приняты такие же соглашения. Одно из основных отличий — в стиле `report` есть команда рубрикации `\chapter`. По умолчанию выходной результат набирается в односторонней манере, но это можно изменить опцией `twoside` команды `\documentstyle`. По умолчанию принят стиль страниц `empty`, то есть, заголовок страницы пустой, а ее номер находится в центре основания. Если используется стиль страниц `headings`, то заголовок страницы состоит из номера главы и номера страницы — если только не выбрана опция `twoside`: в этом случае заголовки четных страниц состоят из номера главы и номера страницы, а нечетных страниц — из номера секции и номера страницы. Это можно изменить с помощью опции `muheadings` и связанных с ней команд.

Если включена декларация `\flushbottom`, то высота текста на всех страницах выходного результата одинакова — при необходимости добавляются вертикальные пробелы — но по умолчанию действует декларация `\raggedbottom`. Это позволяет высоте текста на страницах выходного результата несколько изменяться. (Если включена опция `twoside`, то по умолчанию действует декларация `\flushbottom`). Если присутствует команда `\maketitle` — обычно сразу после начала командных скобок `document` — то создается отдельный титульный лист. Так же, как и в статьях, перед командой `\maketitle` должны стоять декларации `\title` и `\author` и может появиться декларация `\date`. Как уже объяснялось в разделе 6.1.3, в аргументах этих команд могут встретиться команды `\thanks`. Если нужна аннотация, то вскоре после команды `\maketitle` должны располагаться командные скобки `abstract`. Аннотация набирается на отдельной странице.

Если для получения номеров уравнений используются командные скобки `equation` и `eqnarray`, то уравнения нумеруются внутри каждой главы последовательно, начиная с X.1, где X — номер главы, далее X.2, и т.д. Если в документе встречаются рисунки, они также нумеруются в виде X.1, X.2, ..., где X — номер главы. То же можно сказать и про нумерацию таблиц (X.1, X.2, ..., где X — номер главы). Заметим, что и уравнение, и рисунок, и таблица могут иметь один и тот же номер, скажем, X.1. Заметим также, что если командные скобки `equation` или `eqnarray`, рисунок или таблица встретятся перед командой `\chapter`, то X будет равен 0.

См. также описание стиля `article` и рис.5 на стр.33.

`\reversemarginpar` Приводит к тому, что заметки на полях появляются на поле, противоположном принятому по умолчанию. См. `\marginpar`.

`\rfloor` Производит закрывающий символ  $\rfloor$ , но только в математической моде. См. `\lfloor`. Прочная.

`\rgroup` Доступна только в математической моде, где производит ограничитель, который выглядит как большая правая круглая скобка; однако перед этой командой должна стоять либо `\left`, либо `\right`. См. `\lgroup`. Прочная.

`\rhd` Производит символ бинарного оператора  $\triangleright$ , но только в математической моде. Заметим, что в plain TeX'e такой команды нет. Прочная.

`\rho` Производит ординарный символ  $\rho$ , но только в математической моде. Ординарный символ  $\varrho$  получается командой `\varrho`. Прочная.

`\right` Неотделима от своей пары `\left`. См. `\left`.

`\rightarrow` Производит символ бинарного отношения  $\rightarrow$ , но только в математической моде. Тот же символ производит и команда `\to`. Прочная.

`\Rrightarrow` Производит символ бинарного отношения  $\Rightarrow$ , но только в математической моде. Прочная.

`\rightarrowfill` Производит указывающую вправо стрелку, которая заполняет пробел, созданный другими командами. Следовательно, может работать только внутри `\hbox`, таблицы и т.д. Примеры:

`\hbox to 5cm{длинная \rightarrowfill\ стрелка}` .....  
 .....длинная  $\longrightarrow$  стрелка

Графически это показано на рис.13, стр.52.

`\rightharpoonowdown` Производит символ бинарного отношения  $\rightharpoonowdown$ , но только в математической моде. Прочная.

`\rightharpoonoup` Производит символ бинарного отношения  $\rightharpoonoup$ , но только в математической моде. Прочная.

`\rightleftharpoons` Производит символ бинарного отношения  $\rightleftharpoons$ , но только в математической моде. Прочная.

`\rightmargin` Параметр жесткой длины, действующий на внешний вид командных скобок `list`, как это показано на рис.18, стр.81 Прочная команда, перед которой не надо ставить `\protect`.

`\rlap` Команда plain Т<sub>E</sub>X'a; печатает текст, который наползает вправо. Используется внутри аргумента команды `\displaylines`, чтобы поместить метки в крайне левом положении области тела страницы. Синтаксис — `\rlap{...}`. Внимание: в `\rlap` Вы находитесь в текстовой моде, поэтому не забывайте \$, когда это необходимо: `\rlap{\$(\Sigma)$}`. Для примеров см. `\llap`.

`\rm` Изменяет шрифт на романский — см. рис.9 на стр.28. Команду можно использовать глобально `\rm...` или локально `{\rm...}`. Второй вариант часто используется в математической моде. Прочная.

`\rmoustache` Доступна только в математической моде, где производит необычный ограничитель, показанный на рис.5, стр.21 и на рис.6, стр.22. Заметим, что перед этой командой должна стоять команда `\left` или `\right`.

`roman` Один из возможных параметров глобальной декларации `\pagenumbering`. Номера страниц выводятся строчными римскими цифрами.

`\roman{ctr}` Значение счетчика `ctr` выводится строчными римскими цифрами. Прочная.

`Roman` Один из возможных параметров глобальной декларации `\pagenumbering`. Номера страниц печатаются прописными римскими цифрами.

`\Roman{ctr}` Значение счетчика `ctr` выводится прописными римскими цифрами. Прочная.

`\root` Команда plain Т<sub>E</sub>X'a для радикалов: `\root 5\of{1+x^2}` дает  $\sqrt[5]{1+x^2}$  в стиле `\textstyle` и  $\sqrt[5]{1+x^2}$  в `\displaystyle` (посвободнее). Л<sup>A</sup>T<sub>E</sub>X для этой цели имеет команду `\sqrt` — см. ее описание.

```
$$\root\scriptstyle 3\of{-\{q\over 2\}+\sqrt{\{q^2\over 4\}
+\{p^3\over 27\}}+\root\scriptstyle 3\of{-\{q\over 2\}
-\sqrt{\{q^2\over 4\}+\{p^3\over 27\}}}}$$
```

$$\sqrt[3]{-\frac{q}{2} + \sqrt{\frac{q^2}{4} + \frac{p^3}{27}}} + \sqrt[3]{-\frac{q}{2} - \sqrt{\frac{q^2}{4} + \frac{p^3}{27}}}$$

Заметьте использование `\scriptstyle` для некоторого увеличения показателя корня '3' в формуле Кардана. Без этой предосторожности 3 окажется в `\scriptscriptstyle` (и будет слишком маленькой).

`\rule[ $len_1$ ]{ $len_2$ }{ $len_3$ }` Делает закрашенный прямоугольник шириной  $len_2$  и высотой  $len_3$ , который помещается на расстоянии  $len_1$  над или под текущей базовой линией (в зависимости от того, положительное или отрицательное  $len_1$ ). По умолчанию  $len_1$  равно 0 миллиметров.

`\rule[5pt]{10pt}{20pt}` .....



Часто используется для получения невидимой “подпорки” — прямоугольника заданной высоты и нулевой ширины. Хрупкая.

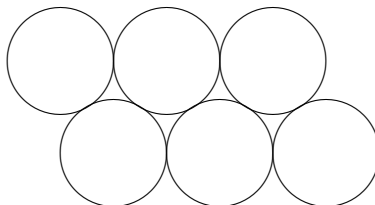
## S

`\samepage` Используется, чтобы расположить материал из ее области действия на одной странице. Разбиение страниц внутри области действия делается командой `\pagebreak`. Хрупкая.

`\S` Во всех модах производит символ параграфа §. Прочная.

`\savebox{cmd}[len][pos]{text}` Вне командных скобок `picture` обрабатывает `text` в LR моде, а результирующий бокс помещает в ячейку памяти, связанную с `cmd`, которая должна начинаться с бэкслэша и быть предварительно определена командой `\newsavebox`. Ширину получаемого бокса можно задать необязательным параметром `len`, который должен быть длиной (например, `2.3in`). Если задана ширина, `text` в полученном боксе центрируется, если не задан `pos`. Если `pos` равен 1, то `text` прижимается к левой границе получаемого бокса, а если `pos` равен `r`, то к его правой границе. Хрупкая.

`\savebox{cmd}(i,j)[pos]{объект рисунка}` В такой форме команда может встретиться только внутри командных скобок `picture`. Параметры  $(i,j)$ , `pos` и `объект рисунка` имеют тот же смысл, что и в случае команды `\makebox`. Параметр `cmd` является именем ячейки памяти, должен начинаться с бэкслэша и быть предварительно введен командой `\newsavebox`. Обратите внимание на интересный пример использования этой команды. Следующее расположение окружностей:





можно получить следующими командами:

```
\newsavebox{\balls}
\savebox{\balls}(120,69.28){%
\begin{picture}(120,69.28)
\multiput(20,34.64)(40,0){3}{\circle{40}}
\multiput(0,69.28)(40,0){3}{\circle{40}}
\end{picture}}
\begin{picture}(340,173.2)
\multiput(0,0)(0,69.28){2}
{\multiput(0,0)(120,0){3}{\usebox{\balls}}}
\end{picture}
```

Здесь команда `\newsavebox` создает *ячейку памяти* с именем `\balls`. Ячейка памяти — это область памяти, в которой хранится бокс. Рекомендуется хранить те боксы, которые требуют большой обработки и используются в документе несколько раз. Если вместо хранимого бокса использовать макроманду,  $\TeX$  будет обрабатывать этот бокс каждый раз заново.

Ту же картинку можно получить и другим способом:

```
\begin{center}
\begin{picture}(100,103.92)
\multiput(20,34.64)(40,0){3}{\circle{40}}
\multiput(0,69.28)(40,0){3}{\circle{40}}
\end{picture}
\end{center}
```

Хрупкая.

`\sb` Альтернативный способ задания нижнего индекса. Как  $x_{83}$ , так и  $x\sb{83}$  дают  $x_{83}$ .

`\sbox{cmd}{text}` Сокращенная форма команды `\savebox`. Вне командных скобок `picture` декларация обрабатывает `text` в LR моде, а результирующий бокс помещается в ячейку памяти, связанную с `cmd`, имя которой должно начинаться с бэкслэша и предварительно объявляться командой `\newsavebox`. Прочная.

`\sc` Декларация, которая меняет шрифт печати на капитель (SMALL CAPITALS) — см. рис.9 на стр.28. Прочная.

`school` (Имя поля ВивТ $\TeX$ 'а.) Название учебного заведения, на рассмотрение которого были представлены тезисы.

`\scriptscriptstyle` В математической моде Т<sub>Е</sub>X работает в четырех *стилях*, `\displaystyle`, `\textstyle`, `\scriptstyle` и `\scriptscriptstyle`. От стиля зависит используемый шрифт и распределение пробелов (горизонтальных и вертикальных). Если в используется шрифт в 10 пунктов, стили `\displaystyle` и `\textstyle` также используют шрифт в 10 пунктов, стиль `\scriptstyle` — шрифт в 7 пунктов, а стиль `\scriptscriptstyle` — шрифт в 5 пунктов. Обычно стиль выбирается автоматически. Но его можно и явно задавать по имени:

$$\text{\displaystyle } N=2^{\text{\scriptstyle } 2^{\text{\scriptscriptstyle } 2^2}} \dots\dots\dots N = 2^{2^{2^2}}$$

Первый показатель степени печатается в `\scriptstyle`, а все последующие — в `\scriptscriptstyle`. Если Вам не нравятся шрифты в 10, 7 и 5 пунктов, можно их заменить командами `\textfont`, `\scriptfont` и `\scriptscriptfont`. Команда примитивного Т<sub>Е</sub>X’а. Прочная.

`\scriptsize` Изменяет размер шрифта, а в этом размере выбирает романский шрифт. Нельзя использовать в математической моде. Обычно выбираемый размер больше `\tiny` и меньше `\footnotesize`. См. рис.14 на стр.62. Хрупкая.

`\scriptstyle` Декларация, которую можно использовать только в математической моде. Заставляет Т<sub>Е</sub>X набирать формулы в индексном стиле, который по умолчанию используется, например, для набора индексов в выключенных формулах и в формулах в тексте. Не только задает шрифт меньшего размера, но и приводит к другому распределению пробелов вокруг символов отношений и бинарных операторов. См. также `\scriptscriptstyle`.

$$\text{\$}\{1\over\Gamma(z)\}=ze^{\text{\scriptstyle } Cz}\prod_1^{\infty}\text{\Bigl}1+\text{\scriptstyle } z\over n\text{\Bigr}e^{\text{\scriptstyle } z\over\text{\scriptstyle } n}\text{\$}$$

$$\frac{1}{\Gamma(z)} = ze^{Cz} \prod_1^{\infty} \left(1 + \frac{z}{n}\right) e^{-\frac{z}{n}}$$

В формуле выше дробь `{z\over n}` является показателем степени, следовательно, печатается в `\scriptstyle`. Если бы не было `\scriptstyle` внутри дроби, то ‘z’ и ‘n’ были бы в `\scriptscriptstyle`, т. е., печатались бы шрифтом в 5 пунктов, а не 7 пунктов. Разборчивость и уравновешенность формулы нарушились бы:

$$e^{\text{\scriptstyle } z\over n}\quad e^{\text{\scriptstyle } z\over\text{\scriptstyle } n}\text{\$} \dots\dots\dots e^{\frac{z}{n}} \quad e^{\frac{z}{n}}$$

Команда примитивного Т<sub>Е</sub>X’а. Пример ее использования подготовлен с использованием команд plain Т<sub>Е</sub>Xа, которые в этом руководстве не описаны. Прочная.

`\scrollmode` Команда примитивного Т<sub>Е</sub>X’а; задает такой режим обработки входного файла, при котором работа не прекратится ни при обнаружении ошибки, ни при введенном требовании пользователя, хотя остановка все же может произойти при попытке включить командой `\input` несуществующий

файл. Одна из немногих команд, которые во входном файле могут стоять перед командой `\documentstyle`. См. также `\batchmode`, `\errorstopmode` и `\nonstopmode`.

`\searrow` (*south east arrow*). Производит символ бинарного отношения  $\searrow$ , но только в математической моде. Прочная.

`\sec` Производит обозначение тригонометрической функции ( $\sec$ ), но только в математической моде. Если этот символ имеет нижний или верхний индексы, они никогда не появляются в виде пределов. Прочная.

`secnumdepth` Счетчик, который указывает, какие единицы рубрикации должны нумероваться. Нумеруются все единицы рубрикации, номер уровня которых меньше или равен значению этого счетчика. См. рис.8 на стр.24 и рис.5 на стр.33.

`section` Счетчик для управления нумерацией секций. Он устанавливается в нуль и увеличивается командой `\section` перед генерацией номера. Значение счетчику можно присвоить командой `\setcounter`. В стилях документа `report` и `book` он устанавливается в нуль командой `\chapter`. Так, первая секция главы X всегда будет иметь номер X.1.

`\section[entry]{heading}` Команда рубрикации, которая открывает новую секцию. В стилях документа `article`, `report` и `book` секции имеют номер уровня 1. В этих стилях секции по умолчанию нумеруются автоматически. (Можно изменить, меняя значение счетчика `secnumdepth`.) Если отсутствует необязательный аргумент *entry*, то в оглавление, если оно производится, заносится *heading*. (Можно изменить, меняя счетчик `tocdepth`.) Если же *entry* присутствует, то он заносится в оглавление, а заголовок *heading* появляется в теле выходного документа. Если присутствует *entry*, то он является подвижным аргументом; если же *entry* отсутствует, подвижным будет аргумент *heading*. Хрупкая.

`\section*{heading}` Команда рубрикации, которая открывает новую секцию. Полученный заголовок не нумеруется и не заносится в оглавление. Хрупкая.

`series` (Имя поля ВВТ<sub>E</sub>X'a.) Некоторые книги издаются сериями. Здесь помещается название такой серии.

`\setcounter{ctr}{i}` Глобальная декларация, присваивающая счетчику *ctr* значение *i*. Пример в описании команды `\thectr`. Хрупкая.

`\setlength{cmd}{len}` Присваивает команде длины *cmd* — которая должна начинаться с бэкслэша — значение *len*. Например, после команды `\setlength{\topmargin}{2in}` параметр `\topmargin` будет равен двум дюймам. Прочная.

`\setminus` Производит символ бинарного оператора  $\setminus$ , но только в математической моде. Команда `\backslash` производит тот же самый знак, но в качестве ординарного символа. Прочная.

|               | Д<br>э<br>в<br>и<br>с | Х<br>е<br>н<br>д<br>р<br>и | П<br>а<br>р<br>р<br>о<br>т<br>т | У<br>а<br>й<br>т |
|---------------|-----------------------|----------------------------|---------------------------------|------------------|
| Стив Дэвис    | ×                     | 23                         | 19                              | 0                |
| Стефен Хендри | 34                    | ×                          | 0                               | 4                |
| Джон Парротт  | 22                    | 0                          | ×                               | 18               |
| Джимми Уайт   | 0                     | 10                         | 11                              | ×                |

Рис. 22. Результаты игр в бильярд между четырьмя лучшими игроками в классификационных соревнованиях сезона 1990–1991 годов.

`\settowidth{cmd}{text}` Присваивает команде длины *cmd* — которая должна начинаться с бэкслэша — естественную ширину результата обработки аргумента *text* в LR моде. Например, после команд

```
\newlength{\ee}
\settowidth{\ee}{[1996 год]}
```

значением новой команды длины `\ee` будет длина текста “1996 год”. Прочная.

`\sf` Декларация, меняющая текущий шрифт на рубленый шрифт — см. рис.9 на стр.28. Прочная.

`\sharp` Производит ординарный символ  $\sharp$ , но только в математической моде. Прочная.

`\shortstack[pos]{col}` Обычно используется для получения текста, расположенного на странице вертикально. Элементы в *col* разделяются командами `\`, а аргумент *pos* может быть равен *l* (когда элементы ряда прижимаются влево), *r* (когда элементы ряда прижимаются вправо) или, по умолчанию, *c* (для центрирования элементов ряда). Например, таблица на рис.22, стр.124 была получена следующими командами:

```
\begin{figure}[t]
\begin{center}
\def\temp#1{\multicolumn{1}{|r|}{#1}}
\begin{tabular}{r|c|c|c|c|} \cline{2-5}
& \shortstack{Д\a\v\и\с}
& \shortstack{Х\е\н\д\р\и}
& \shortstack{\rule{0mm}{1mm}\Pa\p\p\o\т\т}
& \shortstack{У\а\й\т} \ \ \hline
\end{tabular}
\end{center}
\end{figure}
```

```

\temp{Стив Дэвис} & \times & 23 & 19 & 0 \\ \hline
\temp{Стефен Хендри} & 34 & \times & 0 & 4 \\ \hline
\temp{Джон Парротт} & 22 & 0 & \times & 18 \\ \hline
\temp{Джимми Уайт} & 0 & 10 & 11 & \times \\ \hline
\end{tabular}
\end{center}
\caption{Результаты игр в бильярд между четырьмя
лучшими игроками в классификационных соревнованиях сезона
1990--1991~годов.}
\label{stack}
\end{figure}

```

Если команда используется внутри командных скобок `picture`, точкой привязки является нижний левый угол получаемого бокса. Хрупкая.

`\sigma` Производит ординарный символ  $\sigma$ , но только в математической моде. Ординарный символ  $\varsigma$  получается командой `\varsigma`. Прочная.

`\Sigma` Производит ординарный символ  $\Sigma$ , но только в математической моде. Не путайте с символом суммы (который является большим оператором).

`$$\Sigma$`, `$$\sum$`, `$$\displaystyle\sum$` .....  $\Sigma$ ,  $\sum$ ,  $\sum$   
Прочная.

`\signature{text}` Декларация, которую можно использовать только в стиле `letter`. Указывает, что `text` является отправителем письма. Для того, чтобы в аргументе `text` начать новую строку, можно использовать команду `\\`. Пример использования этой команды можно видеть на рис.16 на стр.79, а результат — на рис. 17 на стр.80.

`\sim` Производит символ бинарного отношения  $\sim$ , но только в математической моде. Прочная.

`\simeq` Производит символ бинарного отношения  $\simeq$ , но только в математической моде. Также имеются отношения `\cong` и `\approx`, которые дают  $\cong$  и  $\approx$ . Прочная.

`\sin` Производит обозначение тригонометрической функции (`sin`), но только в математической моде.

`{1\over\sin^2z}=\sum_{-\infty}^{+\infty}{1\over(z-n\pi)^2}`

$$\frac{1}{\sin^2 z} = \sum_{-\infty}^{+\infty} \frac{1}{(z - n\pi)^2}$$

Индексы у этого символа никогда не появляются в виде пределов. Прочная.

- `\sinh` Производит обозначение тригонометрической функции ( $\sinh$ ), но только в математической моде. См. также `\cosh`. Если Вы предпочитаете обозначение “sh”, а не “sinh”, используйте определение
- ```
\def\sh{\mathop{\rm sh}\nolimits}
```
- Индексы у этого символа никогда не появляются в виде пределов. Прочная.
- `\sl` Декларация, которая изменяет шрифт, задавая *наклонный*. Прочная.
- `\sloppy` Действует на разбиение строк: текст очень редко залезает на правое поле, но могут встречаться большие интервалы между словами. Если декларация встречается в конце абзаца, то действует на весь абзац.
- `sloppypar` Командные скобки, которые набирают материал, помещенный внутри их, в абзацной моде и под воздействием декларации `\sloppy`.
- `\small` Декларация, которая изменяет размер шрифта и в этом размере выбирает романский шрифт. Обычно выбранный размер шрифта больше, чем `\footnotesize` и меньше, чем `\normalsize`. См. рис.14 на стр.62. Хрупкая.
- `\smallskip` Производит вертикальный пробел, высота которого задана параметром `\smallskipamount`. Определена как `\vspace{\smallskipamount}`. Заметим, что это отличается от ее определения в plain TeX'e. Хрупкая.
- `\smallskipamount` Эластичная длина, значение которой равно `3pt plus 1pt minus 1pt`.
- `\smash{text}` Это команда plain TeX'a. Набирает аргумент *text* обычным образом, но TeX трактует результат, как если бы он имел ширину и высоту, равные нулю и именно эти значения высоты и глубины использует для своих вычислений по формированию страницы. Вот одно из применений: если ввести `\overrightarrow{AB}`, строка, которая содержит вектор \overrightarrow{AB} , немного опускается (как это происходит сейчас). Результат неприятный. Лучше написать `\smash{\overrightarrow{AB}}`, что дает \overrightarrow{AB} . На этот раз строка не отодвигается от предыдущей.
- `\smile` Производит символ бинарного отношения \smile , но только в математической моде. Прочная.
- `sp` Ключевое слово TeX'a для масштабного пункта (*scaled point*), единицы длины, которую TeX использует для внутренних нужд. Удовлетворяет равенству $1 \text{ pt} = 2^{16} \text{ sp} (= 65,536 \text{ sp})$.
- `\sp` Альтернативный способ получения верхнего индекса. Как `x^{75}` , так и `$x\sp{75}$` оба дают x^{75} . См. `\sb`.
- `\space` Когда используется в аргументе команды `\typeout`, производит на терминале один пробел.

`\spadesuit` Производит ординарный символ ♠, но только в математической моде. Прочная.

`\sqcap` Производит символ бинарного оператора \sqcap , но только в математической моде. Прочная.

`\sqcup` Производит символ бинарного оператора \sqcup , но только в математической моде. Прочная.

`\sqrt[n]{form}` Производит $\sqrt[n]{form}$, если n присутствует, и \sqrt{form} , если n отсутствует, но только в математической моде. Для получения радикалов можно использовать и команду plain TeXa `\root` — см. ее описание. Хрупкая.

`\sqsubset` Производит символ бинарного отношения \sqsubset , но только в математической моде. Заметим, что в plain TeX'e такой команды нет. Прочная.

`\sqsubseteq` Производит символ бинарного отношения \sqsubseteq , но только в математической моде. Прочная.

`\sqsupset` Производит символ бинарного отношения \sqsupset , но только в математической моде. Заметим, что в plain TeX'e такой команды нет. Прочная.

`\sqsupseteq` Производит символ бинарного отношения \sqsupseteq , но только в математической моде. Прочная.

`\ss` Доступна только в абзацной и LR модах для получения немецкой буквы 'эс-зет' (ß). Прочная.

`\stackrel{top}{bot}` Можно использовать только в математической моде, где производится символ для бинарного отношения, в котором top помещается над bot . Заметим, что top обрабатывается в том же стиле, что и верхние индексы. Например, если ввести `\stackrel{\wedge}{=}`, то получится $\overset{\wedge}{=}$. Имеется также команда plain TeXa `\buildrel` — см. ее описание. Прочная.

`\star` Производит символ бинарного оператора \star , но только в математической моде. Прочная.

`\stepcounter{ctr}` Значение счетчика ctr увеличивается на 1, а все вложенные счетчики устанавливаются в нуль.

`\stop` Если обработка файла приостановится из-за ошибки, можно ввести `I\stop` (без всяких пробелов), что прекратит обработку входного файла.

`\stretch{fpr}` Эластичная длина, естественная величина которой равна нулю дюймам, а растяжимость в fpr раз больше, чем `\fill`, где fpr — это вещественное число, которое может быть как положительным, так и отрицательным. Прочная.

`\strut` Доступная во всех модах команда plain TeX'a; производит невидимый вертикальный отрезок, высота которого равна значению `\baselineskip`, а ширина равна нулю. Не путайте `\strut` с `\mathstrut` (высота которой такая же, как у круглой скобки). `\mathstrut` меньше, чем `\strut` и посмотреть на нее можно в ее описании. Невидимая черта очень полезна для единообразия некоторых кодировок (см. `\bar`) и для получения правильных промежутков между строками таблиц. См. также формулу на рис.1 на стр.10.

`.sty` Расширение имени для стилевого файла (такого как `report.sty` или `rep11.sty`), который содержит набор команд и макрокоманд.

`\subitem` Внутри командных скобок `theindex` производит элемент с отступом.

`\subsubitem` Команда, которая используется внутри командных скобок `theindex` для получения элемента, отступ у которого больше, чем отступ, производимый командой `\subitem`.

`subparagraph` Счетчик для управления нумерацией подразделов. Устанавливается в нуль и увеличивается командой `\subparagraph` перед генерацией номера. Значение ему можно присваивать командой `\setcounter`. Счетчик `subparagraph` устанавливается в нуль командами `\chapter`, `\section`, `\subsection`, `\subsubsection` и `\paragraph`.

`\subparagraph[entry]{heading}` Команда рубрикации, которая открывает новый подраздел. Подразделы в стилях документа `article`, `report` и `book` имеют номер уровня 5. В этих стилях подразделы по умолчанию автоматически не нумеруются. (Можно изменить, меняя значение счетчика `secnumdepth`.) По умолчанию заголовок `heading` не заносится в оглавление, если оно производится, но это также можно изменить, изменяя значение счетчика `tocdepth`. Если заголовок подраздела должен быть занесен в оглавление, то туда заносится `entry`, если он присутствует, а если он отсутствует, то в оглавление заносится `heading`. Если `entry` присутствует, то он является подвижным аргументом, а если он отсутствует, то подвижным является аргумент `heading`. Хрупкая.

`\subparagraph*{heading}` Команда рубрикации, открывающая новый подраздел, который не нумеруется и не заносится в оглавление. Хрупкая.

`subsection` Счетчик для управления нумерацией подсекций. Устанавливается в нуль и увеличивается командой `\subsection` перед генерацией номера. Значение счетчику можно присвоить командой `\setcounter`. Устанавливается в нуль командами `\chapter` и `\section`.

`\subsection[entry]{heading}` Команда рубрикации, которая открывает новую подсекцию. Подсекции в стилях документа `article`, `report` и `book` имеют номер уровня 2. В этих стилях документа подсекции по умолчанию автоматически нумеруются. (Можно изменить, меняя значение счетчика

`secnumdepth`.) По умолчанию заголовки *heading* не будут появляться в оглавлении, если оно производится, но это можно изменить, меняя значение счетчика `tocdepth`. Если заголовок подсекции должен появиться в оглавлении, то для этого используется *heading*, если отсутствует необязательный аргумент *entry*, но если присутствует *entry*, то в оглавление заносится он. Если *entry* присутствует, то он является подвижным аргументом. Если же он отсутствует, то подвижным является аргумент *heading*. Хрупкая.

`\subsection*{heading}` Команда рубрикации, открывающая новую подсекцию, которая не нумеруется и не заносится в оглавление. Хрупкая.

`\subset` Производит символ бинарного отношения \subset , но только в математической моде. Прочная.

`\subseteq` Производит символ бинарного отношения \subseteq , но только в математической моде. Прочная.

`subsection` Счетчик для управления нумерацией подподсекций. Инициализируется в нуль и увеличивается командой `\subsection` перед генерацией номера. Значение этому счетчику можно присвоить командой `\setcounter`. Значение `subsection` устанавливается в нуль командами `\chapter`, `\section` и `\subsection`.

`\subsection[entry]{heading}` Команда рубрикации, которая открывает новую подподсекцию. В стилях документа `article`, `report` и `book` подподсекции имеют номер уровня 3. Подподсекции по умолчанию автоматически нумеруются в стиле документа `article` и не нумеруются в стилях документа `report` и `book`. (Можно изменить, меняя значение счетчика `secnumdepth`.) По умолчанию заголовки *heading* появятся в оглавлении, если оно производится, в стиле `article`, и не появятся в стиле `report` или `book`. (Можно изменить, меняя значение счетчика `tocdepth`. Если в оглавлении должен появиться заголовок подподсекции, то для этого используется *heading*, если только не присутствует *entry* — тогда в оглавление заносится он. Если *entry* присутствует, он является подвижным аргументом. Если же *entry* отсутствует, подвижным будет аргумент *heading*. Хрупкая.

`\subsection*{heading}` Команда рубрикации, открывающая новую подподсекцию, которая не нумеруется и не заносится в оглавление. Хрупкая.

`\succ` Производит символ бинарного отношения \succ , но только в математической моде. Прочная.

`\succeq` Производит символ бинарного отношения \succeq , но только в математической моде. Прочная.

`\sum` Производит символ большого оператора Σ и \sum . Только в математической моде. Не путать с буквой Σ :

`\Sigma`, `\sum`, `\displaystyle\sum` Σ , \sum , \sum

В выключенных формулах размер символа больше, чем в формулах в тексте. Кроме того, в выключенных формулах индексы печатаются над и под этим символом в виде пределов. Прочная.

`\sup` Производит обозначение (\sup), но только в математической моде. Если символ встречается в выключенной формуле, индексы печатаются над и под ним — в виде пределов. См. `\inf`. Прочная.

`\supset` Производит символ бинарного отношения \supset , но только в математической моде. Прочная.

`\supseteq` Производит символ бинарного отношения \supseteq , но только в математической моде. Прочная.

`\surd` Производит ординарный символ \surd , но только в математической моде.
`\surd a+\surd\,(b+1)` $\surd a + \surd(b+1)$
 См. также `\sqrt`. Прочная.

`\swarrow` (south west arrow). Производит символ бинарного отношения \swarrow , но только в математической моде. Прочная.

`\symbol{i}` Производит символ текущего шрифта с номером i . Так, `\tt \symbol{92}char` дает `\char`. Если перед i стоит `'`, то номер считается восьмеричным, а если `"` — то шестнадцатеричным. Прочная.

t

`\tchar1char2` или `\t{char1char2}` Конструкция производит акцент связку ($\textcircled{\text{oo}}$) над следующими за ним двумя символами $char_1$ и $char_2$ в LR и абзацной моде. Так, `T\t uut` дает $T\textcircled{\text{oo}}ut$. Прочная.

`tabbing` Командные скобки для получения выровненной информации без возможности включения горизонтальных или вертикальных прямых. Основная идея показана на следующем простом примере:

```

\begin{tabbing}
123\=456\=789\=\kill
один      \>      один  \\  

два       \>      два   \\  

   три    \>\>    три   \\  

   четыре \>\>   четыре \\  

     пять \>\>\>  пять  \\  

     шесть \>\>\>  шесть  \\  

     семь  \>\>   семь   \\  

    восемь \>\>   восемь  \\  

    девять \>     девять  \\  

    десять \>     десять   \\  

\end{tabbing}

```

Здесь первая строка внутри командных скобок `tabbing` устанавливает позиции табуляции — так же, как позиции табуляции на пишущей машинке — а команда `\kill` просто говорит Л^AT_EX’у не производить никакого выходного результата, соответствующего этой строке. Команды `\=` устанавливают позиции табуляции. Заметим, что вне командных скобок `tabbing` команда `\=` делает акцент макрон над следующим за ней символом. Если Вам нужен акцент макрон внутри командных скобок `tabbing`, следует использовать команду `\a=`. Также заметим, что в шрифтах Computer modern, разработанных Кнудом, все цифры имеют ширину, равную половине em. Команда `\>` перемещает начало следующего за ней текста на очередную позицию табуляции.¹⁰

Внутри командных скобок `tabbing` Л^AT_EX поддерживает две переменные, а именно, *next-tab-stop* (следующий таб-стоп) и *left-margin-tab* (таб-левого-поля), значениями которых являются неотрицательные целые числа. Позиции табуляции представляют собой воображаемые вертикальные прямые, вдоль которых выравнивается текст. Каждая такая прямая последовательно нумеруется слева направо, начиная с нуля. В приведенном выше примере заданы три позиции табуляции с номерами 0, 1 и 2. В начале значение *left-margin-tab* равно нулю и позиционирование происходит на текущем левом поле, а начальное значение *next-tab-stop* устанавливается в 1. Команда `\>` выравнивает текст по линии *i* — если значение *next-tab-stop* равно *i* — и увеличивает *next-tab-stop* на 1. (Заметим, что команда конца строки `\` не только прерывает строку, но и устанавливает *next-tab-stop* в $1 + \textit{left-margin-tab}$; на *left-margin-tab* она не действует.)

`\tabbingsep` Когда в командных скобках `tabbing` используется команда `\`, текст выталкивается влево, при этом значение этого параметра жесткой длины задает расстояние, оставляемое между текстом и левым полем (или подхо-

¹⁰В plain T_EX’e команду `\>` можно использовать только в математической моде и там она обозначает пробел средней величины. В Л^AT_EX’e для этой цели служит команда `\:`, но только в математической моде.

дящей позицией табуляции). Прочная команда, перед которой никогда не надо ставить `\protect`.

`\tabcolsep` Параметр жесткой длины, равный половине величины горизонтального пробела, оставляемого между колонками, которые производятся командными скобками `tabular` или `tabular*`. Прочная команда, поэтому перед ней не надо ставить команду `\protect`.

`table` Счетчик для нумерации плавающих вставок, создаваемых командными скобками `tabular` или `tabular*`. Увеличивается, только если внутри командных скобок имеется команда `\caption`.

`table` Командные скобки, которые производят плавающую вставку. Если внутри них присутствует команда `\caption`, то автоматически генерируется слово ‘Table’¹¹ и числовая метка; в противном случае они ведут себя как командные скобки `figure` (см. их описание).

`table*` Очень похожи на командные скобки `table`, за тем исключением, что при опции документа `twocolumn`, таблица будет иметь ширину в две колонки.

`\tableofcontents` Команда производит оглавление в том месте входного файла, где она встретилась. Чтобы получить правильное оглавление, надо пропустить `ЛATEX` как минимум дважды. Приводит к тому, что записывается (или перезаписывается) файл с расширением `toc`, если, конечно, нет команды `\nofiles`.

`tabular` Командные скобки `tabular` могут встретиться в любой моде и имеют синтаксис:

```
\begin{tabular}[pos]{preamble}row-list \end{tabular}
```

Необязательный аргумент *pos* управляет вертикальным положением получаемого бокса. По умолчанию выравнивание происходит по центру бокса, опция `t` выравнивает бокс по его верхнему ряду, а опция `b` — по нижнему ряду. Аргумент *preamble* задает, как надо форматировать колонки таблицы. Элемент *row-list* состоит из одного или нескольких рядов *row*, которые разделяются командами `\\`. В каждом ряду обычно имеется $i - 1$ амперсанд, где i — число колонок в таблице. (Заметим, что если ряд содержит команды `\multicolumn`, то амперсандов требуется меньше.) После команды `\\` могут стоять команды `\hline`. Если две команды `\hline` следуют подряд, то вертикальный пробел, разделяющий их, задается жестким параметром длины `\doubleseprule`. Если Вы хотите, чтобы в конце Вашей таблицы появилась черта, то перед последней командой `\hline` должна стоять команда `\\`. Но она может быть опущена, если за ней не следует команда `\hline`. Например, таблица

¹¹В русифицированном `ЛATEX`’е — ‘Таблица’.

Категория	Интуитивное значение	Типичный элемент
<i>Nml</i>	цифры	N
<i>UnOps</i>	унарные операторы	α
<i>BinOps</i>	бинарные операторы	ω
<i>Ide</i>	идентификаторы	I
<i>Exp</i>	выражения	E
<i>Cmd</i>	команды	C

была получена следующими командами:

```

\begin{center}
\begin{tabular}{|c|c|c|} \hline
Категория      & Интуитивное значение& Типичный элемент\\
\hline \hline
 $\hbox{\it Nml}$  & Цифры                &  $N$                 & \\ \hline
 $\hbox{\it UnOps}$  & унарные операторы   &  $\alpha$            & \\ \hline
 $\hbox{\it BinOps}$  & бинарные операторы &  $\omega$           & \\ \hline
 $\hbox{\it Ide}$     & идентификаторы     &  $I$                 & \\ \hline
 $\hbox{\it Exp}$     & выражения          &  $E$                 & \\ \hline
 $\hbox{\it Cmd}$     & команды            &  $C$                 & \\ \hline
\end{tabular}
\end{center}

```

Они заключены в командные скобки `center` только для того, чтобы результирующая таблица центрировалась на странице. Без этого таблица была бы прижата к левому краю страницы.¹²

Интересный пример командных скобок `tabular` можно найти в описании команды `\shortstack`

tabular* Синтаксис командных скобок `tabular*`:

```
\begin{tabular*}{len}[pos]{preamble}row-list \end{tabular*}
```

где *pos*, *preamble* и *row-list* такие же, как в командных скобках `tabular`, а *len* задает ширину производимого бокса. Надо позаботиться, чтобы материал, который размещается в каждом ряду, имел именно эту ширину; обычно для этой цели используют команду эластичной длины `\hfill`.

\tan Производит обозначение тригонометрической функции (\tan), но только в математической моде. Индексы у символа никогда не появляются в виде пределов. Те, кому нравится обозначение tg , могут использовать определение:

¹²Заметим, что если преамбула начинается не с `|` и не с `@`-выражения, то таблица не будет прижата к левому краю страницы. Перед получаемым боксом будет вставлен горизонтальный пробел, равный параметру `\tabcolsep`. Такой же пробел будет вставлен и после получаемого бокса, если преамбула не оканчивается `|` или `@`-выражением. Поэтому, чтобы сдвинуть таблицу, можно поставить в начале или конце преамбулы `@{`.

`\def\tg{\mathop{\rm tg}\nolimits}}`

Прочная.

`\tanh` Производит обозначение тригонометрической функции (\tanh), но только в математической моде. Если же Вам нравится `th`, см. выше. Индексы у символа никогда не появляются в виде пределов. Прочная.

`\tau` Производит ординарный символ τ , но только в математической моде. Прочная.

`\TeX` Производит логограмму \TeX . Эта логограмма “приклеивается” к следующему слову, поэтому пробел после нее получается так: `{\TeX}`, `\TeX{}` или `\TeX\`.

`\textfloatsep` Параметр эластичной длины, равный величине вертикального пробела, оставляемого между вставкой и текстом над или под ней. Если выбрана опция стиля документа `twocolumn`, эта команда действует только на одноколончатые вставки; для двухколоночных вставок см. команду `\dbltextfloatsep`. Прочная команда, перед которой никогда не надо ставить `\protect`.

`\textfraction` Команда, значением которой является вещественное число между 0 и 1, представляющее собой минимальную часть текстовой страницы, то есть, страницы, содержащей и текст, и вставки, которая должна быть занята текстом. Например, если `\textfraction` равна 0.6, то как минимум 60% каждой текстовой страницы должно быть занято не вставками. Значение можно изменить командой `\def` или `\renewcommand`. По умолчанию оно равно 0.2.

`\textheight` Параметр жесткой длины, действующий на внешний вид каждой страницы результата. Равен нормальной высоте тела страницы — см. рис.9. Прочная команда, перед которой никогда не надо ставить `\protect`.

`\textstyle` Декларация, которую можно использовать только в математической моде. Заставляет \TeX набирать формулы в текстовом стиле, который по умолчанию принят для включенных в текст формул. Ее можно, например, использовать в индексах, чтобы они набирались более крупными символами. Другой стиль печати математики, при котором формулы выделяются на отдельных строках — это `\displaystyle`. В `\textstyle` \TeX старается не слишком раздвигать строки, т.е., соблюдать нормальное расстояние между ними. При внимательном рассмотрении формул, напечатанных в этом стиле, вы увидите, что индексы и показатели степеней не такие высокие, как в `\displaystyle`, что числитель и знаменатель ближе к дробной черте и набраны меньшим шрифтом, и т.д. В этом руководстве можно найти примеры формул в обоих стилях. См. также `\displaystyle`, `\scriptstyle`, `\limits` и `\nolimits`. Команда примитивного \TeX 'а. Прочная.

`\textwidth` Параметр жесткой длины, действующий на внешний вид каждой страницы выходного результата. Равен нормальной ширине тела страницы — см. рис.23, стр.137. Прочная команда, перед которой никогда не надо ставить `\protect`.

`\thanks{text}` Может стоять только внутри аргумента декларации `\author`, `\date` или `\title`. Используется для получения сноски, в качестве которой появляется *text*. Ширина меток таких сносок считается равной нулю, поэтому если команда `\thanks` не находится в конце строки, то после нее надо поместить команду `_`, чтобы вставить междусловный пробел. Слишком большое количество команд `\thanks` приведет к сообщению об ошибке: “Counter too large”. Заметим, что *text* является подвижным аргументом.

`\the ctr` Команда `\newcounter{ ctr }` делает ctr счетчиком, но чтобы вывести значение ctr , надо преобразовать его в текст, например, командами `\arabic` или `\Roman`. Когда Вы вводите новый счетчик, ЛАТ_EX автоматически создает новую команду `\the ctr` , которая производит текстовый результат; по умолчанию `\the ctr` определена как `\arabic{ ctr }`. Например, команды

```
\newcounter{instinct}
\setcounter{instinct}{7}
\theinstinct
```

производят 7, а команды

```
\setcounter{instinct}{11}
\def\theinstinct{\Alph{instinct}}
\theinstinct
```

производят К. В случаях встроенных счетчиков ЛАТ_EX'a результаты работы `\the ctr` сильно различаются.

`thebibliography` Командные скобки, которые используются для самостоятельной подготовки библиографии. Например, библиография к этому руководству была получена следующими командами, которые стоят в самом конце входного файла (непосредственно перед командой `\end{document}`):



```
\begin{thebibliography}
\bibitem{Knuth} Дональд Кнут,
{\it Все про \TeX}, перевод с англ.,
РДТЕХ, 1993.
\bibitem{Lamp} Leslie Lamport,
{\it \LaTeX\ A Document Preparation System},
Addison Wesley, 1985
. . .
\bibitem{Lis} Лисина М.В.,
```

```
{\it Plain \TeX\ Основные понятия и каталог команд},
Препринт ИФВЭ 95--58.
\end{thebibliography}
```

`\theindex` Командные скобки, которые используются для получения алфавитного указателя; результат набирается в двухколоночном формате.

`\theta` Производит ординарный символ θ , но только в математической моде. Ординарный символ ϑ получается командой `\vartheta`. Прочная.

`\Theta` Производит ординарный символ Θ , но только в математической моде. Прочная.

`\thicklines` Декларация для выбора самой большой толщины (из двух стандартных толщин) рисуемых линий в командных скобках `picture`. Для линий в рисунке L^AT_EX имеет две стандартных толщины — тонкая, как в , и толстая, как в . Толщина задается декларациями `\thinlines` и `\thicklines`. По умолчанию принимается `\thinlines`. Эти команды являются обычными декларациями и могут быть использованы в любое время. Прочная.

`\thinlines` Декларация для выбора самой маленькой толщины (из двух стандартных толщин) рисуемых линий в командных скобках `picture`, выбираемая по умолчанию. См. выше описание `\thicklines`. Прочная.

`\thispagestyle{page-style}` Декларация, аналогичная `\pagestyle`, за тем исключением, что действует только на текущую страницу. Параметром *page-style* может быть `plain`, `empty`, `headings` или `myheadings`. Хрупкая. Например, обратите внимание на отсутствие заголовка и номера на этой странице. Это происходит потому, что здесь задана декларация `\thispagestyle{empty}` (которая, кстати, обычно используется при оформлении титульного листа документа).

`\tiny` Изменяет размер шрифта и в новом размере задает романский шрифт. Нельзя использовать в математической моде. Обычно этот размер самый меньший из всех доступных размеров, даже меньше, чем `\scriptsize`. См. рис.14 на стр.62. Хрупкая.

`\tilde` Производит акцент в математической моде.

`\tilde x + \tilde y` $\tilde{x} + \tilde{y}$

Увеличенная `\widetilde` вытягивается вплоть до трех символов:

`\widetilde {xyz} = \widetilde{xy} * \tilde z` $\widetilde{xyz} = \widetilde{xy} * \tilde{z}$

Прочная.

`\times` Производит символ бинарного оператора \times , но только в математической моде. Прочная.

a	<code>\topmargin + 1 in</code>	f	<code>\footheight</code>
b	<code>\headheight</code>	g	<code>\oddsidemargin + 1 in</code>
c	<code>\headsep</code>	g	<code>\evensidemargin + 1 in</code>
d	<code>\textheight</code>	h	<code>\textwidth</code>
e	<code>\footskip - \footheight</code>		

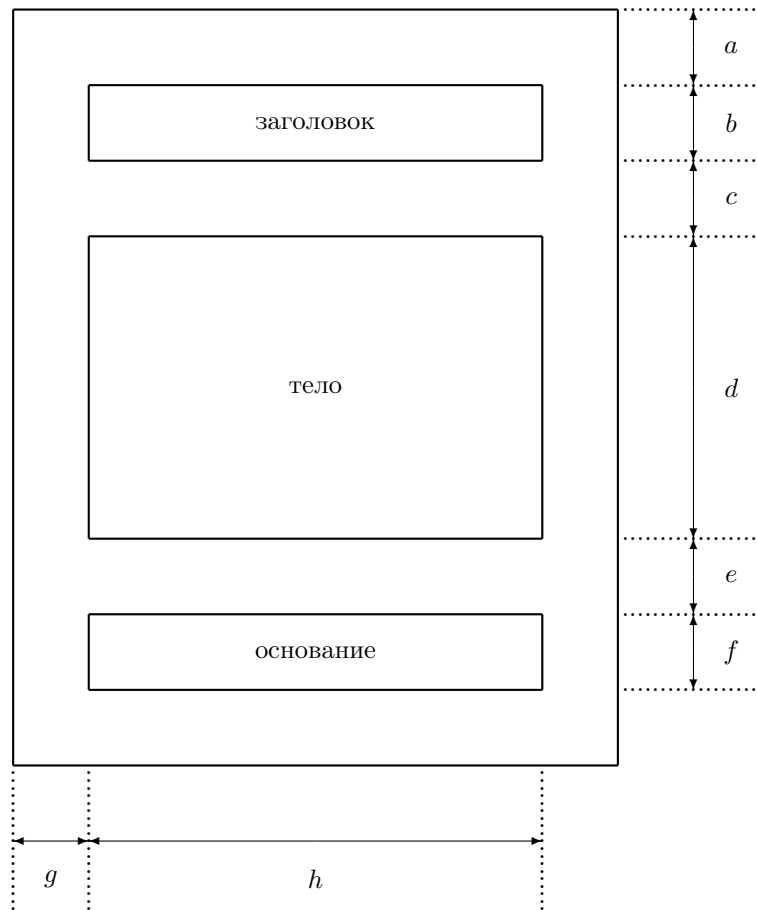


Рис. 23. Параметры, которые действуют на вид страницы результата

- `title` (Имя поля ВВТ_EX'a.) Название книги, статьи или еще чего-нибудь.
- `\title{text}` Используется в стилях документа `article`, `report` и `book` для задания заголовка. (Сам заголовок производится внутри командных скобок `document` командой `\maketitle` — см. рис.7 на стр.24.) Чтобы указать конец строки, можно в аргументе *text* использовать команду `\\`. В этом аргументе можно также использовать одну или несколько команд `\thanks`, которые производят сноски, ширина маркеров которых считается равной нулю. Если команда `\thanks` встречается не в конце строки, то чтобы вставить пробел, перед ней надо поместить команду `_`.
- `titlepage` Опция стиля документа, которая может использоваться, только если выбран стиль документа `article`. Приводит к тому, что и заголовок, и аннотация, если она имеется, помещаются вверху новой страницы.
- `\to` Производит символ бинарного отношения \rightarrow , но только в математической моде. Так же действует и команда `\rightarrow`. Прочная.
- `.toc` Расширение имени файла, который создается или перезаписывается, если входной файл содержит команду `\tableofcontents` (и не содержит команду `\nofiles`); `toc`-файл содержит информацию для получения оглавления документа. Информация поступает из всех тех команд рубрикации во входном файле, номер уровня которых меньше или равен значению счетчика `tocdepth`. Оглавление производится командой `\tableofcontents`, если во время обработки входного файла имеется `toc`-файл.
- `tocdepth` Счетчик, который управляет тем, какие заголовки единиц рубрикации появятся в оглавлении, если оно создается. В оглавлении появятся заголовки всех тех единиц рубрикации, номер уровня которых меньше или равен значению этого счетчика. См. рис.8 на стр.24 и рис.5 на стр.33.
- `\today` Указывает, что дата дня, когда Вы получали из входного файла `dvi`-файл, будет включена в этот `dvi`-файл. Дата имеет американский формат, например, `July 4, 1992`. Команду можно использовать только в LR или абзацной моде. Прочная.
- `\top` Производит ординарный символ \top , но только в математической моде. Также имеются `\perp`, `\dashv` и `\vdash`, которые дают \perp , \dashv и \vdash . Прочная.
- `\topfraction` Значением команды является вещественное число, расположенное между 0 и 1. Указывает, какая часть вверху каждой текстовой страницы может быть использована для плавающих вставок. Значение можно изменить командами `\def` или `\renewcommand`. По умолчанию оно равно 0.7. Если выбрана опция стиля документа `twocolumn`, то команда действует только на вставки шириной в одну колонку; для вставок шириной в две колонки см. `\dbltopfraction`.

- `\topmargin` Параметр жесткой длины, действующий на внешний вид каждой страницы результата. Расстояние между верхней границей листа бумаги и верхом заголовка страницы равно сумме значения `\topmargin` и одного дюйма. См. рис.23 на стр.137. Прочная команда, перед которой никогда не надо ставить `\protect`.
- `topnumber` Счетчик, равный максимальному числу плавающих вставок, то есть, таблиц и рисунков, которые могут встретиться вверху каждой текстовой страницы. По умолчанию равен 2. Если выбрана опция стиля документа `twocolumn`, счетчик действует только на вставки шириной в одну колонку; для вставок шириной в две колонки см. `dbltopnumber`.
- `\topsep` Параметр эластичной длины, который используется командными скобками `list` для управления внешним видом и организацией генерируемого списка. Действие этого параметра можно увидеть на рис.23, стр.137. Прочная команда, перед которой никогда не надо ставить `\protect`.
- `\topskip` Параметр эластичной длины, равный минимальному расстоянию между верхом тела и низом первой строки текста. Его действие похоже на `\baselineskip`, за тем исключением, что он влияет только на первую строку страницы. Прочная команда, перед которой никогда не надо ставить `\protect`.
- `totalnumber` Счетчик, равный максимальному числу плавающих вставок, которые могут помещаться на текстовой странице. По умолчанию равен 3.
- `\triangle` Производит ординарный символ Δ , но только в математической моде. Не путайте с греческой большой буквой `\Delta`:
 `Δ, \triangle Δ , \triangle`
и с бинарным отношением `\bigtriangleup`, которое похоже на символ `\triangle`, но окружено пробелами. Прочная.
- `\triangleleft` Производит ординарный символ \triangleleft , но только в математической моде. Прочная.
- `\triangleright` Производит ординарный символ \triangleright , но только в математической моде. Прочная.
- `trivlist` Командные скобки, являющиеся сокращенной формой командных скобок `list` — см. рис.18, стр.81. Сохраняют значения всех параметров, которые действуют при их открытии, за исключением того, что `\leftmargin`, `\labelwidth` и `\itemindent` устанавливаются равными 0 дюймам, а `\parsep` делается равным текущему значению `\parskip`.
- `\tt` Изменяет стиль печати и задает `typewriter` (“стиль пишущей машинки”) — см. рис.9 на стр.28. Учтите, что в этом стиле не делается перенос слов. Прочная.

`twocolumn` Опция стиля документа, при которой текст на каждой странице располагается в двух колонках.

`\twocolumn{text}` Глобальная декларация, то есть, ее область действия не ограничивается фигурными скобками. Сначала переходит на новую страницу, выполняя команду `\clearpage`, а затем набирает следующие данные в формате в две колонки. Если присутствует необязательный аргумент *text*, то он набирается в одноколонном абзацном боксе, занимающем всю ширину тела страницы. Хрупкая.

`twoside` Возможная опция команды `\documentstyle`. Указывает на то, что страницы с нечетными и четными номерами должны обрабатываться по-разному. Например, различаются принятые по умолчанию левые поля, а также по-разному форматируются бегущие заголовки страниц (если они имеются). Принята по умолчанию для стиля документа `book`.

`type` (Имя поля ВивТ_EX'а.) Имеется много видов технических отчетов. Здесь надо указать, на какой из них делается ссылка.

`\typein[cmd]{text}` Выводит *text* на терминал, а также записывает его в лог-файл. Аргумент *text* является подвижным. Если в аргумент *text* поставить имя команды вместе с `\protect`, то будет выведено имя команды. Имя команды в *text*, определенное декларациями `\newcommand` или `\renewcommand`, замещается на ее определение. Например, после команд

```
\typein[\wife]{Enter wife's name.}
I love \wife\ very much.
```

на экране компьютера появится следующее сообщение:

```
Enter wife's name.
\wife=
```

Если в ответ на приглашение вы введете, например, Marina и нажмете клавишу return, то значением `\wife` станет Marina, а в документе появится текст "I love Marina very much.". Хрупкая.

`\typeout{text}` Выводит *text* на терминал компьютера, а также записывает его в лог-файл. Например:

```
\typeout{Don't foget to revise this!}
```

Аргумент *text* является подвижным. Хрупкая.

u

`\uchar` или `\u{char}` Производит акцент (ö) над следующим одним символом *char* в LR или абзацной моде: `\u a`, `\u e`, `\u i` дают ä, ё и і. Прочная.

`\unboldmath` Отменяет действие предыдущей декларации `\boldmath`. Хрупкая.

`\underbrace{form}` Производит \underbrace{form} , но только в математической моде. Например:

$$\begin{aligned} & \text{\$}\displaystyle\underbrace{a-2b}_{>0} \\ & +\underbrace{y-z+t}_{>0}>0\$ \dots\dots\dots \underbrace{a-2b}_{>0} + \underbrace{y-z+t}_{>0} > 0 \end{aligned}$$

Нематематический аналог этой команды — `\downbracefill` — см. рис.13 на стр.52. Горизонтальная фигурная скобка сверху называется `\overbrace` в математической моде и `\upbracefill` в текстовой. В выключенных формулах нижний индекс производит метку под фигурной скобкой. Прочная.

`\underline{form}` Производит \underline{form} во всех модах.

$$\text{\$}\underline{b+q}=\underline{b}+\underline{q}\$ \dots\dots\dots \underline{b+q} = \underline{b} + \underline{q}$$

Чтобы подчеркивание было на одном уровне, используйте `\strut` или другие определенные Вами невидимые вертикальные черты:

$$\begin{aligned} & \text{\def\strit{\vrule depth 1.5pt width 0pt}} \\ & \text{\$}\underline{\strit b+q}=\underline{\strit b}+\underline{\strit q}\$ \dots\dots\dots \underline{b+q} = \underline{b} + \underline{q} \end{aligned}$$

Хрупкая.

`\unitlength` Параметр жесткой длины, определяющий единицу, которая в командных скобках `picture` используется для позиционирования и рисования объектов рисунка. Например, если перед командными скобками `picture` поместить команду `\unitlength=1mm`, единица длины будет равна одному миллиметру. По умолчанию ее значение равно 1 пункту. Примеры можно найти в описании команд `\circle`, `\line` и других команд, используемых в командных скобках `picture`. Прочная команда, перед которой никогда не надо ставить `\protect`.

`\unlhd` Производит символ бинарного оператора \triangleleft , но только в математической моде. Заметим, что в plain TeX'e такой команды нет. Прочная.

`\unrhd` Производит символ бинарного оператора \triangleright , но только в математической моде. Заметим, что в plain TeX'e такой команды нет. Прочная.

`unsrt` Возможный аргумент команды `\bibliographystyle`. Элементы библиографии размещаются в порядке их первого цитирования и помечаются числами так: [31].

`\uparrow` Производит символ бинарного отношения \uparrow , но только в математической моде. Однако после команд `\left` или `\right` команда производит ограничитель, размер которого зависит от размера того, что он ограничивает. Прочная.

`\Uparrow` Производит символ бинарного отношения \Uparrow , но только в математической моде. Однако после команд `\left` или `\right` команда производит ограничитель, размер которого зависит от размера того, что он ограничивает. Прочная.

`\upbracefill` Производит фигурную скобку, направленную острием вниз, которая растягивается на все отведенное ей расстояние. Графически это показано на рис.13, стр.52.

`\updownarrow` Производит символ бинарного отношения \updownarrow , но только в математической моде. Однако после команд `\left` или `\right` команда производит ограничитель, размер которого зависит от размера того, что он ограничивает. Прочная.

`\Updownarrow` Производит символ бинарного отношения \Updownarrow , но только в математической моде. Однако после команд `\left` или `\right` команда производит ограничитель, размер которого зависит от размера того, что он ограничивает. Прочная.

`\uplus` Производит символ бинарного оператора \uplus , но только в математической моде.

`\$ \uplus \$`, `\$ \biguplus \$`, `\$ \displaystyle \biguplus \$` \uplus , \biguplus , \biguplus

Прочная.

`\usebox{cmd}` Печатает содержимое ячейки памяти `cmd`. См. пример в команде `\savebox`. Прочная.

`\usecounter{ctr}` Декларация, которая может встретиться только в аргументе `dec-list` (список деклараций) командных скобок `list`. Используется для автоматической нумерации элементов списка.

`\upsilon` Производит ординарный символ υ , но только в математической моде. Она очень похожа на букву 'v', когда та печатается в математической моде: `\$v\$` и `\$ \upsilon \$` дают v и υ . Прочная.

`\Upsilon` Производит ординарный символ Υ , но только в математической моде. Прочная.

V

`\vchar` или `\v{char}` Производит акцент (̂) над следующим одним символом `char` в LR или абзацной моде: `\v Cebv\v sev` дает $\check{C}ebv\check{s}ev$. Это нематематическая версия макрокоманды `\check`. Прочная.

`\value{ctr}` Используется для задания значения одного счетчика равным значению другого. Например, `\setcounter{war}{\value{equation}}` делает значение счетчика `war` равным значению счетчика `equation`. (Счетчик `war` должен быть предварительно задан командой `\newcounter{war}`.) Прочная команда, перед которой никогда не надо ставить `\protect`.

`\varepsilon` Производит ординарный символ ε , но только в математической моде. Ординарный символ ϵ получается командой `\epsilon`, а символ бинарного отношения \in — командой `\in`. Прочная.

`\varphi` Производит ординарный символ φ , но только в математической моде. Ординарный символ ϕ получается командой `\phi`. Прочная.

`\varpi` Производит ординарный символ ϖ , но только в математической моде. Ординарный символ π получается командой `\pi`. Прочная.

`\varrho` Производит ординарный символ ϱ , но только в математической моде. Ординарный символ ρ получается командой `\rho`. Прочная.

`\varsigma` Производит ординарный символ ς , но только в математической моде. Ординарный символ σ получается командой `\sigma`. Прочная.

`\vartheta` Производит ординарный символ ϑ , но только в математической моде. Ординарный символ θ получается командой `\theta`. Прочная.

`\vbox{text}` Производит вертикальный бокс. Боксы, которые находятся в `text`, накладываются друг на друга вертикально. Ширина `\vbox` равна наибольшей ширине входящих в него боксов. Следовательно, ширина бокса `\vbox`, который содержит по крайней мере один символ, равна `\hsize`. Команда примитивного Т_EX'a. Пример ее применения можно посмотреть в описании `\downarrow`.

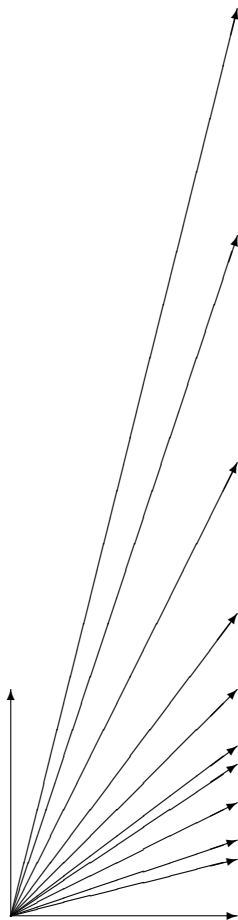
`\vdash` Производит символ бинарного отношения \vdash , но только в математической моде. Имеется также отношение `\dashv` : \dashv . Прочная.

`\vdots` Доступна только в математической моде для получения вертикального многоточия \ddots . Прочная.

`\vec` Помещает маленькую стрелку над следующим символом в математической моде:

`$(\vec\imath,\vec\jmath,\vec k)$` $(\vec{i}, \vec{j}, \vec{k})$

Имеется и макрокоманда `\overrightarrow`, которая автоматически вычисляет длину своей стрелки:



```

\unitlength=0.2mm
\begin{picture}(165,450)(0,-10)
\put(0,0){\vector(0,1){150}}
\put(0,0){\vector(1,0){150}}
\put(0,0){\vector(4,1){150}}
\put(0,0){\vector(3,1){150}}
\put(0,0){\vector(2,1){150}}
\put(0,0){\vector(1,1){150}}
\put(0,0){\vector(3,2){150}}
\put(0,0){\vector(1,2){150}}
\put(0,0){\vector(4,3){150}}
\put(0,0){\vector(1,3){150}}
\put(0,0){\vector(3,4){150}}
\put(0,0){\vector(1,4){150}}
\end{picture}

```

Рис. 24. Наклоны векторов, допустимые в командных скобках `picture` — см. команду `\vector`.

\overrightarrow{AB} , \vec{AB} \overline{AB} , \overline{AB}

Прочная.

`\vector` Может встретиться только в аргументах команд `\put` и `\multiput` в командных скобках `picture`. Команда

`\put(i,j){\vector(p,q){l}}`

рисует стрелку, которая начинается в точке (i,j) , а ее проекция на ось x равна l единиц. (Исключение составляют вертикальные стрелки, у которых l — это их длина.) Наклон стрелки задается парой (p,q) , то есть, p единиц в x -направлении и q единиц в y -направлении. Числа p и q должны быть целыми, располагаться между -4 и $+4$ включительно и не должны иметь общего делителя. Полный набор наклонов векторов с положительными p и q , допустимых в ЛАТ_EX_ε, показан на рис.24, стр.144. Стрелка вектора помещается на конце, противоположном базовой точке. Если длина вектора равна нулю, то рисуется только стрелка, острие которой в данном случае оказывается расположенным как раз в базовой точке. С помощью этого приема к отрезкам прямых линий можно искусственно присоединять стрелку вектора, наиболее подходящую по наклону и тем самым расширить возможный диапазон наклона векторов. Например:

`\put(0,0){\line(6,1){150}}`
`\put(150,25){\vector(4,1){0}}`

Хрупкая.

`\vee` Производит символ бинарного оператора \vee , но только в математической моде. Тот же символ можно получить и командой `\lor`. Прочная.

`\verbchar text char` Выводит *text* шрифтом пишущей машинки точно в том виде, в каком он появляется во входном файле. Параметр *char* может быть любым одним видимым символом — кроме пробела, буквы и * — который не встречается в аргументе *text*. Пробел между *char* и *text* или между *text* и *char* появится во входном результате. Например, `\verb{"\it text \/"}` дает `{\it text \/}`. В аргументе *text* не должно быть символов новой строки.

`\verb*char text char` Похожа на команду `\verb`, за тем исключением, что пробелы в аргументе *text* в выходном результате появляются в виде `_`. Например, `\verb*{"\it text \/"}` дает `{\it_text_\/}`.

`verbatim` Командные скобки для получения текста шрифтом пишущей машинки точно в том виде, в котором он появляется во входном файле. Внутри командных скобок специальные символы и команды ЛАТ_EX_ε не имеют своего обычного значения и выводятся в том виде, в котором вводятся. Единственным исключением является команда `\end{verbatim}`. (Заметим, что между

`\end` и `{verbatim}` не должно быть пробела.) Не могут встретиться в аргументе любой другой команды, хотя и могут встретиться внутри других командных скобок.

`verbatim*` Командные скобки, аналогичные `verbatim`, за тем исключением, что пробелы во входном файле выводятся в виде символов ‘`␣`’.

`verse` Командные скобки для набора стихов. Левые и правые поля отступают от полей страницы на одинаковое расстояние. Строки внутри строф разделяются командами `\\`, а строфы разделяются одной или несколькими пустыми строками.

`\vert` Производит ординарный символ `|`, но только в математической моде; действует точно так же, как команда `|`. Однако после команд `\left` или `\right` команда производит ограничитель, размер которого зависит от размера того, что он ограничивает — см. рис.5 на стр.21, рис.6 на стр.22, а также следующий пример:

$$\| | | u | | = \sup_{x \neq 0} \left\{ \frac{\|u(x)\|}{\|x\|} \right\}$$

Прочная.

`\Vert` Производит ординарный символ `||`, но только в математической моде. Тот же символ можно получить и командой `\|`. Однако после команд `\left` или `\right` команда производит ограничитель, размер которого зависит от размера того, что он ограничивает — см. рис.5 на стр.21, рис.6 на стр.22, а также следующий пример:

$$u \cdot v = \left\| \frac{u+v}{2} \right\|^2 + \left\| \frac{u-v}{2} \right\|^2$$

$$u \cdot v = \left\| \frac{u+v}{2} \right\|^2 + \left\| \frac{u-v}{2} \right\|^2$$

Прочная.

`\vfill` Просто сокращение для `\par\vspace{\fill}`. Хрупкая.

`\vline` Может встретиться только внутри командных скобок `array` и `tabular`, где производит вертикальную прямую, высота которой равна высоте ряда, в котором она встречается. Прочная.

`volume` (Имя поля ВивTeX’a.) Номер тома журнала, книги, трудов конференций.

`\vspace{len}` Добавляет в выходной документ вертикальный пробел величиной `len`, если внутри ее “области действия” не встретился переход на новую страницу. Если команда `\vspace` встречается в середине абзаца, то перед вставкой вертикального пробела оканчивается текущая строка. Хрупкая.

`\vspace*{len}` Такая же, как команда `\vspace`, но вставляет вертикальный пробел, даже если внутри ее “области действия” встретился переход на новую страницу. Хрупкая.

W

`\wedge` Производит символ бинарного оператора \wedge , но только в математической моде. Тот же символ производит и команда `\land` (логическое и). Прочная.

`\widehat` В математической моде производит акцент переменного размера. Его самая широкая форма получается в `$$\widehat{xyz}$$`, что дает \widehat{xyz} . Используется для записи углов:

`$$\hat{x}, \widehat{X}, \widehat{AM!B}$$` $\hat{x}, \widehat{X}, \widehat{AMB}$

Прочная.

`\widetilde` В математической моде производит акцент переменного размера. Самая большая форма получается в `$$\widetilde{xyz}$$`, что дает \widetilde{xyz} . См. `\tilde`:

`$$\tilde{x}$$, $$\widetilde{X}$$,`

`$$\widetilde{XY}$$, $$\widetilde{XYZ}$$ $\tilde{x}, \widetilde{X}, \widetilde{XY}, \widetilde{XYZ}$`

Прочная.

`\wp` Функция \wp Вейерштрасса. Только в математической моде:

`$$\wp(z)={1\over z^2}+\sum_{\omega\in\Omega^*}\biggl[{{1\over(z-\omega)^2}-{1\over\omega^2}}\biggr]$$`

$$\wp(z) = \frac{1}{z^2} + \sum_{\omega \in \Omega^*} \left[\frac{1}{(z - \omega)^2} - \frac{1}{\omega^2} \right]$$

Здесь для увеличения размера квадратных скобок использовались команды plain TeXa `\biggl` и `\biggr`. Прочная.

`\wr` Бинарная операция \wr . Только в математической моде. Применяется для записи “косого произведения” или чтобы показать, что вертикальная стрелка обозначает изоморфизм в диаграмме:

`$$\matrix{A&\buildrel u\over{\hbox to 12mm{\rightarrowfill}}&B\cr\wr\bigg\downarrow\bigg\downarrow\wr\cr}`

`A'\buildrel u'\over{\hbox to 12mm{\rightarrowfill}}&B'\cr\}$`

$$\begin{array}{ccc} A & \xrightarrow{u} & B \\ \wr \downarrow & & \downarrow \wr \\ A' & \xrightarrow{u'} & B' \end{array}$$

Эта диаграмма является примером недоработанной кодировки: плохо распределены вертикальные пробелы, не центрируются вертикальные стрелки. Конечно же, без особых усилий ее можно сильно улучшить. Прочная.

X

`\xi` В математической моде производит ординарный символ ξ . Прочная.

`\Xi` В математической моде производит ординарный символ Ξ . Прочная.

Y

`year` (Имя поля ВивТ_EX'a.) Год, связанный с работой, на которую делается ссылка. Обычно он записывается как-то так: 1976, но стандартный стиль ВивТ_EX'a может обработать и текст типа "Около 1600", четыре последние непунктуационные символы которого должны быть числами.

Z

`\zeta` Производит ординарный символ ζ , но только в математической моде. Например, известная функция Римана

$$\zeta(s) = \sum_{n=1}^{\infty} \frac{1}{n^s}$$

и формула Эйлера

$$\zeta(s) = \frac{1}{\prod_{i=1}^{\infty} \left(1 - \frac{1}{p_i^s}\right)},$$

где $(p_i)_{i \geq 1}$ обозначает последовательность простых чисел. Приведем кодировку второй формулы:

```
$$\zeta(s)={1\over\displaystyle\prod_{i=1}^{\infty}
\Bigl(1-{1\over p_i^s}\Bigr)}\raise 2pt\hbox{,}$$
```

Обратите внимание на команду `\displaystyle` в знаменателе и на конструкцию plain Т_EXа `\raise2pt\hbox{,}`, из-за которой запятая находится на правильном уровне относительно дробной черты. Прочная.

Список литературы

- [1] Дональд Кнут, *Все про Т_EX*, перевод с англ., РДТЕХ, 1993.
- [2] Leslie Lamport, *L_AT_EX A Document Preparation System*, Addison Wesley, 1985.
- [3] Antony Diller, *L_AT_EX Line by Line*, John Wiley & Sons, 1993.
- [4] Бердников А.С., Туртия С.Б., *Т_EX и Графика*, Политехника, Санкт-Петербург, 1995.
- [5] Лисина М.В., *Plain Т_EX Основные понятия и каталог команд*, Препринт ИФВЭ 95-58.