

# Руководство для пользователя $\mathcal{A}\mathcal{M}\mathcal{S}$ - $\mathcal{T}\mathcal{E}\mathcal{X}$

С.В. Клименко, М.В. Лисина, Н.М. Фомина

Июнь 1999

## СОДЕРЖАНИЕ

1. Введение	4
1.1. Файлы, входящие в пакет $\mathcal{A}\mathcal{M}\mathcal{S}$ - $\mathcal{T}\mathcal{E}\mathcal{X}$ версии 2.1	4
1.2. Об этом руководстве	5
2. Подготовка текста в $\mathcal{T}\mathcal{E}\mathcal{X}$ 'е	6
2.1. Символы	6
2.2. Команды	6
2.3. Кавычки	7
2.4. Тире и многоточия	7
2.5. Горизонтальные пробелы	7
2.6. Вертикальные пробелы	8
2.7. Размеры	9
2.8. Группирование	9
2.9. Комментарии	9
2.10. Команды переключения шрифтов в тексте	10
2.11. Символы с акцентами и некоторые особые символы	12
2.12. Отсутствующие клавиши	13
2.13. Горизонтальные и вертикальные прямые	14
2.14. Формирование строк и абзацев; переносы	14
2.15. Разрыв страницы	16
2.16. Списки	16
2.17. Цитаты	18
2.18. Таблицы	19
2.19. Вставки с подписями	19
2.20. Сноски	21
2.21. Заголовки	22
2.22. Библиографические ссылки	23
2.23. Теоремы и доказательства	23
2.24. Присоединение дополнительных файлов	25
2.25. Текст в рамке и другие украшения	25
3. Шрифты, используемые в $\mathcal{A}\mathcal{M}\mathcal{S}$ - $\mathcal{T}\mathcal{E}\mathcal{X}$ 'е	27
3.1. Описание коллекции AMSFonts	27
3.2. Загрузка шрифтов	28
4. Набор математики	30
4.1. Основные принципы	30
4.2. Математические символы	33
4.3. Верхние и нижние индексы	39
4.4. Акценты в математике	42
4.5. Черта, стрелка или скобка над или под формулой	43

4.6.	Дроби и биномиальные коэффициенты	44
4.7.	Большие операторы	47
4.8.	Элементарные функции типа $\log$	49
4.9.	Корни	51
4.10.	Ограничители	52
4.11.	Составные символы	57
4.12.	Текст в формулах	59
4.13.	Коррекция математических формул с помощью дополнительных пробелов	62
4.14.	Матрицы	64
4.15.	Определения перечислением случаев	69
4.16.	Коммутативные диаграммы	70
4.17.	Формулы в рамках	72
4.18.	Многоточия	73
4.19.	Нумерация выключных формул	74
4.20.	Выравнивание выключных формул	76
4.21.	Многострочные формулы	81
4.22.	Шрифты в математике	85
5.	Имена дополнительных символов	90
5.1.	Специальные символы и жирные ажурные буквы	90
5.2.	Команда <code>\newsymbol</code>	90
5.3.	Таблица символов	91
6.	Средства форматирования	95
6.1.	Структура входного файла	95
6.2.	Область <code>\topmatter</code>	96
6.3.	Форматирование книги	99
6.4.	Номера страниц	101
6.5.	Размер страницы	101
6.6.	Колонтитулы	102
6.7.	Библиографии	103
7.	Определение новых команд	105
8.	Исправление ошибок	108
9.1.	Сообщения об ошибках	108
9.2.	Переополнение или недогрузка боксов	109
9.3.	Автоматическая проверка синтаксиса	110
10.	Сравнение с Plain $\TeX$ 'ом	112
11.	Выражение признательности за использование $\mathcal{A}\mathcal{M}\mathcal{S}\text{-}\mathcal{T}\mathcal{E}\mathcal{X}$	114
Приложение А. Пример получения библиографии		115
Приложение Б. Макеты шрифтов коллекции AMSFonts		119
Предметный указатель		125
Библиография		141

## 1. ВВЕДЕНИЕ

$\mathcal{A}\mathcal{M}\mathcal{S}$ - $\mathcal{T}\mathcal{E}\mathcal{X}$  — это макронастройка  $\mathcal{T}\mathcal{E}\mathcal{X}$ 'а, созданная, чтобы упростить набор математических формул и отформатировать результаты в соответствии со спецификациями заданного стиля. Этот раздел описывает все возможности, относящиеся к  $\mathcal{A}\mathcal{M}\mathcal{S}$ - $\mathcal{T}\mathcal{E}\mathcal{X}$  версии 2.1. Материал взят в основном из известной книги Спивака (Spivak, Michael D.) *Восхитительный  $\mathcal{T}\mathcal{E}\mathcal{X}$* , а также из *User's Guide to  $\mathcal{A}\mathcal{M}\mathcal{S}$ - $\mathcal{T}\mathcal{E}\mathcal{X}$  Version 2.1*.

В большей части это руководство предполагает, что вы будете использовать стиль “препринт” — набор макрокоманд, которые задают специфику форматирования документа: вид заголовков, стиль нумерации страниц и т.д.

Мы также считаем, что читатель знаком с основными понятиями и командами Plain  $\mathcal{T}\mathcal{E}\mathcal{X}$ 'а. По этой причине здесь не приводится полный список команд и возможностей Plain  $\mathcal{T}\mathcal{E}\mathcal{X}$ 'а, а делается упор на специфически  $\mathcal{A}\mathcal{M}\mathcal{S}$ - $\mathcal{T}\mathcal{E}\mathcal{X}$ 'овские команды. Сравнение Plain  $\mathcal{T}\mathcal{E}\mathcal{X}$ 'а и  $\mathcal{A}\mathcal{M}\mathcal{S}$ - $\mathcal{T}\mathcal{E}\mathcal{X}$ 'а проводится в специальном подразделе.

Хотя авторские права на  $\mathcal{A}\mathcal{M}\mathcal{S}$ - $\mathcal{T}\mathcal{E}\mathcal{X}$  принадлежат American Mathematical Society, American Mathematical Society не только не ограничивает его использования, но, более того, поощряет его применение для подготовки руководств, предназначенных для публикации как в книгах и журналах, издаваемых этим обществом, так и в других математических изданиях. В знак признания авторских прав Общество требует, чтобы издаваемые документы, подготовленные с помощью  $\mathcal{A}\mathcal{M}\mathcal{S}$ - $\mathcal{T}\mathcal{E}\mathcal{X}$ , включали указание на его использование. Предлагаемые формы для такого указания даны в разделе **Выражение признательности за использование  $\mathcal{A}\mathcal{M}\mathcal{S}$ - $\mathcal{T}\mathcal{E}\mathcal{X}$** .

1.1. Файлы, входящие в пакет  $\mathcal{A}\mathcal{M}\mathcal{S}$ - $\mathcal{T}\mathcal{E}\mathcal{X}$  версии 2.1

В пакет  $\mathcal{A}\mathcal{M}\mathcal{S}$ - $\mathcal{T}\mathcal{E}\mathcal{X}$  версии 2.1, распространяемый American Mathematical Society, входят следующие файлы:

AMSTEX.TEX	макрокоманды $\mathcal{A}\mathcal{M}\mathcal{S}$ - $\mathcal{T}\mathcal{E}\mathcal{X}$ версии 2.1
AMSSYM.TEX	макрокоманды, определяющие символы в шрифтах <code>msam</code> и <code>msbm</code>
AMSPPT.STY	стиль “препринт” для $\mathcal{A}\mathcal{M}\mathcal{S}$ - $\mathcal{T}\mathcal{E}\mathcal{X}$ версии 2.1
AMSPPT.DOC	техническая документация для AMSPPT.STY
AMSGUIDE.TEX	исходный файл для User's Guide
AMSPPT1.TEX	файл обратной совместимости для использования с файлами, подготовленными на $\mathcal{A}\mathcal{M}\mathcal{S}$ - $\mathcal{T}\mathcal{E}\mathcal{X}$ версий, более ранних, чем 2.0
JOYERR.TEX	файл изменений к <i>Восхитительный <math>\mathcal{T}\mathcal{E}\mathcal{X}</math></i> (первое издание)
*.TFM	TFM файлы для AMSFonts версии 2.1
AMSTEX.INI	используется в создании форматных файлов
AMSTEX.BAT	только для инсталляции в DOS

Файл AMSPPT.DOC представляет собой ASCII файл, и он не предназначен для обработки  $\mathcal{T}\mathcal{E}\mathcal{X}$ 'ом. Это файл, в котором содержатся описания макрокоманд, расположенные в том же порядке, в котором эти макрокоманды располагаются в файле макрокоманд. В нем подробно описаны назначение и механизм работы этих макрокоманд. Имеется также отдельный файл AMSTEX.DOC), содержащий документацию к файлу AMSTEX.TEX, который можно получить по специальному запросу.

И наконец, остальные файлы используются для инсталляции системы с дискет.

## 1.2. Об этом руководстве

Это руководство является компиляцией из следующих основных источников:

- (1) Д. Е. Кнут *Все про  $\mathcal{T}\mathcal{E}\mathcal{X}$* ;
- (2) М. Д. Спивак *Восхитительный  $\mathcal{T}\mathcal{E}\mathcal{X}$*
- (3) User's Guide to  $\mathcal{A}\mathcal{M}\mathcal{S}$ - $\mathcal{T}\mathcal{E}\mathcal{X}$  Version 2.1;
- (4) User's Guide to AMSFonts Version 2.2;
- (5) Файл документации AMSPPT.DOC.

Оригинал-макет подготовлен с помощью  $\mathcal{A}\mathcal{M}\mathcal{S}$ - $\mathcal{T}\mathcal{E}\mathcal{X}$  версии 2.1 в стиле “препринт”. При подготовке раздела использовались макрокоманды из *User’s Guide to  $\mathcal{A}\mathcal{M}\mathcal{S}$ - $\mathcal{T}\mathcal{E}\mathcal{X}$  Version 2.1*, которые: переустанавливают шрифты и размеры, переопределяют макрокоманды создания заголовков, чтобы полученный результат был более похож на документацию, а также определяют некоторые макрокоманды, чтобы упростить представление конкретной информации. Однако, вообще говоря, этот документ и файл, из которого он был получен, иллюстрирует внешний вид и входные данные для препринта с бегущими верхними заголовками. Для распечатки результата обработки  $\mathcal{T}\mathcal{E}\mathcal{X}$ ’ом этого руководства требуется коллекция AMSFonts версии 2.0 или выше.

## 2. ПОДГОТОВКА ТЕКСТА В $\text{T}\text{E}\text{X}$ 'Е

Текст (имеется ввиду обычный текст, не содержащий математических формул) готовится в  $\mathcal{A}\mathcal{M}\mathcal{S}$ - $\text{T}\text{E}\text{X}$ 'е точно также, как и в Plain  $\text{T}\text{E}\text{X}$ 'е. Тем, кто этого не умеет, рекомендуем прочесть руководство Д. Кнута *Все про  $\text{T}\text{E}\text{X}$* . Но для удобства изложения и для особо ленивых читателей мы повторим здесь основные понятия.

Входной информацией для  $\text{T}\text{E}\text{X}$ 'а является файл, содержащий рукопись автора, специальные символы, команды и ключевые слова. Этот файл готовится с помощью любого текстового редактора и его имя должно иметь расширение `.tex`. На выходе  $\text{T}\text{E}\text{X}$  автоматически создает два файла: результат работы — файл с расширением `.dvi` (device-independent file), содержащий сформированный выходной документ в виде, не зависящем от типа устройства вывода, и протокол работы — обычно с расширением `.log`.

При подготовке входного файла необходимо знать следующее.

### 2.1. Символы

В  $\text{T}\text{E}\text{X}$ 'е можно использовать все стандартные символы ASCII, а также прописные и строчные буквы русского алфавита, однако 10 выделенных символов можно использовать только специальным образом. Эти *специальные символы* являются служебными и не могут быть напечатаны в тексте обычным образом. Следующая таблица показывает назначение специальных символов и способ их ввода для изображения в выходном документе.

Символ	Назначение	Ввод
<code>\</code>	Сигнальный символ	<code>\backslash</code>
<code>{</code>	Признак начала группы	<code>\{</code>
<code>}</code>	Признак конца группы	<code>\}</code>
<code>\$</code>	Переключатель в математическую моду	<code>\\$</code>
<code>&amp;</code>	Табулятор	<code>\&amp;</code>
<code>#</code>	Признак параметра в макроопределениях	<code>\#</code>
<code>^</code>	Верхний индекс	<code>\^</code>
<code>_</code>	Нижний индекс	<code>\_</code>
<code>%</code>	Символ комментария	<code>\%</code>
<code>~</code>	Неразрываемый пробел	<code>\~</code>
<code>@</code>		<code>\@</code>

### 2.2. Команды

Формат  $\mathcal{A}\mathcal{M}\mathcal{S}$ - $\text{T}\text{E}\text{X}$  включает в себя все *примитивы* (т. е. команды низшего уровня)  $\text{T}\text{E}\text{X}$ 'а, почти все макрокоманды Plain  $\text{T}\text{E}\text{X}$ 'а, а также свои собственные команды. Все команды начинаются с символа `\` (бэкслэш). Имена команд (их часто называют *управляющими последовательностями*) состоят либо только из букв, и тогда такая управляющая последовательность называется управляющим или командным словом, либо из одной не буквы, и тогда она называется управляющим или командным символом.  $\text{T}\text{E}\text{X}$  игнорирует пробелы после управляющих слов и не игнорирует из после управляющих символов. Признаком конца управляющего слова считается первая не буква. Прописные и строчные буквы в именах команд различаются (`\bigl` и `\Bigl` — это разные команды). Имеются также слова, которые в определенном контексте имеют особый смысл. Такие слова называются *ключевыми*. Например, в конструкции `\hbox to 5cm{ ... }` слово `to` — ключевое.

### 2.3. Кавычки

Обычные левые `'` и правые `'` кавычки, имеющиеся на клавиатуре дают ‘одинарные’ кавычки. Чтобы получить принятую в типографском тексте “двойную кавычку”, надо соответствующую одинарную кавычку ввести дважды. Если на клавиатуре нет символов одинарных открывающей и закрывающей кавычек, вместо них можно использовать, соответственно, команды `\lq` и `\rq`.

### 2.4. Тире и многоточия

Не путайте различного вида тире, дефис и знак минус. Обычно существуют по меньшей мере четыре разных символа:

дефис или знак переноса: -  
 черточка или en-тире (тире длиной примерно в букву n): –  
 тире или em-тире (тире длиной примерно в букву m): —  
 знак минуса: −

Дефисы используются в составных словах типа “мальчик-с-пальчик” и “N-кратный”. En-тире применяется для указания диапазонов чисел, типа “страницы 13–34”, “упражнения 1.2.6–1.2.8”. Em-тире служит знаком пунктуации в предложениях — это то, что мы обычно называем простым тире. Знак минуса используется в формулах. Добросовестный пользователь будет внимательно отличать эти четыре применения. Вот как это делается:

для дефиса надо печатать дефис (-)  
 для en-тире — печатать два дефиса (--)  
 для em-тире — печатать три дефиса (---)  
 для знака минуса — печатать дефис в математической моде (\$-\$).

Для того, чтобы задать в тексте многоточие, не следует вводить подряд три точки (что приведет к такому некрасивому результату: ...). Вместо этого следует использовать команду `\dots`, которая в тексте дает симпатичное многоточие: ...

## 2.5. Горизонтальные пробелы

При наборе текста несколько пробелов подряд считаются за один пробел. Конец входной строки эквивалентен пробелу, а пустая строка обозначает конец абзаца. Пробелы после командных слов игнорируются, а после командных символов — нет (напомним, что командным словом в  $\TeX$ 'е называется управляющая последовательность, состоящая только из символа бэкслэш и следующих за ним букв, а командным символом — командная последовательность, состоящая из символа бэкслэш и одного не буквенного символа, например `\$`. Если же после командного слова вам все-таки нужен пробел, то следует сказать об этом явно, используя команду для *принудительного пробела* — `\_`.

Количество пробелов после знаков препинания также несущественно, однако как минимум один пробел следует поставить. Поскольку  $\TeX$  автоматически увеличивает пробелы после знаков препинания, то для представления знаков препинания, пробел после которых не следует увеличивать,  $\mathcal{A}\mathcal{M}\mathcal{S}\text{-}\TeX$  имеет управляющие символы `\.`, `\,`, `\!` и так далее. Пробелы после этого управляющего символа  $\TeX$ , естественно, не игнорирует. Следует отметить, что в математических формулах управляющие последовательности `\.` и `\!` используются для совсем других целей — там они обозначают крошечный (1/10 от тонкого пробела), соответственно, положительный и отрицательный пробел.

Иногда требуется запретить  $\TeX$ 'у разрывать строку на пробеле, например, чтобы инициалы не оказались на одной строке, а фамилия — на другой. Пробел, на котором нельзя разрывать строку, задается символом `\~`.

Для задания *горизонтальных пробелов* в тексте и в формулах  $\mathcal{A}\mathcal{M}\mathcal{S}\text{-}\TeX$  имеет целый ряд команд:

<code>\quad</code>	“квадрат”, равен 1em или трем обычным пробелам
<code>\qquad</code>	двойной “квадрат” равен 2em или шести пробелам
<code>\,</code> ( <code>\thinspace</code> )	тонкий пробел, равен 1/6 квадрата
<code>\!</code> ( <code>\negthinspace</code> )	отрицательный тонкий пробел, $-1/6$ квадрата
<code>\;</code> ( <code>\thickspace</code> )	широкий пробел, равен 5/18 квадрата
<code>\negthickspace</code>	отрицательный пробел, равен $-5/18$ квадрата
<code>\medspace</code>	средний пробел
<code>\negmedspace</code>	отрицательный средний пробел

Кроме этого, в тексте можно явно задать величину пробела командой `\hskip`: например, `\hskip1cm` означает горизонтальный пробел, величиной в 1 сантиметр. Величина такого пробела может быть задана в любых допустимых в  $\TeX$ 'е единицах.

## 2.6. Вертикальные пробелы

Для задания дополнительных *вертикальных пробелов* между абзацами  $\TeX$  имеет команды `\smallskip`

(маленький пробел величиной в 3pt с допуском плюс-минус 1pt), `\medskip` (средний пробел) и `\bigskip` (большой пробел величиной в 12pt с допуском плюс-минус 4pt):

`\smallskip`        `\medskip`        `\bigskip`    

Можно также явно задать величину вертикального пробела командой `\vskip`: например, `\vskip 1cm` означает вертикальный пробел, величиной в 1 сантиметр. Величина такого пробела может быть задана в любых допустимых в  $\TeX$ 'е единицах.

В стиле препринт имеются и дополнительные команды `\smallpagebreak`, `\medpagebreak` и `\bigpagebreak` для получения таких вертикальных пробелов:

`\smallpagebreak`        `\medpagebreak`        `\bigpagebreak`    

Их следует использовать только между абзацами (либо в самом начале или в самом конце абзаца). Эти команды подсказывают  $\TeX$ 'у, что разрыв страницы в этом месте не только возможен, но и желателен. Если в этом месте происходит разрыв страницы, то пробел исчезает. Если встречаются несколько таких пробелов, то они не суммируются, а выбирается наибольший.

## 2.7. Размеры

Для своих целей (например, для указания величины пробела)  $\TeX$  использует *размеры*, выраженные в различных единицах измерения, как традиционных полиграфических, так и метрических:

<code>pt</code>	point	пункт (расстояние между базовыми линиями в этом руководстве равно 12 pt)
<code>pc</code>	pica	пика (1 pc = 12 pt)
<code>in</code>	inch	дюйм (1 in = 72.27 pt)
<code>bp</code>	big point	большой пункт (72 bp = 1 in)
<code>cm</code>	centimeter	сантиметр (2.54 cm = 1 in)
<code>mm</code>	millimeter	миллиметр (10 mm = 1 cm)
<code>dd</code>	didot point	дидот-пункт (1157 dd = 1238 pt)
<code>cc</code>	cicero	цицero (1 cc = 12 dd)
<code>sp</code>	scaled point	суперпункт (65536 sp = 1 pt)

Кроме указанных выше, в  $\TeX$ 'е используются две единицы измерения, которые зависят от текущего шрифта: `em` — немного меньше, чем ширина заглавной буквы М текущего шрифта и `ex` — приблизительно высота строчной буквы x текущего шрифта.

Размеры всегда выражаются числом (положительным или отрицательным, целым или с десятичной точкой) и следующей за ним единицей измерения. Пробел между числом и единицей измерения, а также пробел между числом и знаком + или - не имеет значения. Более того, сам знак + можно опустить. Вот типичные размеры:

`-.5 in`  
`12pt`  
`+70 bp`  
`-0.2in`  
`0cm`

При указании нулевого размера нельзя набирать просто 0, а следует писать `0pt`, `0cm` или что-то в этом роде.

## 2.8. Группирование

Иногда необходимо часть рукописи рассматривать как одну единицу и указать, где эта часть начинается, а где кончается. Такая часть называется *группой* и для ее задания используются *символы группирования* `{` и `}`. Все указания, которые  $\TeX$  получил внутри группы, немедленно забываются, как только группа закончилась. Иными словами, команды внутри группы действуют *локально*.

## 2.9. Комментарии

Иногда автор хочет сделать некоторую часть своего входного файла невидимой для посторонних. Для этого ее можно “закомментировать”: специальный символ % делает “невидимой” часть текста, начиная от себя самого и вплоть до конца строки, на которой он находится. Иными словами, он ограничивает строку входного файла, не вводя при этом пробела, который Т<sub>Е</sub>X обычно вставляет, переходя к следующей строке.

Чтобы “закомментировать” несколько строк, надо поставить % в начале каждой строки. Кроме этого, в А<sub>М</sub>S-Т<sub>Е</sub>X’е есть и другой способ, более подходящий для большого объема текста — это конструкция

```
\comment
...
\endcomment
```

Команда `\endcomment` *обязательно* должна находиться на отдельной строке. Не следует вкладывать конструкции `\comment` друг в друга, т.е. использовать конструкцию

```
\comment
...
\comment
...
\endcomment
...
\endcomment
```

## 2.10. Команды переключения шрифтов в тексте

Когда Вы начинаете готовить входной файл для Т<sub>Е</sub>X’а, Вам надо знать, какими полиграфически возможностями Вы можете располагать. В А<sub>М</sub>S-Т<sub>Е</sub>X’е можно получать символы из коллекции шрифтов AMSFonts, которые обеспечивают набор самых разнообразных документов. В каждом из шрифтов Т<sub>Е</sub>X может иметь до 256 символов. Символ шрифта можно задать по его номеру. Так, например, если Вы запрашиваете `\char’35` в шрифте `cmr10`, то получаете *Æ*. Текстовые шрифты включают *лигатуры* и *акценты*.

Каждый шрифт в рукописи Т<sub>Е</sub>X’а связан с командной последовательностью, например, 10-пунктовый шрифт в этом руководстве вызван командой `\tenrm`, а соответствующий 8-пунктовый шрифт называется `\eightrm`. Наклонные шрифты, которые соответствуют `\tenrm` и `\eightrm`, вызываются `\tensl` и `\eighths`. Эти командные последовательности не являются реальными именами шрифтов; предполагается, что когда в рукопись вводятся новые шрифты, пользователи Т<sub>Е</sub>X’а определяют дополнительные имена. Такие командные последовательности используются для изменения типа печати. Для этой цели служит команда `\font`. Так, конструкция `\font\myfont= cmr17` делает `\myfont` командой для включения очень крупного романского шрифта (размером в 17 пунктов).

Для изменения шрифтов А<sub>М</sub>S-Т<sub>Е</sub>X уже имеет следующие команды:

```
\rm — обычный прямой шрифт (roman)
\sl — наклонный (slanted)
\it — курсив (italic)
\tt — шрифт пишущей машинки (typewriter)
\bf — жирный шрифт (bold)
\smc — КАПИТЕЛЬ (SMALL CAPITAL)
```

В начале работы получается прямой шрифт (`\rm`), если только Вы не указали другое. Включенный шрифт является текущим до тех пор, пока либо не встретится другая команда переключения шрифта, либо не закончится группа, т.е. команды шрифтов являются *локальными*.

Внимательный читатель здесь должен быть смущен, потому, что сначала мы сказали, что `\tenrm` — это командная последовательность, которая включает романский шрифт, а позднее — что это делает `\rm`. Правда в том, что `\rm` означает “включи романский шрифт в текущем размере”, а `\tenrm` — “включи романский шрифт в 10-пунктовом размере”. В А<sub>М</sub>S-Т<sub>Е</sub>X’е имеются команды `\tensl`, `\tenit`, `\tentt`, `\tenbf` и `\tensmc`, а также команды для шрифтов в 8 пунктов `\eightrm`,

`\heightsl`, `\heightit`, `\heightbf` и `\heightsmc`. Кроме того, существует командная последовательность `\tenpoint`, которая указывает, что `\rm` означает `\tenrm`, `\sl` означает `\tensl`, и т. д., пока `\heightpoint` не изменит определения, так что `\rm` будет означать `\eightrm`, и т. д. Этот механизм постоянно переопределяет аббревиатуры `\rm` и `\sl` и им подобные в зависимости от места и освобождает печатающего от необходимости помнить, какой размер или стиль печати используется в текущий момент.

Заметим, что два шрифта имеют “наклон”: *наклонный шрифт* — это, по существу, такой же шрифт, как *прямой*, но его буквы слегка наклонены в сторону, тогда как буквы курсива пишутся по-другому. Если после курсивного слова следует прямое, для компенсации уменьшения пробела между ними необходимо перед включением прямого шрифта поместить команду `\/` — так называемую *курсивную поправку*.

Имеется возможность использовать шрифт нескольких различных размеров, увеличивая или сжимая изображение символов. Каждый шрифт имеет так называемый *проектный размер*, обычно присущий ему по умолчанию; например, проектный размер шрифта `cmr9` — 9 пунктов. Существует также диапазон размеров, в которых Вы можете использовать определенный шрифт, уменьшая или увеличивая его размеры.

Оказалось удобным иметь шрифты с коэффициентами увеличения 1.2 и 1.44 (что есть  $1.2 \times 1.2$ ), а возможно также с увеличением 1.728 ( $= 1.2 \times 1.2 \times 1.2$ ) и даже выше. Тогда можно увеличивать размер напечатанного в целом в 1.2 или 1.44 раза и все еще оставаться внутри набора доступных шрифтов. Plain  $\text{T}\text{E}\text{X}$  содержит аббревиатуры `\magstep0` для масштаба 1000, `\magstep1` для масштаба 1200, `\magstep2` для 1440 и так далее до `\magstep5`. Например, чтобы загрузить шрифт `cmr10` в  $1.2 \times 1.2$  его нормального размера, вы говорите `\font\bigtenrm=cmr10 scaled\magstep2`.

Это шрифт `cmr10` в нормальном размере (`\magstep0`).

Это шрифт `cmr10`, увеличенный в 1.2 (`\magstep1`).

Это шрифт `cmr10`, увеличенный в 1.44 (`\magstep2`).

Существует также `\magstephalf`, которое увеличивает в  $\sqrt{1.2}$  раза, т.е., посередине между шагами 0 и 1.

При увеличении выходного документа увеличиваются все заданные размеры. Если требуется, чтобы какой-нибудь размер не менялся, его надо задавать в неизменяемых единицах с использованием ключевого слова `true`, например, `\hsize=115 true mm`.

## 2.11. Символы с акцентами и некоторые особые символы

Мы уже говорили, как вводить обычные символы, специальные символы и знаки препинания. Теперь поговорим о более сложных символах. Прежде всего, это символы с акцентами. Каждый акцент над буквой выражается управляющей последовательностью с аргументом. В приведенной ниже таблице в качестве аргумента используется буква `o`. В пределах шрифта эти акценты разработаны так, что они появляются на высоте, правильной для этой буквы, но их также можно использовать над любой другой буквой;  $\text{T}\text{E}\text{X}$ , когда надо, поднимет знак акцента.

<i>Вводим</i>	<i>получаем</i>	
<code>\'o</code>	ò	(гравис)
<code>\'o</code>	ó	(акут)
<code>\^o</code>	ô	(циркумфлекс или “шляпка”)
<code>\"o</code>	ö	(умлаут)
<code>\~o</code>	õ	(тильда)
<code>\V o</code>	ō	(черта “над”)
<code>\b o</code>	o̲	(черта “под”)
<code>\D o</code>	ȯ	(точка “над”)

<code>\d o</code>	o	(точка “под”)
<code>\u o</code>	ö	(акцент краткости)
<code>\v o</code>	ø	(гачек или “галочка”)
<code>\H o</code>	ő	(венгерский умляут)
<code>\t oo</code>	öo	(лига)

Следует иметь ввиду, что для помещения акцентов над буквами *i* и *j*, следует использовать их бесточечные версии *i* и *j*, которые получаются, соответственно, командами `\i` и `\j`. Например, `\Vi` даст  $\bar{i}$ , тогда как `\V i` даст  $\bar{i}$ .

Есть также несколько специальных букв:

*Вводим получаем*

<code>\oe, \OE</code>	œ, Œ	(французская лигатура OE)
<code>\ae, \AE</code>	æ, Æ	(латинская и скандинавская лигатура AE)
<code>\aa, \AA</code>	å, Å	(скандинавская A-с-кружочком)
<code>\o, \O</code>	ø, Ø	(скандинавское O-перечеркнутое)
<code>\l, \L</code>	ł, Ł	(польская перечеркнутая L)
<code>\ss</code>	ß	(немецкая “эс-цет” или острое S)

TeX считает некоторые комбинации символов лигатурами:

<code>ff</code> дает ff	<code>ffi</code> дает ffi	<code>‘ ‘</code> дает “	<code>! ‘</code> дает ¡
<code>fi</code> дает fi	<code>ffl</code> дает ffl	<code>’ ’</code> дает ”	<code>? ‘</code> дает ¿
<code>fl</code> дает fl	<code>--</code> дает –	<code>---</code> дает —	

Следующие символы выглядят одинаково независимо от того, используете ли вы шрифты `\rm`, `\sl`, `\bf`, `\it` или `\tt`:

*Вводим получаем*

<code>\dag</code>	†	(кинжал или обелиск)
<code>\ddag</code>	‡	(двойной кинжал или двойной крестик)
<code>\S</code>	§	(знак номера параграфа)
<code>\P</code>	¶	(знак абзаца)

В  $\mathcal{A}\mathcal{M}\mathcal{S}$ -TeX’e можно получать числа в старинном написании. Для этого служит команда `\oldnos`: если набрать `\oldnos{5,283.06}`, то получится 5,283.06. Можно также использовать `\oldnos` в математических формулах (в математической моде); числа, стоящие в индексах, получатся правильного размера, и хотя в математической моде после запятой обычно вставляется тонкий пробел, внутри `\oldnos` никаких пробелов после запятой не будет.

Для защиты своих авторских прав можно использовать и `\copyright` — знак ©.

## 2.12. Отсутствующие клавиши

На вашей клавиатуре могут отсутствовать некоторые клавиши о которых здесь упоминалось. Нажатие таких клавиш можно заменить вводом соответствующей управляющей последовательности:

<i>Команда</i>	<i>Клавиша</i>
<code>\lbrack</code>	[
<code>\lq</code>	‘
<code>\rbrack</code>	]
<code>\rq</code>	’
<code>\sp</code>	^
<code>\sb</code>	~
<code>\tie</code>	~
<code>\vert</code>	

Естественно, вам потребуются команды для комбинация с этими клавишами:

<i>Команда</i>	<i>Для</i>
<code>\acuteaccent</code>	<code>\`</code>
<code>\graveaccent</code>	<code>\`</code>
<code>\hataccent</code>	<code>\^</code>
<code>\tildeaccent</code>	<code>\~</code>
<code>\underscore</code>	<code>\_</code>
<code>\Vert</code>	<code>\ </code>

Для получения фигурных скобок можно принять команды:

<code>\lbrace</code>	<code>\{</code>
<code>\rbrace</code>	<code>\}</code>

### 2.13. Горизонтальные и вертикальные прямые

$\TeX$  имеет некие особые объекты — это так называемые линейки. Линейку можно рассматривать как прямоугольник, покрашенный черной краской. Для задания линеек имеются команды `\hrule`, `\vrule` и `\hrulefill`.

Команда `\hrule height 2pt depth 1pt width 3cm` проводит горизонтальную черту высотой 2 pt, глубиной 1 pt и длиной 3 см. Когда  $\TeX$  встречает эту команду, он переходит в вертикальную моду, т.е. начинает новый абзац. Нельзя воспользоваться этой командой внутри абзаца, поскольку там Вы находитесь в горизонтальной моде. Параметры `height`, `depth` и `width` должны быть расположены именно в таком порядке. Часть этих параметров может отсутствовать. Если ширина `width` отсутствует,  $\TeX$  считает, что `width` равняется ширине самого большого бокса, помещенного в вертикальный бокс, в котором находится эта черта (например, черта идет от одного поля страницы до другого). Если `depth` отсутствует,  $\TeX$  присваивает ему значение 0 pt. Если отсутствует `height`, он ему дает значение 0.4 pt. Команда `\hrule` сама по себе и без параметров в тексте заканчивает текущий абзац и рисует черту толщиной 0.4 pt, идущую от одного поля страницы до другого. Помните также, что  $\TeX$  не добавляет вертикальных пробелов ни до, ни после `\hrule`.

Чтобы провести черту в горизонтальной моде, используйте `\vrule`. Она работает в горизонтальной моде, следовательно, при использовании этой команды надо находиться внутри абзаца, в `\hbox` или внутри таблицы. Можно задавать размеры черты: `\vrule height 12pt depth 5pt width 1pt`. Если `height` отсутствует,  $\TeX$  рисует черту до потолка бокса, который ее содержит. Аналогично, если отсутствует `depth`, черта опускается до нижней границы включающего ее бокса. Отсутствие `width` — это особый случай: по умолчанию значение ширины равно 0.4 pt. Чтобы получить невидимую черту, надо задать ей ширину 0 pt. И последнее замечание: черта в вертикальном боксе не отделяется от боксов сверху и снизу никакими пробелами.

Команда `\hrulefill` служит для проведения горизонтальной черты, которая растягивается на всю ширину содержащего ее горизонтального бокса. *Она не занимает места*, следовательно, `\hbox{\hrulefill}` и `{\hrulefill}` ничего не делают.

### 2.14. Формирование строк и абзацев; переносы

Одна из главных обязанностей  $\TeX$ 'а — это взять длинную последовательность слов и разбить ее на строки подходящего размера, сформировав тем самым абзац. Как уже говорилось, признаком конца абзаца служит пустая строка. Для этой цели можно также использовать команду  $\TeX$ 'а `\par`.  $\TeX$  рассматривает абзац как единое целое; слова в конце абзаца могут даже повлиять на вид первой строки. В результате пробелы между словами насколько это возможно единообразны, и можно во много раз уменьшить количество переносов слов или формул, разорванных между строками.

$\TeX$  присоединяет к каждой строке некоторую числовую величину, так называемую *плохость* (badness), чтобы оценить эстетическое восприятие пробелов между словами, а также имеет параметр `\tolerance` (допуск). Задавая различные значения параметра `\tolerance`, можно получать различные результаты. Более высокий допуск означает, что допускаются более широкие пробелы.  $\TeX$  будет искать самый лучший способ напечатать каждый абзац в соответствии с принципом наименьшей плохости.

Если вы хотите заставить  $\TeX$  сделать разрыв между строками в некоторой точке в середине абзаца, надо просто поместить в этой точке команду `\linebreak`. Однако, это может привести к тому,

что на строке будут слишком широкие пробелы. Если вы хотите, чтобы  $\TeX$  прервал строку и чтобы следующая строка была без отступа, используйте команду `\newline`. Если набрать `\newline\newline`, то получится пустая строка. Предотвратить разрыв строки на некотором пробеле можно, либо используя вместо этого пробела “связку” `~`, либо командой `\nolinebreak`. Имеется также команда `\allowlinebreak`, которая разрешает разрыв строки там, где  $\TeX$  обычно этого не делает.

При необходимости при формировании строк делается перенос слов. Для этого  $\TeX$  использует свой внутренний словарь переносов и список исключений, который можно задавать командой `\hyphenation`. Например, команда

```
\hyphenation{man-u-script man-u-scripts ap-pen-dix}
```

сообщает о правилах переноса трех исключительных слов. Можно также явно подсказать  $\TeX$ ’у, что в некотором месте можно сделать перенос с помощью команды возможного переноса `\-`. Так, например, поскольку  $\TeX$  автоматически не переносит составные слова типа “физико-математический”, то их можно вводить так:

```
фи\~\-зи\~\-ко\~\-ма\~\-те\~\-ма\~\-ти\~\-чес\~\-кий
```

Если вас интересует мнение  $\TeX$ ’а по поводу возможности переноса каких-либо слов, вы можете спросить его об этом командой

```
\showhyphen{слово-1 слово-2 ... слово-n}
```

В результате такой команды  $\TeX$  выведет на терминал перечисленные в фигурных скобках слова со вставленными в них возможными переносами.

$\TeX$  не умеет переносить слова типа “ввод/вывод”. Для указания такой /, на котором  $\TeX$  может разорвать строку, имеется команда `\slash`:

```
ввод\slash вывод
```

$\mathcal{A}\mathcal{M}\mathcal{S}\text{-}\TeX$  игнорирует любые пробелы перед `\slash`.

Может возникнуть желание и запретить перенос. Это можно сделать несколькими способами. Во-первых, набрать `\text{слово}`, потому что команда `\text` делает свой аргумент неразрывным фрагментом или вместо `\text` использовать команду Plain  $\TeX$ ’а `\hbox`, которая также “запирает” фрагмент строки и делает его неразрывным: `\hbox{ ... }`. Можно также в конце слова поставить возможный перенос `\-` ( $\TeX$  никогда не добавит перенос в то слово, в котором уже есть дефис или возможный перенос) или же вставить в файл команду `\hyphenation{слово}`.

Иногда бывает нужно, чтобы каждой строке на входе соответствовала строка на выходе.  $\TeX$  предусматривает команду `\obeylines`, которая каждый конец строки рассматривает как `\par`. После того, как вы зададите `\obeylines`, вы будете получать по одной строке выхода на каждую строку входа, если только входная строка не оканчивается % или если она не настолько длинна, что должна разрываться.

Кроме того, можно получать абзацы со строками, сдвинутыми к левому краю. Это достигается командой `\flushpar`. Так, абзац, который вы читаете, был получен так: `\flushpar` Кроме того, ...

Напомним, что в начале абзаца  $\TeX$  делает *абзацный отступ*. Величина этого отступа определяется стилем и задается параметром `\parindent`. Подавить этот отступ можно командой `\noindent`, а включить его (если он не делается автоматически) — командой `\indent`.

Кроме того, можно получать абзацы со строками, сдвинутыми к левому краю. Это достигается командой `\flushpar`. Так, абзац, который вы читаете, был получен так: `\flushpar` Кроме того, ...

## 2.15. Разрыв страницы

Аналогично команде для разрыва строк `\linebreak`,  $\mathcal{A}\mathcal{M}\mathcal{S}\text{-}\TeX$  для разрыва страниц имеет команду `\pagebreak`. Если эту команду поместить между абзацами, то произойдет переход на следующую страницу, а оставшийся на предыдущей странице текст будет растянут до нижнего поля. При использовании внутри абзаца команда `\pagebreak` приводит к переносу текста на следующую страницу *по окончании той строки, на которой она находится*.

В обычной математической моде `\pagebreak` не допускается, но ее можно использовать в выключных формулах. Чтобы вызвать разрыв страницы после выключной формулы, следует поместить `\pagebreak` внутри этой формулы, поскольку, если поставить `\pagebreak` сразу после формулы, то на странице после нее появится дополнительное пустое пространство.

Аналогично для предотвращения разрыва страницы используется команда `\nopagebreak`. В частности, если нужно избежать разрыва страницы после выключной формулы, следует поставить `\nopagebreak` *внутри* нее. Вне выключной формулы эта команда не подействует. Можно также “запереть” фрагмент текста в `\vbox`, после чего этот фрагмент становится неразрывным: `\vbox{...}`.

По аналогии с `\newline`,  $\mathcal{A}\mathcal{M}\mathcal{S}$ - $\mathcal{T}\mathcal{E}\mathcal{X}$  имеет команду `\newpage`. Ее можно использовать только между абзацами. Команда `\newpage` вызывает разрыв текста и перенос его на следующую страницу, причем оставшаяся часть текущей страницы остается пустой (текст не растягивается).

## 2.16. Списки

Для получения списков в  $\mathcal{A}\mathcal{M}\mathcal{S}$ - $\mathcal{T}\mathcal{E}\mathcal{X}$ ’е можно использовать как традиционные команды получения списков Plain  $\mathcal{T}\mathcal{E}\mathcal{X}$ ’а, так и собственные средства.

В Plain  $\mathcal{T}\mathcal{E}\mathcal{X}$ ’е для получения списков используются макрокоманды `\item` и `\itemitem`. Так, например, если ввести

```
\item{1.} Это первый из нескольких перенумерованных случаев,
занимающий целый абзац. Отступ равен абзацному отступу.
\itemitem{a)} Это первый подслучай.
\itemitem{b)} Это второй подслучай. Заметим, что подслучаи имеют
в два раза больший отступ.
\item{2.} Второй случай аналогичен.
```

то получится следующий результат:

1. Это первый из нескольких перенумерованных случаев, занимающий целый абзац. Отступ равен абзацному отступу.
  - a) Это первый подслучай.
  - b) Это второй подслучай. Заметим, что подслучаи имеют в два раза больший отступ.
2. Второй случай аналогичен.

Средствами  $\mathcal{A}\mathcal{M}\mathcal{S}$ - $\mathcal{T}\mathcal{E}\mathcal{X}$ ’а списки получаются конструкцией

```
\roster
\item ...
...
\item ...
\endroster
```

`\roster` часто используется внутри абзаца, так что текст после окончания списка не начинается с нового абзаца, т.е. список прерывает абзац наподобие выключной формулы.

Каждый `\item` списка по умолчанию нумеруется автоматически. В стиле препринт элементы списка будут помечены как (1), (2), ... . Можно задавать конкретный номер при помощи факультативного аргумента. Так, если задать

```
\item[3] ...
```

то элемент списка получит номер три независимо от его положения в списке. Последующие `\item` будут нумероваться, начиная с 4. Разумеется, `\item[3]` на самом деле даст пометку типа (3), или (iii), или даже [e] — это зависит от стиля, но этим тоже можно управлять: если вы наберете

```
\item"{\bf 5}" ...
```

то элемент списка получит номер **5** — точно так, как набрано.

Размер отступа может быть отрегулирован в соответствии с шириной пометок. Для этого, прежде чем открыть `\roster`, надо набрать, например, `\widestnumber\item{(viii)}`. Такое соглашение временное: принятое по умолчанию значение будет восстановлено после команды `\endroster`.

Для ссылок на конкретный \item  $\mathcal{A}\mathcal{M}\mathcal{S}$ -TeX имеет команду \therosteritem. Если набрать \therosteritem, то в зависимости от стиля получится (7) или (vii) или [g] ...

Некоторые авторы предпочитают первое условие помещать внутри абзаца и лишь последующие располагают со специальным отступом. Таким образом, получается: (1) первое условие внутри текущего абзаца;

(2) второе условие без всякого дополнительного пробела перед ним;

(3) последнее условие, после которого следует небольшой дополнительный пробел.

Чтобы получить такой список, нужно вместо первого \item набрать \runitem.

Когда первый элемент списка оформляется подобным образом, может оказаться желательным, чтобы последующие строки первого элемента списка Для этого надо поставить \Runitem непосредственно перед абзацем, в котором появился \roster\runitem. Тогда получится (1) первое условие внутри текущего абзаца;

(2) второе условие без всякого дополнительного пробела перед ним;

(3) последнее условие, после которого следует небольшой дополнительный пробел.

\Runitem будет действовать только на первый \roster в этом абзаце. Если же нужны два подобных списка в одном абзаце, то следует набрать

```
\Runitem ... \roster\runitem
...
\endroster\Runitem
... \roster\runitem ... \endroster
```

При получении списка командой \roster, можно регулировать величину отступа так, чтобы поместилась ширина номеров пунктов. Просто надо перед началом \roster набрать, например, \widestnumber\item{(viii)}. Это новое значение временное, и после \endroster восстанавливается величина отступа, принятая по умолчанию.

## 2.17. Цитаты

Как и для списков, для получения цитат в  $\mathcal{A}\mathcal{M}\mathcal{S}$ -TeX'e можно использовать как традиционные команды Plain TeX'a, так и собственные средства. В Plain TeX'e цитаты получаются командой \narrower. Конструкция

```
{\narrower\smallskip\noindent
У этого абзаца будут более узкие строки, чем вокруг,
потому что он использует команду Plain TeX a
\narrower. Прежние поля будут восстановлены после
окончания группы.\smallskip}
```

дает следующий результат:

У этого абзаца будут более узкие строки, чем вокруг, потому что он использует команду Plain TeX'a . Прежние поля будут восстановлены после окончания группы.

Второй \smallskip в этом примере заканчивает абзац. Важно закончить абзац перед окончанием группы, иначе действие \narrower исчезнет еще до того, как TeX начнет выбирать разрывы строк.

В  $\mathcal{A}\mathcal{M}\mathcal{S}$ -TeX'e для цитат предназначена структура \block ...\endblock, которая пропустит небольшой интервал по вертикали, напечатает цитату с отступом от правого и левого поля и оставит еще небольшой пробел по вертикали. Ее следует использовать в середине абзаца для указания цитат, взятых из другого источника. Например, если ввести

```
\block
Этот абзац получен описываемой конструкцией  $\mathcal{A}\mathcal{M}\mathcal{S}$ -TeX'a.
Читатель может сравнить полученный результат с тем, что
производит описанная выше команда Plain TeX'a \narrower.
\endblock
```

то получится следующее:

Этот абзац получен описываемой конструкцией  $\LaTeX$ 'а. Читатель может сравнить полученный результат с тем, что производит описанная выше команда Plain  $\TeX$ 'а.

Не следует оставлять перед конструкцией `\block ... \endblock` пустую строку или ставить `\par`, иначе перед цитатой получится дополнительный пробел.

## 2.18. Таблицы

В  $\LaTeX$  нет специальных средств для набора таблиц. Для этой цели можно применять команды plain  $\TeX$ 'а `\settabs` и `\halign` (их описание см. в книге Д. Кнута *Все про  $\TeX$* . Более сложные пакеты макрокоманд для таблиц имеются в других источниках. Также см. ниже раздел **Вставки с подписями**.

## 2.19. Вставки с подписями

Рисунки, таблицы и некоторые другие виды объектов часто обрабатываются как вставки. Эти объекты готовятся отдельно от основного документа, а затем в него вклеиваются, для чего для них надо оставлять место. Такие объекты обычно имеют подписи; подписи могут быть расположены над ними (для таблиц) или под ними (для рисунков).

Можно задавать вставки какверху страницы, так и в ее “середине”, т.е. прямо в том месте, где она встречается в тексте. Такие вставки набираются, соответственно, командами `\topinsert` и `\midinsert`. Подписи могут помещаться над и под вставкой с помощью, соответственно, `\topcaption` и `\botcaption`.

Для вставок с подписямиверху применяется следующая структура:

```
\topinsert или \midinsert
\captionwidth{⟨размер⟩} (не обязательно)
\topcaption{⟨метка подписи⟩}
⟨необязательный текст подписи⟩
\endcaption
\vspace{⟨размер⟩} или ⟨необязательная кодировка тела вставки⟩
\endinsert
```

Аналогично, для вставки с подписьювнизу:

```
\topinsert или \midinsert
\captionwidth{⟨размер⟩} (не обязательно)
\vspace{⟨размер⟩} или ⟨необязательная кодировка тела вставки⟩
\botcaption{⟨метка подписи⟩}
⟨необязательный текст подписи⟩
\endcaption
\endinsert
```

Здесь запись `⟨размер⟩` означает допустимый размер  $\TeX$ 'а.

Чтобы оставить место для последующей вклейки объекта, следует использовать опцию `\vspace{⟨размер⟩}`. Значением `⟨размер⟩` должна быть точная высота вклеиваемого объекта, поскольку величина пробелов вокруг объекта и подписи зависит от стиля документа и устанавливается автоматически.

Чтобы перекрыть заданную по умолчанию и определяемую стилем ширину подписи, можно использовать `\captionwidth{⟨размер⟩}`.

`⟨метка подписи⟩` — это нечто вроде “Рис. 1” или “Таблица 2а.” Не ставьте в конце знака пунктуации; он уже предусмотрен. Метки подписи печатаются капителью. Иногда желателен некоторый пояснительный текст, который задается в области `⟨необязательный текст подписи⟩`. В стиле препринт он печатается романским шрифтом. Даже если нет пояснительного текста, должна присутствовать команда `\endcaption`.

Например, если ввести

```
\midinsert
```

```

\vspace{.5in}
\captionwidth{17pc}
\botcaption{Рис. 2.19.1}Пример вставки с подписью внизу
\endcaption
\endinsert

```

то получится

Рис. 2.19.1.. Пример вставки с подписью  
внизу

Если вы решили для рисунков, таблиц или других подписанных объектов использовать кодировку TeX'a, то опустите команду `\vspace`, а в подходящем месте (перед `\botcaption` или после `\endcaption` для `\topcaption`) поместите эту кодировку.

Иногда таблица настолько маленькая, что нет необходимости помещать ее во вставку. Если подпись должна размещаться над такой таблицей, то это указывается так:

```

\topcaption{⟨метка подписи⟩}
⟨необязательный текст подписи⟩
\endcaption
⟨кодировка тела таблицы⟩

```

Если подпись размещается внизу, входные данные должны выглядеть так:

```

⟨кодировка тела таблицы⟩
\botcaption{⟨метка подписи⟩}
⟨необязательный текст подписи⟩
\endcaption

```

Чтобы не столкнуться с проблемой разбиения страниц, такой вид “вставки” следует использовать только для очень маленьких объектов.

## 2.20. Сноски

Чтобы получить сноску, которую вы сейчас читаете,<sup>1</sup> надо просто набрать

```

Чтобы получить сноску, которую вы сейчас
читаете, \footnote{Не следует злоупотреблять
сносками!} надо просто набрать ...

```

В стиле `asmppt` (препринт) сноски нумеруются подряд по всему документу цифрами в верхних индексах: <sup>1</sup>, <sup>2</sup>, ... . Но иногда авторам требуется что-нибудь другое — например, пометить в некоторой части текста (в заголовках или предисловии) сноски звездочками \*, \*\*, и т.д. Чтобы получить желаемую метку, можно воспользоваться командой `\footnote` с аргументом “в виде литеры”. Например, для сноски, помеченной звездочкой\*, надо набрать:

```

\footnote"*{Как, например, эта.}

```

Пробелы после первой " и перед второй " *не игнорируются*. Если же набрать `\footnote"{ ... }`, то знак сноски будет отсутствовать, а внизу страницы появится примечание.

Кроме аргумента “в виде литеры”, стиль препринт позволяет пользоваться командой `\footnote` с необязательным аргументом. Например, если набрать

```

\footnote [2]{ ... }

```

---

<sup>1</sup>Не следует злоупотреблять сносками!

\*Как, например, эта.

то получится сноска с меткой <sup>2</sup> (конечно же, в стиле `asmrpt` — в других стилях метка может быть другой). Сноска с необязательным аргументом прерывает стандартную нумерацию, но не действует на нумерацию сносок, получаемых командами `\footnote` без необязательных аргументов: как только появится `\footnote{ ... }`, прерванная нумерация продолжится, как будто эта команда `\footnote[ ... ]{ ... }` и не вводилась.

В тех случаях, когда команда `\footnote` не работает (например, в выключных формулах), ее можно разложить на пару команд:

```
\footnotemark\footnotetext{ ... }
```

которые эквивалентны команде `\footnote`: `\footnotemark` даст очередную метку сноски в основном тексте, а `\footnotetext` — текст сноски внизу страницы, метка которого будет той же, что у последнего `\footnotemark` или у команды `\footnote`. Например, если ввести:

```
$$
y=x \quad \text{для некоторого}\footnotemark\ $x>0$
$$
\footnotetext{Необычное место для сноски!}%
```

то получится

$$y = x \quad \text{для некоторого}^2 x > 0$$

Обратите внимание, что `\footnotetext` находится вне формулы, но следует позаботиться, чтобы она располагалась достаточно близко от `\footnotemark`, чтобы текст сноски появился на той же странице. Пробел *перед* `\footnotemark` всегда игнорируется, но *после* `\footnotemark` нужен `\`, поскольку тут пробел автоматически не вставляется. Следует также быть внимательным и не вводить лишний пробел перед текстом, который следует за `\footnotetext`.

Если же у вас возникло экзотическое желание поместить две сноски в одной выключной формуле, то вам следует внутри этой формулы поместить две команды `\footnotemark`, а сразу же после нее — две команды `\footnotetext`. Но тогда у текста из первого `\footnotetext` будет номер второго `\footnotemark!` Эту проблему решает команда `\adjustfootnotemark`, которая изменяет номер сноски. Конструкция

```
\adjustfootnotemark{-1}%
\footnotetext{ ... }%
\adjustfootnotemark{1}%
\footnotetext{ ... }%
```

сначала уменьшит текущий номер на 1, что даст правильный номер для первой `\footnotetext`, а затем увеличит его на 1, и получится правильный номер для второй `\footnotetext`. Команду `\adjustfootnotemark` можно использовать и для того, чтобы с какого-то места заново начать нумерацию сносок.

Обе команды `\footnotemark` и `\footnotetext` допускают использование аргумента “в виде литеры”, так что можно получать двойные сноски<sup>15,16</sup>, например, введя

```
... так что можно получать двойные
сноски\footnotemark"^{15,16}$", например,
\footnotetext"^{15}$"{Первая сноска}%
\footnotetext"^{16}$"{Вторая сноска}%
введя ...
```

## 2.21. Заголовки

В стиле `asmrpt` имеется четыре уровня заголовков (не считая `\title`):

```
\specialhead... \endspecialhead
\head... \endhead
```

<sup>2</sup>Необычное место для сноски!

<sup>15</sup>Первая сноска

<sup>16</sup>Вторая сноска

```
\subhead... \endsubhead
\subsubhead... \endsubsubhead
```

В большинстве статей заголовком первого уровня будет `\head ... \endhead`, а заголовком второго уровня (подзаголовком) — `\subhead ... \endsubhead`. Заголовок этого раздела был набран так:

```
\head {\smc 2. Подготовка текста в \TeX'e}
\endhead
```

А подзаголовок этого подраздела — так:

```
\subhead 2.21. Заголовки \endsubhead
```

Точки после подзаголовков проставляются автоматически, так что набирать их не надо.

Для длинных статей, которые имеют дополнительное подразделение, имеется `\specialhead` для указания уровня, выше чем `\head`. В стиле препринт `\specialhead` использует жирный шрифт и справа не выровнен; `\head` — капитель и центрирован; `\subhead` — жирный шрифт, прижат влево и включен в следующий текст; `\subsubhead` — курсив, включен в следующий текст, причем делается абзацный отступ и автоматически проставляется точка. Обратите внимание, что в этом руководстве эти соглашения несколько изменены.

Автоматическую вставку точки в подзаголовках или в подподзаголовках можно отменить, если поместить после команды `\subhead` или, соответственно, `\subsubhead` команду `\nofrills`.

В `\head` или `\specialhead` явное разбиение строк достигается командой `\\`, но в `\subhead` и `\subsubhead`, которые являются частью абзацев, надо просто использовать `\linebreak`, как в обычных абзацах.

Если вы готовите монографию, стиль заголовков будет другим. Подробности приведены ниже в разделе **Форматирование книги**.

## 2.22. Библиографические ссылки

В большинстве математических статей ссылки на литературу представляют собой цифру или идентификатор, заключенный в квадратные скобки:

Из теоремы 10 из [4], используя результаты, опубликованные в [D-K], получим ...

Иногда в квадратные скобки помещают и дополнительную информацию:

Из теоремы [4, Теорема 10], используя результаты, опубликованные в [D-K], получим ...

Команда `\cite` производит библиографическую ссылку, напечатанную романским шрифтом и заключенную (при использовании стиля `amsrpt` в квадратные скобки. Приведенный выше пример был подготовлен так:

```
Из теоремы \cite{4, Теорема 10}, используя результаты,
опубликованные в \cite{D-K}, получим ...
```

## 2.23. Теоремы и доказательства

В математических работах принято выделять формулировки определений, теорем, лемм, предположений, доказательств и т. д. `AMS-TeX` предоставляет пользователю такую возможность, причем оформление этих структур зависит от стиля документа.

В стиле `amsrpt` для таких целей имеются следующие возможности:

```
\proclame... \endproclame
\demo... \enddemo
\definition... \enddefinition
\example... \endexample
\remark... \endremark
```

Для формулировок утверждений (теорем, лемм и т. п.) служит конструкция

```
\proclaim... \endproclaim
```

Например, если ввести

```
\proclame{Теорема Пифагора}В прямоугольном треугольнике
квaдрaт гипотенузы равен сумме квадратов катетов.
\endproclaim
```

то получится

**Теорема Пифагора.** *В прямоугольном треугольнике квадрат гипотенузы равен сумме квадратов катетов.*

Название теоремы напечатается жирным шрифтом, точка после него проставится автоматически, а сама формулировка теоремы будет напечатана наклонным шрифтом. Обратите внимание, что перед и после теоремы вставлены дополнительные пробелы.

Доказательства оформляются конструкцией

```
\demo ..\enddemo
```

Например, если ввести

```
\demo{Доказательство{\gtr 1}} Доказательство теоремы Пифагора можно
найти в любом учебнике геометрии.
\enddemo
```

то получится

*Доказательство 1.* Доказательство теоремы Пифагора можно найти в любом учебнике геометрии.

Текст, заключенный в фигурные скобки, напечатается курсивом, знак препинания после него (точка, двоеточие или что-нибудь другое — это определяется стилем) проставится автоматически, а текст внутри конструкции будет печататься обычным прямым шрифтом. Обратите внимание на команду `\gtr` в заголовке доказательства: в большинстве издательств требуется, чтобы даже в курсивном тексте цифры были прямые.

Конец доказательства может быть помечен специальным знаком ‘□’, отделенным от предшествующего ему материала пробелом, величиной в типографский квадрат `\quad`, что получается командой `\qed`. Опять-таки перед и после так набранного доказательства вставлены дополнительные пробелы.

Если вы забудете поставить `\endproclame`, то как только встретится следующая `\demo`,  $\LaTeX$  выдаст сообщение об ошибке. Пропуск же `\enddemo` никогда не приведет к сообщению об ошибке, поскольку никогда нельзя быть уверенным, что `\demo` закончилось: иногда `\demo` содержит внутри себя другое `\proclame`, а иногда даже `\demo` для этого `\proclame`.

`\definition` и `\example` в стиле препринт идентичны: оба оставляют дополнительный промежуток перед заголовком, сам заголовок печатается жирным шрифтом, а точка в конце заголовка проставляется автоматически. Конструкция `\remark` аналогична `\demo` за тем исключением, что в отличие от `\enddemo \endremark` не вставляет дополнительный вертикальный пробел.

Шрифт, которым печатается заголовок, скажем, теоремы, можно легко изменить, поставив перед этим заголовком команду шрифта. Например, достаточно набрать

```
\proclaim{\smc Основная теорема} ...
\endproclaim
```

и заголовок теоремы будет напечатан капиталью:

ОСНОВНАЯ ТЕОРЕМА. ...

Но если вы хотите вместо точки в конце заголовка поставить, скажем, восклицательный знак, то для этого надо отменить автоматическое форматирование. Это делается командой `\nofrills`. Так, если набрать

```
\proclaim\nofrills{\smc Основная теорема!} ...
\endproclaim
```

то получится

ОСНОВНАЯ ТЕОРЕМА! ...

Следует иметь в виду, что при использовании `\nofrills` не только отменяется автоматическая вставка знака препинания, но также и не вставляется пробел после заголовка, поэтому лучше набирать

```
\proclaim\nofrills{\smc Основная теорема!\usualspace} ...
\endproclaim
```

Команда `\usualspace` вставляет после знака пунктуации обычный пробел.

#### 2.24. Присоединение дополнительных файлов

Если у вас длинная рукопись, вы можете разбить ее на части, готовить и обрабатывать каждую часть отдельно, а потом соединить эти части вместе. Для этой цели  $\TeX$  использует управляющую последовательность `\input`: вставить во входной файл команду `\input file.tex` равносильно помещению в это место целиком файла `file.tex`. Тот же эффект будет, если ввести просто `\input file` — при отсутствии явно заданного расширения файла  $\TeX$  по умолчанию берет файл с расширением `.tex`. Если вставляется файл с каким-нибудь другим расширением, то его имя надо задавать полностью, например, `\input file.def`.

Например, если у вас есть, скажем, собственная библиотека макроопределений `macros.def`, которую вы используете практически в каждой работе, то нет необходимости вводить все ее команды вначале каждого файла, а надо просто поместить туда `\input macros.def`, а если разделы статьи уже готовы и находятся в файлах `chap01.tex`, `chap02.tex`, ..., то входной файл для этой статьи не будет содержать почти ничего, кроме `\input chap01.tex`, `\input chap01.tex`, ...

#### 2.25. Текст в рамке и другие украшения

Иногда при подготовке текста возникает желание как-то выделить какую-то его часть, например Поместить текст в рамке, подчеркнуть или надчеркнуть его, или экзотическое желание охватить текст скобкой сверху или скобкой снизу. Для получения подобного украшения в  $\mathcal{A}\mathcal{M}\mathcal{S}$ - $\TeX$ 'е имеются управляющие последовательности `\boxed`, `\underline`, `\overline`, `\underbrace` и `\overbrace`. Правда, эти последовательности работают только в математической моде, но, проявив некоторую изобретательность (а именно, описанную ниже в разделе **Текст в формулах** команду `\text`), можно использовать их и в тексте. Например, приведенное выше было получено следующими командами:

```
... например $\boxed{\text{Поместить текст в рамке}}$,
$\underline{\text{подчеркнуть}}$ или
$\overline{\text{надчеркнуть}}$ его,
или экзотическое желание охватить текст
$\underbrace{\text{скобкой сверху}}$ или
$\overbrace{\text{скобкой снизу}}$.
```

Если так украшенный текст надо выделить на отдельной строке, то следует использовать так называемую “выключку” — конструкцию `$$ ... $$`. Например, для получения

Выключный текст в рамке

надо ввести

```
$$\boxed{\text{Выключный текст в рамке}}$$,
```

Подробнее о математических модах и доступных в них командах рассказывается в разделе **Набор математики**.

### 3. ШРИФТЫ, ИСПОЛЬЗУЕМЫЕ В $\mathcal{A}\mathcal{M}\mathcal{S}$ - $\mathcal{T}\mathcal{E}\mathcal{X}$ 'Е

Кроме стандартных шрифтов гарнитуры Computer Modern, разработанных Дональдом Кнутом и входящих в стандартный дистрибутив  $\mathcal{T}\mathcal{E}\mathcal{X}$ 'а, специально для системы  $\mathcal{A}\mathcal{M}\mathcal{S}$ - $\mathcal{T}\mathcal{E}\mathcal{X}$  были созданы новые шрифты. Это как шрифты гарнитуры Computer Modern с размерами, которых ранее не было, так и новые буквенные и символьные шрифты, предназначенные для использования в математических выражениях. Все эти шрифты составляют коллекцию AMSFonts версии 2.1. Перед тем, как вы начнете использовать стиль препринт  $\mathcal{A}\mathcal{M}\mathcal{S}$ - $\mathcal{T}\mathcal{E}\mathcal{X}$ 'а или как-нибудь сослаться на такие шрифты, вам следует установить их на своем компьютере.

#### 3.1. Описание коллекции AMSFonts

Коллекция AMSFonts содержит следующие шрифты в указанных размерах:

- семейство шрифтов Эйлера, все, кроме `euex` в размерах 5, 6, 7, 8, 9 и 10 пунктов:
  - Fraktur (готический), средний и жирный (`eufm eufb`);
  - “романский” курсив, средний и жирный (`eurm` и `eurb`);
  - рукописный (`script`), средний и жирный (`eusm` и `eusb`);
  - расширенный (`extension`) шрифт, совместимый со шрифтом Эйлера (`euex`), в размерах 7, 8, 9 и 10 пунктов;
- дополнительные размеры некоторых математических шрифтов из гарнитуры Computer Modern:
  - жирный математический курсив (`cmmb`), в размерах 5, 6, 7, 8 и 9 пунктов;
  - жирные математические символы (`cmbsy`), в размерах 5, 6, 7, 8 и 9 пунктов;
  - математический расширенный шрифт (`cmex`), в размерах 7, 8 и 9 пунктов;
- дополнительные математические символы, в размерах 5, 6, 7, 8, 9 и 10 пунктов:
  - первая группа, средние (`msam`);
  - вторая группа, включая жирный ажурный, средней толщины (`msbm`);
- кириллические, разработанные Вашингтонским университетом:
  - светлый (`wncyr`), в размерах 5, 6, 7, 8, 9 и 10 пунктов;
  - жирный (`wncyb`), в размерах 5, 6, 7, 8, 9 и 10 пунктов;
  - курсив (`wncyi`), в размерах 5, 6, 7, 8, 9 и 10 пунктов;
  - капитель (`wncysc`), в 10 пунктов;
  - рубленый (`wncyss`), в размерах 8, 9 и 10 пунктов;
  - виртуальные шрифты соответствующего списка (`.vpl`) файлов, чтобы можно было использовать эти шрифты с альтернативными кодировками и транслитерационными схемами;
- “макетный шрифт”, используемый  $\mathcal{A}\mathcal{M}\mathcal{S}$ - $\mathcal{T}\mathcal{E}\mathcal{X}$ 'ом для синтаксического контроля, существует только в виде метрик (`dummy.tfm`) и не имеет формы символов.

Каждый шрифт в каждом размере можно увеличивать семью стандартными магнификациями  $\mathcal{T}\mathcal{E}\mathcal{X}$ 'а, с `\magstep` от 0 до 5, включая `\magstephalf`. Схемы макетов шрифтов приведены в Приложении В.

#### 3.2. Загрузка шрифтов

Некоторые из этих шрифтов автоматически загружаются стилем препринт и становятся в нем доступными, другие могут быть загружены по требованию. Доступные шрифты и команды для их загрузки описаны ниже.

*Шрифты, загружаемые в стиле препринт.* Некоторые шрифты загружаются стилем препринт автоматически для всеобщего использования:

- `scscs8` новый размер шрифта капитель в Computer Modern.
- `cmex8` и `cmex7` новые размеры шрифта Computer Modern math extension. `cmex8` используется стилем препринт в аннотациях и других восьмипунктовых контекстах; `cmex7` используется для нижних и верхних индексов.

*Математические шрифты, загружаемые стилем препринт.*

- `msam` и `msbm` — содержат дополнительные символы. Символы и имена, которые их производят, показаны ниже в разделе НАБОР МАТЕМАТИКИ подразделе **Имена символов**. Если вы

не используете стиль препринт, каждый из этих шрифтов должен быть специально загружен, соответственно, командами `\loadmsam` или `\loadmsbm`.

- `eufm` — это шрифт medium-weight Euler Fraktur (готический). Если не используется стиль препринт, его также можно загрузить командой `\loadeufm`.

*Математические шрифты, загружаемые `\loadbold`.* Подробно доступ к конкретным символам этих шрифтов описан в разделе НАБОР МАТЕМАТИКИ в подразделах **Жирные символы в математической моде** и **Жирные греческие буквы**.

- `cmmb` — это жирный математический курсив Computer Modern. Также содержит жирные греческие буквы.
- `cmbsy` — содержит жирные математические символы Computer Modern.

*Дополнительные шрифты Эйлера для использования в математике, загружаемые `\loadeu...`*

- `eufb` — это жирный Fraktur (`\loadeufb`).
- `eusm` — это medium-weight рукописный (`\loadeusm`).
- `eusb` — это жирный рукописный (`\loadeusb`).
- `eurm` — это medium-weight “романский курсив” (`\loadeurm`).
- `urb` — это жирный “романский курсив” (`\loadeurb`).

*Вашингтонская кириллица.* Названия книг в библиографиях изданий AMS традиционно приводятся на языке оригинала. Для книг, изданных на русском или других славянских языках. Это часто приводит к необходимости использовать кириллический алфавит. Кириллический шрифт разработан в AMS, причем в качестве модели были взяты шрифты `am`. Шрифты с макетом AMS включены в коллекцию AMSFonts с разрешения разработчиков из Вашингтонского университета. Кириллические шрифты основаны на форме букв шрифтов гарнитуры Computer Modern. Стили набора включают обычный прямой, жирный (основанный на CM жирный расширенный), капитель, курсив и прямой рубленый. Основные текстовые шрифты (прямой, курсив и жирный) представлены в размерах от 5 до 10 пунктов; рубленый — в размерах 8, 9 и 10 пунктов; капитель — только в размере 10 пунктов.

Вашингтонская кириллица не входит в распространяемый дистрибутив `AMS-TeX`’а и все подробности приведены здесь только для информации.

*Размышления и предупреждения.* Команды для загрузки указанных шрифтов должны находиться в преамбуле между строками `\documentstyle{...}` и `\topmatter`. Каждая команда `\load...` загружает соответствующие шрифты (включая индексные размеры), присваивает им номер “математического семейства” и определяет команду математического шрифта. Имена таких команд те же самые, что и имена шрифтов: `\eufm`, `\eufb`, `\eusm`, `\eusb`, `\eurm` и `\urb`. Используются они точно так же, как и `\roman` или `\bold`, например, `\eufb{M}` или `\eufb M`. Для шрифтов `\eufm` (средний Euler Fraktur) `AMS-TeX` также определяет пары синонимов, `\frac` и `\goth`.

`TeX` в математической моде может вместить в себя только шестнадцать семейств шрифтов; восемь уже определены `plain TeX`’ом до `AMS-TeX`’а, стиль препринт загружает еще три: (`msam`, `msbm` и `eufm`) — всего получается одиннадцать. По этой причине, загружая дополнительные шрифты, будьте осторожны и загружайте только те из них, которые вам понадобятся.

## 4. НАБОР МАТЕМАТИКИ

## 4.1. Основные принципы

Основные преимущества  $\TeX$ 'а особенно ярко видны при наборе математических выражений.  $\TeX$  разработан таким образом, чтобы большинство сложных математических выражений можно было легко вводить. Основная идея в том, что сложные формулы довольно просто собираются из менее сложных формул. Менее сложные формулы, в свою очередь, сделаны из простых комбинаций формул еще меньшей сложности, и т. д.

*Математические моды.* Признаком начала и конца математической формулы, включенной в текст абзаца, является знак  $\$$ , т.е. формула вводится прямо в том месте, где она должна набираться в виде  $\$формула\$$ . Например, если ввести  $\$x+y >3\$$ , то получится  $x + y > 3$ . При наборе математических формул  $\TeX$  работает в *математической моде*.

Различаются *текстовая математическая мода*, как в только что приведенном примере, и *выключная математическая мода*, когда формула не включается в текст абзаца, а печатается на отдельной строке (расположение формулы на этой строке зависит от стиля). Выключные математические формулы заключаются в двойные знаки доллара:  $\$\$формула\$\$$ . Например, если ввести  $\$\$x+y >3.\$\$$ , то получится

$$x + y > 3.$$

Выключка используется для длинных формул или для формул, к которым автор хочет привлечь внимание читателя. Обратите внимание, что знак препинания после выключной формулы должен находиться перед закрывающими знаками  $\$\$$ , поскольку в противном случае он окажется в начале следующей строки: если ввести  $\$\$x+y >3\$\$.$ , то получится следующий странный результат:

$$x + y > 3$$

. Как видите, точка оказалась совсем не там, где вы рассчитывали. Выключная математическая формула сама по себе только временно прерывает текущий абзац — часть абзаца после этой формулы печатается без абзацного отступа.

В выключной математической моде некоторые математические знаки (например, большие операторы) оказываются более крупными, чем в формулах в тексте, а также по другому располагаются пределы у больших операторов, есть также отличие в наборе дробей. Эти и другие особенности будут описаны в соответствующих разделах.

*Пробелы в математической моде.*  $\TeX$  большинство пробелов в формулах делает сам и *игнорирует* любые пробелы, которые вы поставили между знаками  $\$$ . Например, если вы вводите  $\$x\$$  и  $\$2\$$ , то это будет означать то же самое, что  $\$x\$$  и  $\$2\$$ . Можно ввести  $\$(x + y)/(x - y)\$$  или  $\$(x+y)/(x-y)\$$ , но в обоих случаях в результате будет  $(x + y)/(x - y)$ , т. е., формула, в которой знаки  $+$  и  $-$  окружены небольшими дополнительными пробелами, а знак  $/$  — нет. Таким образом, вы не должны запоминать сложные правила распределения математических пробелов и можете использовать пробелы любым способом, который вам нравится. Конечно, пробелы используются еще и для обычных целей, чтобы отметить конец управляющей последовательности.

Тем не менее имеются управляющие последовательности для задания таких пробелов в формулах, которые  $\LaTeX$  не игнорирует:

$\backslash\text{quad}$	“квадрат”, равен 1em или трем обычным пробелам
$\backslash\text{qqquad}$	двойной “квадрат” равен 2em или шести пробелам
$\backslash,$ ( $\backslash\text{thinspace}$ )	тонкий пробел, равен 1/6 квадрата
$\backslash!$ ( $\backslash\text{negthinspace}$ )	отрицательный тонкий пробел, $-1/6$ квадрата
$\backslash;$ ( $\backslash\text{thickspace}$ )	широкий пробел, равен 5/18 квадрата
$\backslash\text{negthickspace}$	отрицательный пробел, равен $-5/18$ квадрата
$\backslash\text{medspace}$	средний пробел
$\backslash\text{negmedspace}$	отрицательный средний пробел

Этими командами можно корректировать автоматически вставляемые пробелы.

*Интерпретация символов клавиатуры.* Все символы клавиатуры в соответствии с обычными соглашениями математических изданий имеют в математических формулах специальную интерпретацию.

- (1) Первый знак \$, который вы вводите, переводит в математическую моду, а второй — возвращает обратно. Так что если пропустить один \$ или ввести слишком много \$, Т<sub>Э</sub>X, вероятно, совершенно запутается, и вы получите некоторое сообщение об ошибке.
- (2) Буквы обозначают символы курсива (от *A* до *Z* и от *a* до *z*), которые математики называют “переменными”. Т<sub>Э</sub>X называет их просто “ординарными символами”, потому что они составляют большую часть математических формул. В Т<sub>Э</sub>X’е существует два варианта буквы *l* нижнего регистра, а именно, *l* и *ℓ* (которую вы получаете, просто вводя `\ell`). Математики в своих рукописях обычно пишут что-то похожее на *ℓ*, но делают это единственно для того, чтобы отличить ее от цифры 1. Этой проблемы нет в печатных математических работах, поскольку курсивная *l* сильно отличается от 1, поэтому принято использовать *l* вместо того, чтобы специально запрашивать *ℓ*.
- (3) Т<sub>Э</sub>X также трактует 18 символов

0 1 2 3 4 5 6 7 8 9 ! ? . | / ‘ ’

как ординарные символы, т. е. не вставляет никаких дополнительных пробелов, когда эти символы следуют один за другим или рядом с буквами. В отличие от букв, эти 18 символов, когда появляются в формулах, остаются в прямом шрифте. Вам не надо о них помнить ничего особенного, за исключением того, что слэш обозначает наклонную черту дроби, а вертикальная черта служит для обозначения “абсолютного значения”:  $\$|x|\$$  дает  $|x|$ . Кроме того, надо отличать 0 и ноль.

- (4) Три символа +, - и \* называются “бинарными операциями”, потому что они оперируют с двумя частями формулы. Например, + — это знак плюс, который используется для суммы двух чисел; - — это знак минус. Звездочка (\*) в математике используется редко, но она тоже ведет себя как бинарная операция. Заметим, что - и \* дают математические символы, абсолютно отличные от тех, которые вы получаете в обычном тексте. Знак дефис (-) становится знаком минуса (−), а поднятая звездочка (\*) опускается на более низкий уровень (\*).

Т<sub>Э</sub>X не рассматривает знак / как бинарную операцию, хотя он обозначает деление (которое в математике считается бинарной операцией). Причина в том, что наборщики традиционно ставят дополнительные пробелы вокруг символов +, - и \* и не ставят их вокруг /. Если бы Т<sub>Э</sub>X считал / бинарной операцией, то формула  $\$1/2\$$  получилась бы в виде  $1 / 2$ , что было бы неправильно; поэтому Т<sub>Э</sub>X рассматривает / как ординарный символ.

Имеются и другие бинарные операции, которые задаются управляющими последовательностями, а не единичными символами.

- (5) Т<sub>Э</sub>X трактует символы =, <, > и : как “отношения”, потому что они выражают отношение между двумя величинами. Например,  $x < y$  означает, что  $x$  меньше, чем  $y$ . Такие отношения значительно отличаются по смыслу от бинарных операций типа +, и эти символы печатаются несколько иначе. Имеются и другие символы отношения, которые задаются управляющими последовательностями.
- (6) Два символа “,” (запятая) и “;” (точка с запятой) трактуются в формулах как *знаки пунктуации*; это означает, что Т<sub>Э</sub>X ставит небольшой дополнительный пробел после них и не ставит до них. В математических формулах не принято ставить дополнительный пробел после точки, поэтому Т<sub>Э</sub>X трактует точку как ординарный символ. Если вы хотите, чтобы символ “:” трактовался как знак пунктуации, а не как отношение, просто вызывайте его командной последовательностью `\colop`. Если вы хотите использовать запятую как обычный символ (например, когда она появляется в большом числе), поставьте ее в фигурных скобках; Т<sub>Э</sub>X трактует все, что находится в фигурных скобках как ординарный символ.
- (7) Символы ( и [ называются “открывающими”, а ) и ] — “закрывающими” *ограничителями*; они прекрасно действуют как ординарные символы, к тому же помогают Т<sub>Э</sub>X’у разобраться, когда бинарные операции в действительности не используются бинарным способом.

Хотя символы { и } указывают группирование, управляющие последовательности `\{и \}` дают открывающую и закрывающую фигурные скобки { и }.

Подробнее ограничители описаны ниже в специальном разделе.

- (8) Затем существует символ `'`, который используется как сокращение для верхнего индекса `\prime`.
- (9) Специальные символы `^` и `_` обозначают верхние и нижние индексы и не должны использоваться вне формул.  $\TeX$  использует такие и аналогичные факты, чтобы обнаружить во входном файле пропущенный знак доллара до того, как такие ошибки вызовут слишком много неприятностей.

Кроме символов клавиатуры, описанных выше, математики используют множество специфических обозначений, которые в  $\TeX$ 'е задаются управляющими последовательностями. Имеются специальные управляющие последовательности и для ординарных символов, и для различных операторов, отношений, ограничителей и т. д., позволяющие получать огромное разнообразие математических формул. Полный список таких обозначений приведен в разделах 4.2 и 5.3.

*Разрыв формул.* Когда в абзаце встречается формула,  $\TeX$  может разбить ее между строками. Это неизбежное зло, такое же, как перенос слов. Хочется избежать его, если только альтернатива не хуже.

Формула будет разбиваться только после символов отношения типа  $=$ ,  $<$  или  $\rightarrow$ , или после символов бинарной операции типа  $+$ ,  $-$  или  $\times$ , когда отношения или бинарные операции находятся на “внешнем уровне” формулы (т.е., не заключены в  $\{ \dots \}$ , не находятся в числителе или знаменателе дроби и т. п.). Например, если вы вводите

$$\text{\$f(x,y) = x^2-y^2 = (x+y)(x-y)\$}$$

в середине абзаца, то есть шанс, что  $\TeX$  разорвет строку либо после знака  $=$  (он предпочитает это), либо после  $-$ ,  $+$  или  $\times$  (в крайнем случае). Но ни в коем случае не будет разрыва после запятой — запятая, после которой желателен разрыв, не должна появляться между знаками  $\$$ . В формулах можно использовать “разрывный знак умножения”: если вы введете  $\text{\$(x+y)\*(x-y)\$}$ , то на месте `\*` будет разрешен разрыв строки так же, как при переносе слов. Однако, вместо вставки знака переноса  $\TeX$  вставит знак  $\times$  в текстовом размере.

Если в этом примере вы не хотите разрешать никаких других разрывов, кроме как после знака  $=$ , то можете ввести

$$\text{\$f(x,y) = \{x^2-y^2\}=\{(x+y)(x-y)\}\$},$$

поскольку дополнительные фигурные скобки “спаивают” подформулы, помещая их в неразрываемые боксы. Имеется средство как для указания места возможного разрыва формулы, так и для запрещения такого разрыва — это, соответственно, команды `\mathbreak` и `\nomathbreak`.

Если вы хотите разрешить разрыв в некоторой точке на внешнем уровне формулы, то можно использовать и команду Plain  $\TeX$ 'а `\allowbreak`. Например, если формула

$$\text{\$(x_1,\ldots,x_m,\allowbreak y_1,\ldots,y_n)\$}$$

появляется в тексте абзаца,  $\TeX$  позволяет разорвать ее на два куска  $(x_1, \dots, x_m, \text{ и } y_1, \dots, y_n)$ .

Но нет необходимости заранее суетиться по поводу таких вещей, пока  $\TeX$  на самом деле неудачно не разорвет формулу, поскольку вероятность этого довольно мала.

## 4.2. Математические символы

Все символы, которые вводятся в математической моде, делятся на несколько групп<sup>3</sup>:

Ординарные	/
Большие операторы	<code>\sum</code>
Бинарные операции	$+$
Отношения	$=$
Ограничители	$($
Знаки пунктуация	,

<sup>3</sup>Это некоторое упрощение того, что происходит на самом деле. Подробности см. в книге Д. Кнута *Все про  $\TeX$* .

От того, к какому виду относится символ, зависит распределение пробелов около него. Это касается не только символов клавиатуры, символов, задаваемых управляющими последовательностями.

На самом деле правила  $\TeX$ 'а по расстановке пробелов в формулах совершенно просты. Формула преобразуется в математический список. Этот список состоит главным образом из “атомов” восьми основных типов: Ord (ординарный или обычный) Op (большой оператор), Bin (бинарная операция), Rel (бинарное отношение), Open (открывающий ограничитель), Close (закрывающий ограничитель), Punct (пунктуация) и Inner (ограниченная подформула). Другие виды атомов все трактуются как Ord, а дроби считаются типа Inner. Для определения пробелов между парами соседних атомов используется следующая таблица:

		<i>Правый атом</i>							
		Ord	Op	Bin	Rel	Open	Close	Punct	Inner
<i>Левый атом</i>	Ord	0	1	(2)	(3)	0	0	0	(1)
	Op	1	1	*	(3)	0	0	0	(1)
	Bin	(2)	(2)	*	*	(2)	*	*	(2)
	Rel	(3)	(3)	*	0	(3)	0	0	(3)
	Open	0	0	*	0	0	0	0	0
	Close	0	1	(2)	(3)	0	0	0	(1)
	Punct	(1)	(1)	*	(1)	(1)	(1)	(1)	(1)
	Inner	(1)	1	(2)	(3)	(1)	0	(1)	(1)

Здесь 0, 1, 2 и 3 обозначают, соответственно, отсутствие пробела, тонкий пробел, средний пробел и толстый пробел. Элемент таблицы заключен в скобки, если пробел вставляется только в выключном и текстовом стилях, а не индексах. Например, в ряде Rel колонки Rel часто встречается (3). Это означает, что обычно перед и после символов отношения типа = вставляется толстый пробел, но в индексах он не вставляется. Некоторые элементы таблицы равны \*. Такие случаи никогда не возникают, поскольку атомам Bin должны предшествовать, а также и следовать за ними, атомы, совместимые с природой бинарных операций.

Приведем список математических символов и команд для их получения, доступных в  $\TeX$ 'е, придерживаясь их принадлежности к указанным группам. Если не утверждается обратное, математические символы доступны только в математической моде. Например, если вы скажете `\alpha` в тексте,  $\TeX$  сообщит об ошибке и попытается вставить знак \$.

#### *Строчные греческие буквы.*

$\alpha$	<code>\alpha</code>	$\iota$	<code>\iota</code>	$\varrho$	<code>\varrho</code>
$\beta$	<code>\beta</code>	$\kappa$	<code>\kappa</code>	$\sigma$	<code>\sigma</code>
$\gamma$	<code>\gamma</code>	$\lambda$	<code>\lambda</code>	$\varsigma$	<code>\varsigma</code>
$\delta$	<code>\delta</code>	$\mu$	<code>\mu</code>	$\tau$	<code>\tau</code>
$\epsilon$	<code>\epsilon</code>	$\nu$	<code>\nu</code>	$\upsilon$	<code>\upsilon</code>
$\varepsilon$	<code>\varepsilon</code>	$\xi$	<code>\xi</code>	$\phi$	<code>\phi</code>
$\zeta$	<code>\zeta</code>	$o$	<code>o</code>	$\varphi$	<code>\varphi</code>
$\eta$	<code>\eta</code>	$\pi$	<code>\pi</code>	$\chi$	<code>\chi</code>
$\theta$	<code>\theta</code>	$\varpi$	<code>\varpi</code>	$\psi$	<code>\psi</code>
$\vartheta$	<code>\vartheta</code>	$\rho$	<code>\rho</code>	$\omega$	<code>\omega</code>

Здесь нет `\omicron`, поскольку она выглядит так же, как  $o$ . Заметим, что буква `\upsilon` ( $\upsilon$ ) здесь чуть шире, чем  $v$  ( $v$ ); и  $tu$ , и другую следует отличать от `\nu` ( $\nu$ ). Аналогично, `\varsigma` ( $\varsigma$ ) не следует путать с `\zeta` ( $\zeta$ ).

*Прописные греческие буквы.* Прописные греческие буквы имеются в трех начертаниях — в прямом, наклонном и жирном.

$\Gamma$	<code>\Gamma</code>	$\Xi$	<code>\Xi</code>	$\Phi$	<code>\Phi</code>
$\Delta$	<code>\Delta</code>	$\Pi$	<code>\Pi</code>	$\Psi$	<code>\Psi</code>
$\Theta$	<code>\Theta</code>	$\Sigma$	<code>\Sigma</code>	$\Omega$	<code>\Omega</code>

$\Lambda$  `\Lambda`                       $\Upsilon$  `\Upsilon`

Остальные прямые греческие прописные буквы печатаются из романского алфавита (`\Alpha`  $\equiv$  `{\rm A}`, `\Beta`  $\equiv$  `{\rm B}`, и т.д).

$\Gamma$ <code>\varGamma</code>	$\Xi$ <code>\varXi</code>	$\Phi$ <code>\varPhi</code>
$\Delta$ <code>\varDelta</code>	$\Pi$ <code>\varPi</code>	$\Psi$ <code>\varPsi</code>
$\Theta$ <code>\varTheta</code>	$\Sigma$ <code>\varSigma</code>	$\Omega$ <code>\varOmega</code>
$\Lambda$ <code>\varLambda</code>	$\Upsilon$ <code>\varUpsilon</code>	

Остальные наклонные греческие прописные буквы печатаются из курсивного шрифта (`\Alpha`  $\equiv$  `{\it A}`, `\Beta`  $\equiv$  `{\it B}`, и т.д).

$\Gamma$ <code>\boldGamma</code>	$\Xi$ <code>\boldXi</code>	$\Phi$ <code>\boldPhi</code>
$\Delta$ <code>\boldDelta</code>	$\Pi$ <code>\boldPi</code>	$\Psi$ <code>\boldPsi</code>
$\Theta$ <code>\boldTheta</code>	$\Sigma$ <code>\boldSigma</code>	$\Omega$ <code>\boldOmega</code>
$\Lambda$ <code>\boldLambda</code>	$\Upsilon$ <code>\boldUpsilon</code>	

Остальные жирные греческие прописные буквы печатаются из романского алфавита жирным шрифтом (`\Alpha`  $\equiv$  `{\bf A}`, `\Beta`  $\equiv$  `{\bf B}` и т. д).

*Рукописные прописные буквы.* Чтобы получить буквы  $\mathcal{A} \dots \mathcal{Z}$ , надо вводить `\Cal A$ ... \Cal Z$`.  $\mathcal{A}\mathcal{M}\mathcal{S}\text{-}\mathcal{T}\mathcal{E}\mathcal{X}$  разрешает использовать в математике и другие алфавиты, что описано в разделе **Шрифты в математической моде**.

*Разнообразные ординарные символы.*

$\aleph$ <code>\aleph</code>	$\prime$ <code>\prime</code>	$\forall$ <code>\forall</code>
$\hbar$ <code>\hbar</code>	$\emptyset$ <code>\emptyset</code>	$\exists$ <code>\exists</code>
$\imath$ <code>\imath</code>	$\nabla$ <code>\nabla</code>	$\neg$ <code>\neg</code> , <code>\not</code>
$\jmath$ <code>\jmath</code>	$\surd$ <code>\surd</code>	$\flat$ <code>\flat</code>
$\ell$ <code>\ell</code>	$\top$ <code>\top</code>	$\natural$ <code>\natural</code>
$\wp$ <code>\wp</code>	$\perp$ <code>\perp</code>	$\sharp$ <code>\sharp</code>
$\Re$ <code>\Re</code>	$\ $ <code>\ </code> , <code>\ </code>	$\clubsuit$ <code>\clubsuit</code>
$\Im$ <code>\Im</code>	$\angle$ <code>\angle</code>	$\diamondsuit$ <code>\diamondsuit</code>
$\partial$ <code>\partial</code>	$\triangle$ <code>\triangle</code>	$\heartsuit$ <code>\heartsuit</code>
$\infty$ <code>\infty</code>	$\backslash$ <code>\backslash</code>	$\spadesuit$ <code>\spadesuit</code>
$\int$ <code>\int</code>	$\dagger$ <code>\dagger</code>	$\ddagger$ <code>\ddagger</code>
$\P$ <code>\P</code>	$\S$ <code>\S</code>	

*“Большие” операторы.* Размер следующих символов различается в зависимости от того, в какой математической моде они используются. В выключенных формулах они имеют больший размер.

$\sum$ <code>\sum</code>	$\bigcap$ <code>\bigcap</code>	$\bigodot$ <code>\bigodot</code>
$\prod$ <code>\prod</code>	$\bigcup$ <code>\bigcup</code>	$\bigotimes$ <code>\bigotimes</code>
$\coprod$ <code>\coprod</code>	$\bigsqcup$ <code>\bigsqcup</code>	$\bigoplus$ <code>\bigoplus</code>
$\bigvee$ <code>\bigvee</code>	$\biguplus$ <code>\biguplus</code>	$\bigwedge$ <code>\bigwedge</code>

Важно отметить отличие этих “больших операторов” от похожих, но меньших символов “бинарных операций”, у которых такие же имена, за исключением приставки `big`. Большие операторы обычно встречаются в начале формулы или подформулы и обычно имеют индексы, а бинарные операции обычно встречаются между двумя символами или подформулами и редко имеют индексы.

Большие операторы `\sum`, `\prod` и `\coprod` также надо отличать от символов `\Sigma` ( $\Sigma$ ), `\Pi` ( $\Pi$ ) и `\amalg` ( $\amalg$ ), соответственно.

Большим операторам в этом руководстве посвящен специальный раздел (см. **Большие операторы**).

*Интегралы.* Интегралы также относятся к “большим операторам” и их размер тоже различается в зависимости от того, в какой математической моде они используются. В выключенных формулах они имеют больший размер.

$\int$	$\int$	<code>\int</code>	$\oint$	$\oint$	<code>\oint</code>
$\iint$	$\iint$	<code>\iint</code>	$\iiint$	$\iiint$	<code>\iiint</code>
$\iiint$	$\iiint$	<code>\iiint</code>	$\int \cdots \int$	$\int \cdots \int$	<code>\idotsint</code>

(Подробнее см. **Большие операторы**).

*Бинарные операции.* Кроме уже упомянутых  $+$  и  $-$ , бинарными операциями считаются и следующие символы:

$\pm$	<code>\pm</code>	$\cap$	<code>\cap</code>	$\vee$	<code>\vee</code>
$\mp$	<code>\mp</code>	$\cup$	<code>\cup</code>	$\wedge$	<code>\wedge</code>
$\setminus$	<code>\setminus</code>	$\oplus$	<code>\oplus</code>	$\otimes$	<code>\otimes</code>
$\cdot$	<code>\cdot</code>	$\sqcap$	<code>\sqcap</code>	$\ominus$	<code>\ominus</code>
$\times$	<code>\times</code>	$\sqcup$	<code>\sqcup</code>	$\otimes$	<code>\otimes</code>
$*$	<code>\ast</code>	$\triangleleft$	<code>\triangleleft</code>	$\oslash$	<code>\oslash</code>
$\star$	<code>\star</code>	$\triangleright$	<code>\triangleright</code>	$\odot$	<code>\odot</code>
$\diamond$	<code>\diamond</code>	$\wr$	<code>\wr</code>	$\dagger$	<code>\dagger</code>
$\circ$	<code>\circ</code>	$\bigcirc$	<code>\bigcirc</code>	$\ddagger$	<code>\ddagger</code>
$\bullet$	<code>\bullet</code>	$\bigtriangleup$	<code>\bigtriangleup</code>	$\amalg$	<code>\amalg</code>
$\div$	<code>\div</code>	$\bigtriangledown$	<code>\bigtriangledown</code>	$\&$	<code>\&amp;</code>

Обратите внимание, что когда  $\dagger$  и  $\ddagger$  используются не как обычные символы, а как бинарные операции, применяются команды `\dagger` и `\ddagger`.

*Бинарные отношения.* Кроме уже упомянутых  $<$ ,  $>$  и  $=$ , бинарными отношениями считаются и следующие символы:

$\leq$	<code>\leq, \le</code>	$\geq$	<code>\geq, \ge</code>	$\equiv$	<code>\equiv</code>
$\prec$	<code>\prec</code>	$\succ$	<code>\succ</code>	$\sim$	<code>\sim</code>
$\preceq$	<code>\preceq</code>	$\succeq$	<code>\succeq</code>	$\simeq$	<code>\simeq</code>
$\ll$	<code>\ll</code>	$\gg$	<code>\gg</code>	$\asymp$	<code>\asymp</code>
$\subset$	<code>\subset</code>	$\supset$	<code>\supset</code>	$\approx$	<code>\approx</code>
$\subseteq$	<code>\subseteq</code>	$\supseteq$	<code>\supseteq</code>	$\cong$	<code>\cong</code>
$\sqsubseteq$	<code>\sqsubseteq</code>	$\sqsupseteq$	<code>\sqsupseteq</code>	$\bowtie$	<code>\bowtie</code>
$\in$	<code>\in</code>	$\ni$	<code>\ni</code>	$\propto$	<code>\propto</code>
$\vdash$	<code>\vdash</code>	$\dashv$	<code>\dashv</code>	$\models$	<code>\models</code>
$\smile$	<code>\smile</code>	$\mid$	<code>\mid</code>	$\doteq$	<code>\doteq</code>
$\frown$	<code>\frown</code>	$\parallel$	<code>\parallel</code>	$\perp$	<code>\perp</code>
$\neq$	<code>\neq, \ne</code>	$\notin$	<code>\notin</code>		

`\mid` и `\parallel` дают те же самые символы, что и  $|$  и  $\parallel$ , но трактуемые как бинарные отношения, поэтому окруженные с обеих сторон дополнительными пробелами.

Можно получить отрицание многих этих отношений, поместив перед ними `\not`. Например `\not\subset` дает  $\not\subset$ . Символ `\not` является символом отношения нулевой ширины, так что он будет перекрывать отношение, которое следует за ним. Его положение не всегда идеально, поскольку некоторые символы отношения шире других. Например, `\not\in` дает  $\notin$ , но лучше иметь более крутое зачеркивание  $\notin$ .

*Стрелки.* Стрелки также относятся к бинарным отношениям, хотя вертикальные стрелки относятся к “ограничителям” со всеми вытекающими последствиями (в частности, они меняют свой размер, когда используются после увеличивающих размеры команд — см. **Ограничители**).

$\leftarrow$	<code>\leftarrow, \gets</code>	$\longleftarrow$	<code>\longleftarrow</code>	$\uparrow$	<code>\uparrow</code>
--------------	--------------------------------	------------------	-----------------------------	------------	-----------------------

$\Leftarrow$	<code>\Leftarrow</code>	$\Leftrightarrow$	<code>\Longleftarrow</code>	$\Uparrow$	<code>\Uparrow</code>
$\rightarrow$	<code>\rightarrow, \to</code>	$\longrightarrow$	<code>\longrightarrow</code>	$\Downarrow$	<code>\downarrowarrow</code>
$\Rightarrow$	<code>\Rightarrow</code>	$\Longrightarrow$	<code>\Longrightarrow</code>	$\Updownarrow$	<code>\Downarrow</code>
$\leftrightarrow$	<code>\leftrightarrow</code>	$\longleftrightarrow$	<code>\longleftrightarrow</code>	$\Uparrow$	<code>\updownarrow</code>
$\Leftrightarrow$	<code>\Leftrightarrow</code>	$\Leftrightarrow$	<code>\Longleftrightarrow</code>	$\Downarrow$	<code>\Updownarrow</code>
$\mapsto$	<code>\mapsto</code>	$\longmapsto$	<code>\longmapsto</code>	$\nearrow$	<code>\nearrow</code>
$\hookrightarrow$	<code>\hookrightarrow</code>	$\hookrightarrow$	<code>\hookrightarrow</code>	$\searrow$	<code>\searrow</code>
$\lleftarrow$	<code>\lleftarrow</code>	$\rightharpoonup$	<code>\rightharpoonup</code>	$\swarrow$	<code>\swarrow</code>
$\lleftarrow$	<code>\lleftarrow</code>	$\rightharpoonup$	<code>\rightharpoonup</code>	$\nwarrow$	<code>\nwarrow</code>
$\leftharpoonup$	<code>\leftharpoonup</code>	$\rightharpoonup$	<code>\rightharpoonup</code>		
$\leftharpoonup$	<code>\leftharpoonup</code>	$\rightharpoonup$	<code>\rightharpoonup</code>		
$\rightleftharpoons$	<code>\rightleftharpoons</code>				

Команда `\iff` дает стрелку наподобие `\Longleftrightarrow`, но с чуть большими пробелами вокруг нее.  $\mathcal{A}\mathcal{M}\mathcal{S}\text{-}\mathcal{T}\mathcal{E}\mathcal{X}$  также имеет `\implies` и `\impliedby`, которые дают точно такие же стрелки, как, соответственно, `\longrightarrow` и `\Longleftarrow`, но опять-таки с чуть большими пробелами вокруг.

Над и под стрелками можно размещать символы для получения новых отношений, например, можно получить такое хитрое отношение  $\xrightarrow{\alpha\beta}$ , в котором  $\alpha\beta$  размещается над стрелкой `\longrightarrow`. Получение таких сложных отношений описано в разделе **Составные символы**. Здесь же мы скажем еще об одном способе, который касается только стрелок.

Если набрать в математической моде `@>>>`, то получится правая стрелка  $\rightarrow$ , а если набрать `@<<<` — левая стрелка  $\leftarrow$ . Если же в такой конструкции между первым и вторым символом поместить некоторое математическое выражение, то оно будет напечатано над стрелкой. Если же выражение поместить между вторым и третьим символом, то оно будет напечатано под стрелкой. Например

```

$@> \alpha+\beta+\gamma >>>$      \xrightarrow{\alpha+\beta+\gamma}
$@< \alpha+\beta+\gamma <<<$      \xleftarrow{\alpha+\beta+\gamma}
$@> \alpha+\beta>\gamma >$       \xrightarrow[\gamma]{\alpha+\beta}

```

Конечно же, если во вставляемом математическом выражении уже есть знаки  $>$  или  $<$ , то во избежание путаницы это выражение следует заключить в фигурные скобки.

```

$@> {x+y}\>\{z}>>>$           \xrightarrow{x+y>z}

```

Здесь вручную вставлены тонкие пробелы вокруг  $>$ , поскольку в индексном размере  $\mathcal{T}\mathcal{E}\mathcal{X}$  автоматически это не делает.

*Ограничители.* Следующие символы относятся к ограничителям:

(	(	[	[, \lbrack	{	{, \lbrace
)	)	]	], \rbrack	}	}, \rbrace
⌊	\lfloor	⌈	\lceil	⟨	\langle
⌋	\rfloor	⌋	\rceil	⟩	\rangle
	, \vert		\ , \Vert	/	/
\	\backslash				

Как было уже упомянуто, ограничителями могут служить и вертикальные стрелки. Ограничителям в этом руководстве посвящен специальный раздел (см. **Ограничители**).

*Пунктуация.* После запятых и точек с запятой, которые встречаются в математических формулах,  $\mathcal{T}\mathcal{E}\mathcal{X}$  помещает тонкий пробел. То же самое он делает и для двоеточия, которое вызывается командой `\colon`. В других случаях двоеточие считается отношением, как в  $x := y$  и в  $a : b :: c : d$ , которые вводятся как `$x:=y$` и `$a:b::c:d$`. Многоточия в математических формулах и команды для их получения обсуждается в специальном разделе **Многоточия**.

$\mathcal{A}\mathcal{M}\mathcal{S}\text{-}\mathcal{T}\mathcal{E}\mathcal{X}$  имеет также большой набор дополнительных символов (точнее, символов из дополнительных шрифтов, описанных в разделе **5. Имена дополнительных символов**).

### 4.3. Верхние и нижние индексы

В математических формулах у символов можно получить верхний индекс <sup>(вверху)</sup> и нижний индекс <sub>(внизу)</sub> используя `^` и `_`, как это показано в следующих примерах:

<i>Вход</i>	<i>Выход</i>
<code>\$x^2\$</code>	$x^2$
<code>\$x_2\$</code>	$x_2$
<code>\$2^x\$</code>	$2^x$
<code>\$x^2y^2\$</code>	$x^2y^2$
<code>\$x ^2y ^2\$</code>	$x^2y^2$
<code>\$x_2y_2\$</code>	$x_2y_2$
<code>\$_2F_3\$</code>	${}_2F_3$

Если на вашем компьютере нет клавиш `^` и `_`, то их можно заменить командами `\sp` для получения верхних индексов и `\sb` — для нижних. Если за этими командами следует буква, то ее обязательно следует отделить пробелом.

Можно иметь одновременно верхний и нижний индексы и указывать их в любом порядке:

<code>\$x^2_3\$</code>	$x_3^2$
<code>\$x_3^2\$</code>	$x_3^2$
<code>\$x^{\{31415\}_\{92\}}+\pi\$</code>	$x_{92}^{31415} + \pi$
<code>\$x_{\{y^a_b\}^{\{z_c^d\}}\$</code>	$x_{y_b^c}^{z_d^a}$

Заметим, что одновременные индексы <sup>верхний</sup>/<sub>нижний</sub> располагаются один под другим. Однако многие математики в некоторых ситуациях предпочитают располагать верхние и нижние индексы на разных расстояниях от буквы. Можно вынудить `TeX` отодвинуть индекс, вставив пустую группу `{}`:

<code>\$x_ i{\}^2\$</code>	$x_i^2$
<code>\$R_i{\}^{\{jk\}}_1\$</code>	$R_i^{jk}_1$

Команды `^` и `_` действуют только на единственную литеру после нее, так что следующие записи не вызовут никаких недоразумений:

<code>\$x^2y^2\$</code>	$x^2y^2$
<code>\$x ^2y ^2\$</code>	$x^2y^2$
<code>\$x_2y_2\$</code>	$x_2y_2$
<code>\$_2F_3\$</code>	${}_2F_3$

Если вы хотите в качестве верхнего или нижнего индекса использовать несколько символов, заключите их в фигурные скобки:

<code>\$x^{\{2y\}}\$</code>	$x^{2y}$
<code>\$2^{\{2^x\}}\$</code>	$2^{2^x}$
<code>\$2^{\{2^{\{2^x\}}\}}\$</code>	$2^{2^{2^x}}$
<code>\$y_{\{x_2\}}\$</code>	$y_{x_2}$
<code>\$y_{\{x^2\}}\$</code>	$y_{x^2}$
<code>\$2^{\{32\}}\$</code>	$2^{32}$
<code>\$x^{\alpha}\$</code>	$x^\alpha$

Фигурные скобки здесь используются, чтобы указать “подформулы”, т. е., простые части более сложной формулы. Фигурные скобки служат и для обычных целей группирования.

Обратите внимание, что верхний индекс `32` представляет собой два символа, а `\alpha` — всего лишь один символ, поэтому управляющую последовательность `\alpha` не надо заключать в фигурные скобки. Тем не менее с управляющими последовательностями следует обращаться осторожно,

и вот почему. Предположим, например, вам нужен символ  $A_{\neq}$ . По аналогии с  $A_{\alpha}$  поместим команду `\ne` в индекс и не заключим ее в фигурные скобки. Получим нечто странное:

$$\text{\$A\_ne\$} \qquad A_{\neq}$$

Фокус в том, что `\ne` — вовсе не один обособленный символ, а просто аббревиатура для `\not=`. Так что  $\text{T}\text{E}\text{X}$ , получив команду `\$A\_ne\$`, оттранслирует ее в `\$A\_not=\$`, после чего поставит / в качестве нижнего индекса буквы A. Хотя такие ситуации крайне редки, фигурные скобки вокруг командной последовательности вам не помешают.

Вы, конечно, обратили внимание, что индексы печатаются более мелким шрифтом, а повторные индексы — еще мельче.  $\text{T}\text{E}\text{X}$  печатает индексы в так называемом *стиле индекса* и *стиле повторного индекса*. Эти стили не только меняют размер символов, они также меняют правила автоматической расстановки пробелов.

Чтобы показать, что верхний или нижний индекс относится ко всему выражению, математики пользуются круглыми, квадратными или фигурными скобками:

$$\begin{aligned} \text{\$(x+1)^3\$} & (x+1)^3 \\ \text{\$(x^2)^3\$} & (x^2)^3 \\ \text{\$[x^2]^3\$} & [x^2]^3 \\ \text{\$\{x^2\}^3y\$} & \{x^2\}^3y \end{aligned}$$

$\text{T}\text{E}\text{X}$ , пунктуально следуя инструкции, ставит 3 в качестве верхнего индекса правой скобки. Если же вы заключите формулу в фигурные скобки, показатель степени будет относиться ко всему выражению:

$$\begin{aligned} \text{\$\{(x^2)\}^3\$} & (x^2)^3 \\ \text{\$[{x^2}]^3\$} & [x^2]^3 \\ \text{\$\{\{x^2\}\}^3y\$} & \{x^2\}^3y \\ \text{\$\{((x^2)^2)\}^4\$} & ((x^2)^2)^4 \end{aligned}$$

Иногда требуется получить такое выражение

$$\text{\$a^{\{b^c\}\$} \qquad a^{b^c}$$

Поскольку вся формула  $b^c$  служит верхним индексом, то выражение  $b^c$  следует заключить в фигурные скобки. Освоив этот принцип, можно выдавать самые разнообразные формулы:

$$\begin{aligned} \text{\$a^{\{b^{\{c+1\}\}\$} & a^{b^{c+1}} \\ \text{\$2^{\{2^x\}\$} & 2^{(2^x)} \\ \text{\$2^{\{2^{\{2^x\}\}\$} & 2^{2^{2^x}} \\ \text{\$2^{\{(a+b)^2\}\$} & 2^{(a+b)^2} \\ \text{\$x_{y_2}\$} & x_{y_2} \\ \text{\$x_{y^2}\$} & x_{y^2} \end{aligned}$$

Иногда бывает нужно использовать в качестве верхнего индекса какой-нибудь акцент (см. раздел **Акценты в математике**), например  $\hat{\cdot} : (I+M)^{\hat{\cdot}}$ . Чтобы получить “крышечку” в таком качестве, нельзя набрать `^{\hat{}}` или `\sp{\hat{}}`, так как `\hat{}` — это не символ, а команда для получения акцента над чем-либо. У  $\mathcal{A}\mathcal{M}\mathcal{S}$ - $\text{T}\text{E}\text{X}$  имеется `\sphat`, которая работает так же, как вы могли бы ожидать от `\sp{\hat{}}`, а также есть `\spcheck`, ..., `\spvec`.

Верхние или нижние индексы могут быть представлены и бинарным оператором

$$\begin{aligned} \text{\$x_{ij}^*\$} & z_{ij}^* \\ \text{\$f^*(x) \cap f_*(\nu)\$} & f^*(x) \cap f_*(\nu) \end{aligned}$$

Более того, можно даже получить что-то вроде

$$\text{\$f_+\$} \qquad f_+$$

`$f_ -$`  $f_-$

Последние формулы выглядят лучше, если использовать дополнительные фигурные скобки

`$f_ {+}$`  $f_+$   
`$f_ {-}$`  $f_-$

Кроме верхних и нижних индексов математики часто используют обозначение  $f'$ .  $\text{\TeX}$  имеет управляющую последовательность `\prime`, но если вы наберете

`$f\prime$`  $f'$

то получите совсем не то, что хотели. Штрихи следует употреблять как верхние индексы:

`$f^\prime$`  $f'$

Когда  $\text{\TeX}$  находится в математической моде, он транслирует ' в `^\prime`; более того, '' транслируется в `^\prime\prime`, а ''' — в `^\prime\prime\prime`, и т. д.

`$f'[g(x)]g'(x)$`  $f'[g(x)]g'(x)$   
`$y_1'+y_2''+y_3'''$`  $y_1' + y_2'' + y_3'''$

У  $\mathcal{AMS}\text{-}\text{\TeX}$ 'а нет специального способа для получения штрихов в нижнем индексе, поскольку они довольно редко бывают нужны, поэтому для получения, например,  $F'_i(w, z)$  надо просто набрать

`$F_ \prime(w,z)$`

Иногда бывает нужен `\prime` в случаях вроде этого:

`$g{^\prime2}$`  $g'^2$

Здесь также можно воспользоваться пустой группой:

`$g'\}^2$`  $g'^2$

В заключение напомним, что верхние и нижние индексы используются только в математической моде.

#### 4.4. Акценты в математике

Математики любят над буквами использовать акценты, потому что это часто бывает эффективным способом указать связь между математическими объектами и сильно расширяет количество доступных символов без увеличения количества необходимых шрифтов. В разделе 2.11 обсуждается, как использовать акценты в обычном тексте, но математические акценты — это особый случай, потому что здесь другая расстановка пробелов:  $\text{\TeX}$  для акцентов в формулах использует специальные соглашения, так что два вида акцентов не следует путать друг с другом.  $\mathcal{AMS}\text{-}\text{\TeX}$ 'ом предусмотрены следующие математические акценты:

`$\hat a$`  $\hat{a}$   
`$\check a$`  $\check{a}$   
`$\tilde a$`  $\tilde{a}$   
`$\acute a$`  $\acute{a}$   
`$\grave a$`  $\grave{a}$   
`$\dot a$`  $\dot{a}$   
`$\ddot a$`  $\ddot{a}$   
`$\dddots a$`  $\ddots$   
`$\breve a$`  $\breve{a}$   
`$\bar a$`  $\bar{a}$   
`$\vec a$`  $\vec{a}$

Как и в тексте, в математической моде, когда акцент располагается над  $i$  или  $j$ , следует использовать форму “без точек”  $\hat{i}$  и  $\check{j}$ , которые получаются, соответственно, командами `\imath` и `\jmath`:

<code>\$\$\hat{\imath}</code>	$\hat{i}$
<code>\$\$\check{\jmath}</code>	$\check{j}$

Любовь математиков к символам с акцентами не знает границ. Иногда им необходимы даже двойные акценты. Если использовать обычные акценты

<code>\$\$\hat{\hat{A}}</code>	$\hat{\hat{A}}$
--------------------------------	-----------------

то символы акцентов печатаются не в точности друг над другом, что не очень красиво. Поэтому у  $\mathcal{A}\mathcal{M}\mathcal{S}\text{-}\mathcal{T}\mathcal{E}\mathcal{X}$  для команды `\hat` имеется альтернатива `\Hat`, которая хотя и усложняет работу  $\mathcal{T}\mathcal{E}\mathcal{X}$ , но акценты располагает как следует:

<code>\$\$\Hat{\Hat{A}}</code>	$\hat{\hat{A}}$
--------------------------------	-----------------

Для команд получения других акцентов также имеются альтернативные команды `\Check`, `\Tilde`, `\Acute`, `\Grave`, `\Dot`, `\Ddot`, `\Breve`, `\Bar` и `\Vec`.

На первый взгляд кажется излишним иметь такие пары команд, потому что, команды, начинающиеся с прописной буквы вроде бы вполне заменяют свои “строчные” пары. Но  $\mathcal{A}\mathcal{M}\mathcal{S}\text{-}\mathcal{T}\mathcal{E}\mathcal{X}$  предоставляет и `\hat`, и `\Hat`, потому что `\Hat` очень неэкономична и пожирает компьютерное время, и для единичного акцента разумно использовать только `\hat`.

*Широкие акценты.* Для двух видов акцентов  $\mathcal{A}\mathcal{M}\mathcal{S}\text{-}\mathcal{T}\mathcal{E}\mathcal{X}$  имеет более широкие варианты, получаемые командами `\widehat` и `\widetilde`. Эти команды дают акцент переменной величины, в зависимости от того, над чем он расположен:

<code>\$\$\widehat{x}</code> , <code>\$\$\widetilde{x}</code>	$\widehat{x}$ , $\widetilde{x}$
<code>\$\$\widehat{xy}</code> , <code>\$\$\widetilde{xy}</code>	$\widehat{xy}$ , $\widetilde{xy}$
<code>\$\$\widehat{xyz}</code> , <code>\$\$\widetilde{xyz}</code>	$\widehat{xyz}$ , $\widetilde{xyz}$
<code>\$\$\widehat{xyzu}</code> , <code>\$\$\widetilde{xyzu}</code>	$\widehat{xyzu}$ , $\widetilde{xyzu}$
<code>\$\$\widehat{xyzuv}</code> , <code>\$\$\widetilde{xyzuv}</code>	$\widehat{xyzuv}$ , $\widetilde{xyzuv}$

Эти более широкие акценты находятся в шрифтах семейства `msbm`. Если `msbm` загружен, то команды `\widehat` и `\widetilde` при необходимости будут автоматически выбирать более широкие варианты; в противном случае самым широким вариантом будет символ со строки 3. Если вы используете стиль `\amsrpt`, `msbm` загружается автоматически.

#### 4.5. Черта, стрелка или скобка над или под формулой

Команды `\underline` и `\overline` позволяют провести черту нужной длины под или над формулой. Они автоматически выбирают правильный размер черты в зависимости от контекста:

<code>\$\$\underline{4}</code>	$\underline{4}$
<code>\$\$\underline{\underline{4+x}}</code>	$\underline{\underline{4+x}}$
<code>\$\$x^{\underline{m+n}}</code>	$x^{\underline{m+n}}$
<code>\$\$\overline{\overline{x^3}+x^{x^3}}</code>	$\overline{\overline{x^3}+x^{x^3}}$

То же самое относится и к стрелкам:

<code>\$\$\overrightarrow{x+y}</code>	$\overrightarrow{x+y}$
<code>\$\$\overleftarrow{x-y}</code>	$\overleftarrow{x-y}$
<code>\$\$A^{\overleftarrow{\overrightarrow{x+y}}}</code>	$A^{\overleftarrow{\overrightarrow{x+y}}}$

Стрелки под формулами можно получить управляющими последовательностями `\underrightarrow`, `\underleftarrow` и `\underleftrightharrow`. Самые распространенные стрелки, `\overrightarrow` и `\underrightarrow`, имеют также краткие имена `\overarrow` и `\underarrow`.

При помощи `\overbrace` и `\underbrace` можно над и под формулами рисовать горизонтальные скобки.

$$\begin{array}{l} \overbrace{x+\dots+x} \\ \underbrace{x+y+z} \end{array}$$

Можно поместить над `\overbrace` или под `\underbrace` какие-либо еще формулы или текст, если их набрать просто как верхний и нижний индексы, как если бы вы имели дело с большими операторами (см. **Текст в формулах** и **Большие операторы**):

$$\begin{array}{l} \overbrace{x+y+z}^{\text{\textit{k times}}} \\ \underbrace{x+y+z}_{>0} \end{array}$$

(Во втором примере был вставлен тонкий пробел, поскольку `\TeX` в индексах автоматически не оставляет пробелов вокруг бинарных операторов.)

#### 4.6. Дроби и биномиальные коэффициенты

Для набора дробей `\mathcal{A}\mathcal{M}\mathcal{S}\text{-}\TeX` имеет ряд команд. Самая важная из них — `\frac`. `\frac` — это команда с двумя аргументами: числителем над чертой дроби и знаменателем под ней

$$\frac{n+1}{n+3}$$

Дроби в выключенной математической моде печатаются более крупно, чем в обычной математической моде. Например, если мы введем `\frac{n+1}{n+3}`, то получим  $\frac{n+1}{n+3}$ . Приведенная выше выключная формула имеет так называемый размер *d-size* (от *displaysize*), а ее числитель и знаменатель — обычный размер *t-size* (от *textsize*). Во втором же случае (когда формула включена в текст абзаца), вся дробь имеет размер *t-size*, а числитель и знаменатель имеют меньший размер *s-size* (от *scriptsize*).

Полученные посредством `\frac` дроби автоматически правильно располагаются относительно бинарных операций и отношений:

$$z = \frac{x+y^2}{x-y^2} - 1$$

Фигурные скобки иногда можно опустить:

$$\begin{array}{l} \frac{2}{3} \\ \frac{1}{n+1} \\ \frac{N-1}{2} \end{array}$$

Поскольку, как уже говорилось, `\frac` — это команда с двумя аргументами, она даже в первом из этих случаев будет считать 2 своим первым аргументом, т. е. числителем, а 3 — вторым, т. е. знаменателем. Фигурные скобки же нужны, когда числителем или знаменателем является подформула.

Если в дроби в свою очередь содержится дробь, то это получается так:

$$\begin{array}{l} \frac{x}{1+\frac{x}{2}} \\ \frac{\frac{x}{2}+1}{2} \end{array}$$

В обоих случаях, по-видимому, лучше было бы изобразить дробь  $\frac{x}{2}$  через “косую черту”  $x/2$ :

$$\begin{array}{l} \frac{x}{1+x/2} \\ \frac{x/2+1}{2} \end{array}$$

*Изменение размера дроби.* Если вы настаиваете на том, чтобы формула в тексте абзаца печаталась в виде  $\frac{x}{2}$  (размера d-size), то у  $\mathcal{A}\mathcal{M}\mathcal{S}\text{-}\mathcal{T}\mathcal{E}\mathcal{X}$ 'а есть команда `\dspace`, вынуждающая набирать формулу в размере d-size:

$$\begin{array}{ll} \text{\$}\frac{x}{2}\text{\$} & \frac{x}{2} + \frac{x}{2} \\ \text{\$}\dspace\frac{x}{2}\text{\$} & \frac{x}{2} + \frac{x}{2} \end{array}$$

Команда `\dspace` вызывает переключение на d-size всей формулы, и его действия ограничены этой формулой.

На самом деле  $\mathcal{A}\mathcal{M}\mathcal{S}\text{-}\mathcal{T}\mathcal{E}\mathcal{X}$  располагает еще лучшим способом получения дроби размера d-size. Управляющая последовательность `\dfraction` автоматически даст размер d-size; таким образом, набор `\dfraction ab` эквивалентен набору `{\dspace\frac{ab}`.  $\mathcal{A}\mathcal{M}\mathcal{S}\text{-}\mathcal{T}\mathcal{E}\mathcal{X}$  также имеет управляющую последовательность `\tsize` для получения формул размера t-size. В свою очередь дроби размера t-size часто требуются в выключных формулах, так что у  $\mathcal{A}\mathcal{M}\mathcal{S}\text{-}\mathcal{T}\mathcal{E}\mathcal{X}$ 'а есть также и `\tfrac` для получения `\frac` размера t-size.  $\mathcal{A}\mathcal{M}\mathcal{S}\text{-}\mathcal{T}\mathcal{E}\mathcal{X}$ оснащен также `\ssize` и `\sssize` для превращения размера формулы в s-size или ss-size (размер повторного индекса). Когда дробь появляется в верхнем индексе, размер которого s-size, то числитель и знаменатель печатаются еще мельче, а именно, в размере ss-size:

$$\text{\$}e^{-n+\frac{1}{12n}}\text{\$}$$

*Изменение толщины дроби.* Можно варьировать толщину черты дроби. Для этого служит команда `\thickfrac`:

$$\text{\backslashthickfrac\thickness\{число\}}$$

Например, `\thickness2` делает черту дроби вдвое толще, `\thickness1.5` делает ее толще в 1.5 раза, и т. п.

*Дроби с ограничителями.* Если вам нужны ограничители вокруг дроби, можно вместо `\frac` использовать команду

$$\text{\backslashfracwithdelims\langle левый ограничитель \rangle \langle правый ограничитель \rangle}$$

Например, “символ Лежандра” можно получить так

$$\text{\$}\fracwithdelims()ab\text{\$}$$

Можно, конечно поместить вокруг `\frac ab` скобки с помощью конструкции `\left ... \right` (см. **Ограничители**), но использование `\fracwithdelims` предпочтительнее, поскольку в этом случае  $\mathcal{T}\mathcal{E}\mathcal{X}$  проставляет особые пробелы.

Имеется также команда `\thickfracwithdelims` для изменения толщины черты дроби с ограничителями. Например, введя

$$\text{\$}\thickfracwithdelims\langle \rangle\thickness0 nk\text{\$}$$

можно получить “число Эйлера”  $\langle \frac{n}{k} \rangle$  (в виде дроби с дробной чертой нулевой толщины, заключенной в угловые скобки).

*Цепные дроби.* В математике часто используются так называемые “цепные дроби”.  $\mathcal{A}\mathcal{M}\mathcal{S}\text{-}\mathcal{T}\mathcal{E}\mathcal{X}$  предоставляет простой способ для их набора. Так, цепная дробь

$$a_0 + \frac{1}{a_1 + \frac{1}{a_2 + \frac{1}{a_3 + \frac{1}{a_4}}}}$$

получается командами

$$\text{\$}\$a_0 + \cfrac{1\backslash\backslash$$

```

a_1 + \cfrac 1\
a_2 + \cfrac 1\
a_3 + \cfrac 1\
a_4\endcfrac$$

```

Каждый раз, как только вы начинаете новую “подробь”, набирайте `\cfrac` и используйте, как обычно, `\` чтобы отделить строки. Затем все заканчиваете единственным `\endcfrac`.

Некоторые математики предпочитают цепные дроби вида

$$a_0 + \frac{1}{a_1 + \frac{1}{a_2 + \frac{1}{a_3 + \frac{1}{a_4}}}}$$

где все числители, за исключением последнего, сдвинуты влево. Это было набрано так:

```

$$a_0 + \lcfrac1\
a_1 + \lcfrac 1\
a_2 + \lcfrac 1\
a_3 + \cfrac 1\
a_4\endcfrac$$

```

с подстановкой `\lcfrac` вместо `\cfrac` везде, где числитель должен быть сдвинут влево. Для числителей, сдвигаемых вправо, есть средство `\rcfrac`.

*Биномиальные коэффициенты.*

Кроме дробей, математики пользуются особым обозначением — “биномиальным коэффициентом”

```

$$\binom nk$$

```

Действие `\binom` несколько не отличается от действия `\frac`, соглашения относительно размеров верхней и нижней части те же самые:

```

$$\binom n{\frac k2}$$
$$\frac{\binom nk}2$$

```

$\mathcal{A}\mathcal{M}\mathcal{S}$ - $\text{T}\text{E}\text{X}$  имеет также `\dbinom` и `\tbinom` для `\binom` размера d-size или t-size.

#### 4.7. Большие операторы

Математики часто используют для обозначения “суммы” знак  $\sum$ , а для обозначения “интеграла” — знак  $\int$ . Если вы наборщик, а не математик, вам надо запомнить, что `\sum` дает  $\sum$ , а `\int` —  $\int$ . Символы типа  $\sum$  и  $\int$  (и несколько других символов типа  $\cup$ ,  $\prod$ ,  $\oint$  и  $\otimes$ ) называются *большими операторами* и вводятся почти так же, как обычные символы или буквы. Отличие в том, что  $\text{T}\text{E}\text{X}$  в выключном стиле выберет *большой* большой оператор (размер d-size), чем в текстовом стиле (размер t-size). Например,

```

 $\sum x_n$       дает  $\sum x_n$       (t-size)
 $\sum x_n$       дает  $\sum x_n$       (d-size).

```

Таблица имеющихся в  $\mathcal{A}\mathcal{M}\mathcal{S}$ - $\text{T}\text{E}\text{X}$ ’е больших операторов и команд для их получения приводится в разделе **Математические символы**.

*Большие операторы типа  $\sum$ .* Сумма в выключном стиле обычно бывает с “пределами”, т. е. с подформулами, которые появляются над и под ней. Пределы вводятся так же, как если бы это были верхние и нижние индексы. Например, если вы хотите получить формулу

$$\sum_{n=1}^m,$$

то введите либо  $\sum_{n=1}^m$ , либо  $\sum_{n=1}^m$ . Если же эту формулу ввести в тексте абзаца, то  $\text{T}\mathcal{E}\mathcal{X}$ , в соответствии со своими обычными правилами,  $\text{T}\mathcal{E}\mathcal{X}$  заменит ее на  $\sum_{n=1}^m$  (т. е. без пределов).

Иногда у больших операторов бывают многострочные пределы, как, например

$$\sum_{\substack{0 \leq i \leq m \\ 0 < j < n}} P(i, j)$$

Между  $\backslash\text{Sb}$  и  $\backslash\text{endSb}$  каждый  $\backslash\backslash$  свидетельствует о переходе на новую строку. Точно так же имеются  $\backslash\text{Sp}\dots\backslash\text{endSp}$  для получения многострочных верхних пределов.

*Большие операторы типа  $\int$ .* Интегралы слегка отличаются от суммы тем, что в них даже в выключном стиле верхние и нижние индексы не устанавливаются как пределы:

$$\int_{-\infty}^{+\infty} \quad \text{дает} \quad \int_{-\infty}^{+\infty} \quad (\text{t-size})$$

$$\int_{-\infty}^{+\infty} \quad \text{дает} \quad \int_{-\infty}^{+\infty} \quad (\text{d-size}).$$

Знак интеграла  $\int$  в выключной формуле будет большего размера, точно так же, как и в случае знака  $\sum$ , но индексы не станут “пределами” и не переместятся над и под интеграл.

Кроме  $\int$  в  $\mathcal{A}\mathcal{M}\mathcal{S}\text{-}\mathcal{T}\mathcal{E}\mathcal{X}$ е также имеется  $\oint$ , который выдает  $\oint$  и  $\oint$ . Знаки  $\int$  часто появляются группами, например  $\iiint$  и  $\iiiiiint$ . Для них существуют специальные символы

$$\iint$$

$$\iiint$$

$$\iiiiiint$$

$$\int \dots \int$$

*Расстановка пробелов.* Вопрос о расстановке “пределов” в больших операторах на самом деле решает не  $\text{T}\mathcal{E}\mathcal{X}$ , а стиль, который вы используете. Но если даже в данном журнале пределы у знаков  $\int$  обычно ставятся справа, для той или иной частной формулы можно вынудить поставить пределы для  $\int$  сверху и снизу. Например, в выражении

$$\int_{\partial(M - \bigcup_{i=1}^n U_i)} r^* \Pi = - \sum_{i=1}^n \int_{\partial U_i} r^* \Pi$$

длинный предел, поставленный под первым интегралом, выглядит лучше, чем

$$\int_{\partial(M - \bigcup_{i=1}^n U_i)} r^* \Pi.$$

Если вы набираете  $\int\limits$ , то  $\text{T}\mathcal{E}\mathcal{X}$  проставит все верхние и нижние индексы как “пределы”. Этот прием удобно использовать, если вы хотите, чтобы в тексте были формулы типа  $\sum_{n=1}^{\infty} \frac{(-1)^n}{n}$ , потому что задаваемая автоматически текстовая формула  $\sum_{n=1}^{\infty} \frac{(-1)^n}{n}$ , выглядит слегка приплюснута.

$\int\limits$  имеет свою противоположность  $\int\nolimits$ , который приводит к тому, что у больших операторов пределы остаются справа, даже если обычно они переставляются. Однако  $\int\limits$  и  $\int\nolimits$  должны использоваться только в особых случаях.

Хотя стиль  $\text{amsrpt}$  обычно использует “пределы” для  $\sum$  и некоторых других больших операторов, это соглашение можно изменить, набрав команду  $\backslash\text{NoLimitsOnSums}$ . (Если набрать  $\backslash\text{LimitsOnSums}$ , вернемся к старому соглашению.) Аналогично обстоит дело с  $\backslash\text{LimitsOnInts}$  и  $\backslash\text{NoLimitsOnInts}$  для  $\int$ ,  $\oint$ ,  $\iint$  и т. п. Эти управляющие последовательности “глобальны”, т. е. действуют на весь последующий текст, даже если были использованы в группе или между знаками  $\$$ .

#### 4.8. Элементарные функции типа log

Именами алгебраических переменных обычно являются курсивные или греческие буквы, но общепринятые математические функции типа “log” всегда печатаются прямым шрифтом. Такие функции задаются специальными командами, которые не только устанавливают для названия функции прямой шрифт, но и обеспечивают соответствующие пробелы. Иногда такие функции называют *операторами* (не путать с *большими операторами*). Например

$$\begin{array}{ll} \text{\$}\sin 2\theta=2\sin\theta\cos\theta\text{\$} & \sin 2\theta = 2 \sin \theta \cos \theta \\ \text{\$}O(n\log n\log\log n)\text{\$} & O(n \log n \log \log n) \\ \text{\$\Pr}(X>x)=\exp(-x/\mu)\text{\$} & \Pr(X > x) = \exp(-x/\mu) \\ \text{\$\$\max_{1\le n\le m}\log_2 P_n\$\$} & \max_{1 \leq n \leq m} \log_2 P_n \\ \text{\$\$\lim_{x\to 0}\{\sin x\over x\}=1\$\$} & \lim_{x \rightarrow 0} \frac{\sin x}{x} = 1 \end{array}$$

Последние две формулы, которые являются выключными, показывают, что  $\text{\TeX}$  обрабатывает некоторые операторы как “большие операторы” с пределами (типа  $\sum$ ). Нижний индекс в  $\text{\max}$  трактуется не так, как нижний индекс в  $\text{\log}$ . Приведем список известных  $\mathcal{A}\mathcal{M}\mathcal{S}\text{-}\text{\TeX}$  операторов; те, для которых верхние и нижние индексы устанавливаются как “пределы”, помечены буквой (L):

$\text{\arcoss}$	$\text{\cot}$	$\text{\exp}$	(L)	$\text{\lim}$	$\text{\sec}$
$\text{\arcsin}$	$\text{\coth}$	(L) $\text{\gcd}$		$\text{\ln}$	$\text{\sin}$
$\text{\arctan}$	$\text{\csc}$	$\text{\hom}$		$\text{\log}$	$\text{\sinh}$
$\text{\arg}$	$\text{\deg}$	(L) $\text{\inf}$	(L)	$\text{\max}$	(L) $\text{\sup}$
$\text{\cos}$	(L) $\text{\det}$	$\text{\ker}$	(L)	$\text{\min}$	$\text{\tan}$
$\text{\cosh}$	$\text{\dim}$	$\text{\lg}$	(L)	$\text{\Pr}$	$\text{\tanh}$

Все эти команды дают операторы, соответствующие своим названиям.  $\mathcal{A}\mathcal{M}\mathcal{S}\text{-}\text{\TeX}$  также имеет  $\text{\liminf}$ ,  $\text{\limsup}$ ,  $\text{\injlim}$  и  $\text{\projlim}$ , которые дают, соответственно, “lim inf”, “lim sup”, “inj lim” и “proj lim”. Некоторые математики используют и другие обозначения:

$$\begin{array}{ll} \text{\$\varliminf\$\} & \underline{\underline{\lim}} \\ \text{\$\varlimsup\$\} & \overline{\overline{\lim}} \\ \text{\$\varinjlim\$\} & \underline{\lim} \\ \text{\$\varprojlim\$\} & \overline{\lim} \end{array}$$

Несмотря на такой впечатляющий список имеющихся операторов, математики продолжают изобретать все новые и новые. Автор может сам создавать новые операторы. Например, команда

$\text{\operatorname\{Tor\}}$

в любой формуле будет печатать Tor, как оператор (прямым шрифтом и с соответствующими пробелами). Если же нужен оператор с пределами, используется команда  $\text{\operatornamewithlimits}$ . Так, если ввести

$$\text{\$\$\operatornamewithlimits\{Res\}_{x=0}\frac{f(x)}{x}\$\$},$$

то получится

$$\text{Res}_{x=0} \frac{f(x)}{x}.$$

Разумеется, если эти обозначения в работе встречаются часто, немисливо каждый раз вводить такую длинную конструкцию, а лучше определить новую команду.

*Модуль (mod)*. Так же, как и названия перечисленных выше элементарных функций, слово ‘mod’ обычно печатается в формулах романским (прямым) шрифтом, но по смыслу это обозначение отличается от элементарных функций. В математике ‘mod’ может использоваться в двух разных смыслах: если это бинарный оператор, стоящий между двумя величинами, то употребляется  $\text{\bmod}$ , а в конце формул в скобках (так называемое обозначение “по модулю”) используется  $\text{\pmod}$ :

$$\text{\$\gcd(m,n)=gcd(n,m\bmod n)\$} \qquad \text{gcd}(m,n) = \text{gcd}(n, m \bmod n)$$

$$\text{\$x\equiv y+1\pmod{m^2}\$} \qquad x \equiv y + 1 \pmod{m^2}$$

Обратите внимание, что скобки вокруг `\pmod` проставляются автоматически.

Иногда в выражениях “по модулю” `mod` ставится без скобок. В этом случае надо использовать команду `\mod`:

$$\text{\$x\equiv y+1\mod{m^2}\$} \qquad x \equiv y + 1 \pmod{m^2}$$

Некоторые авторы “по модулю” обозначают так: вовсе не ставят `mod`, зато  $m^2$  помещают с скобках. Для таких случаев есть команда `\pod`:

$$\text{\$x\equiv y+1\pod{m^2}\$} \qquad x \equiv y + 1 \pmod{m^2}$$

#### 4.9. Корни

*Квадратные корни.* Знак квадратного корня получается командой `\sqrt`. Если под квадратным корнем должна быть подформула, а не простой символ, то эту подформулу следует заключить в фигурные скобки.

$$\begin{aligned} \text{\$\$\sqrt{2}\$\$} & \qquad \sqrt{2} \\ \text{\$\$\sqrt{\frac{ab+1}{b}}\$\$} & \qquad \sqrt{\frac{a}{b} + 1} \end{aligned}$$

Знаки квадратного корня печатаются в различных видах, в зависимости от высоты, глубины и ширины выражения, из которого извлекается квадратный корень:  $\sqrt{a}$ ,  $\sqrt{d}$  и  $\sqrt{y}$ . Если у вас есть формула, в которой только один `\sqrt`, правила позиционирования работают прекрасно. Но вот вместо формулы  $\sqrt{a} + \sqrt{d} + \sqrt{y}$  вам бы хотелось иметь более симметричное выражение  $\sqrt{a + \sqrt{d} + \sqrt{y}}$ . Это можно получить посредством

$$\text{\$\sqrt{\mathstrut a}+\sqrt{\mathstrut d}+\sqrt{\mathstrut y}\$\$}$$

Команда `\mathstrut` дает невидимый символ, который простирается над и под строкой на расстояние, достаточное для того, чтобы перекрыть любую букву.

Поскольку математические формулы могут получаться ужасающе большими,  $\mathcal{T}\mathcal{E}\mathcal{X}$  должен иметь какой-нибудь способ создавать все более увеличивающиеся квадратные корни. Например, если вы вводите

$$\text{\$\$\sqrt{1+\sqrt{1+\sqrt{1+\sqrt{1+\sqrt{1+\sqrt{1+x}}}}}}}\$\$}$$

то результат показывает ряд имеющихся в наличии знаков квадратного корня:

$$\sqrt{1 + \sqrt{1 + \sqrt{1 + \sqrt{1 + \sqrt{1 + \sqrt{1 + \sqrt{1 + x}}}}}}}$$

Три наибольших знака здесь, по существу, одинаковы, за исключением того, что вертикальный сегмент “|” повторяется столько, сколько необходимо, чтобы получить желаемый размер, но более маленькие знаки — это отдельные символы в математических шрифтах  $\mathcal{T}\mathcal{E}\mathcal{X}$ 'а.

*Корни с другими показателями степени.* Чтобы получить корни, степень которых отлична от 2, надо использовать конструкцию `\root ... \of ...` :

$$\begin{aligned} \text{\$\$\root3\of x\$\$} & \qquad \sqrt[3]{x} \\ \text{\$\$\root\alpha+\beta\of{1+\frac{ab}{b}}\$\$} & \qquad \alpha+\beta \sqrt[1 + \frac{a}{b}]{} \end{aligned}$$

Если показателем степени корня является подформула, то ее можно не заключать в фигурные скобки, поскольку она ограничивается `\root` и `\of`.  $\mathcal{A}\mathcal{M}\mathcal{S}\text{-}\mathcal{T}\mathcal{E}\mathcal{X}$  стремится расположить корень правильно, но если вы хотите несколько подправить его расположение, то воспользуйтесь `\uproot{⟨число⟩}` после `\root`, чтобы передвинуть корень вверх на ⟨число⟩ единиц, а чтобы сдвинуть его влево на

$\langle \text{число} \rangle$  единиц, надо набрать  $\backslash\leftrootof{\langle \text{число} \rangle}$  после него. Можно одновременно использовать управляющие последовательности  $\backslash\uprootof{\langle \text{число} \rangle}$  и  $\backslash\leftrootof{\langle \text{число} \rangle}$  (в произвольном порядке), пока ничего иного между ними и  $\backslash\root$  не находится. Например, если ввести

$$\backslash\root\uproot 3\leftrootof{-2}\alpha+\beta\of{1+\frac{ab}}{\quad}}$$

то получится

$$\sqrt[\alpha+\beta]{1+\frac{a}{b}}$$

Единицы, в которых выражается  $\langle \text{число} \rangle$  и на которые перемещается  $\backslash\root$ , чрезвычайно мелкие, так что методом проб и ошибок можно добиться желаемого результата.

#### 4.10. Ограничители

*Основные ограничители.* Как уже говорилось выше при описании символов клавиатуры, некоторые символы математики считают *открывающими и закрывающими ограничителями*. В математике важны ограничители, поскольку они хорошо визуально подчеркивают структуры в сложных выражениях — ограничивают отдельные подформулы. Приведем список 22 основных ограничителей, предусмотренных в TeX'e.

<i>Вход</i>	<i>Ограничитель</i>
(	левая круглая скобка: (
)	правая круглая скобка: )
[ или $\backslash\lbrack$	левая квадратная скобка: [
] или $\backslashrbrack$	правая квадратная скобка: ]
\{ или $\backslashlbrace$	левая фигурная скобка: {
\} или $\backslashrbrace$	правая фигурная скобка: }
\lfloor	левая “кочерга вниз”: [
\rfloor	правая “кочерга вниз”: ]
\lceil	левая “кочерга вверх”: [
\rceil	правая “кочерга вверх”: ]
\langle	левая угловая скобка: <
\rangle	правая угловая скобка: >
/	слэш: /
\backslash	бэкслэш: \
или $\backslash\vert$	вертикальная черта:
\  или $\backslash\Vert$	двойная вертикальная черта:
\uparrow	стрелка вверх: ↑
\Uparrow	двойная стрелка вверх: ↑↑
\downarrow	стрелка вниз: ↓
\Downarrow	двойная стрелка вниз: ↓↓
\updownarrow	двусторонняя стрелка: ⇕
\Updownarrow	двойная двусторонняя стрелка: ⇕⇕

Иногда есть два способа получить один и тот же ограничитель. Например, можно задать левую квадратную скобку, введя либо [, либо  $\backslash\lbrack$ . Последнее предпочтительнее, поскольку [ имеется не на всех клавиатурах. Помните, однако, что вы никогда не должны пытаться задавать левую и правую фигурные скобки просто как { или }. Символы { и } зарезервированы для группирования. Правильным будет ввести \{, \}, \lbrace, \rbrace.

Можно вводить < или > как условное сокращение для  $\backslash\langle$  и  $\backslash\rangle$ . Например,  $\backslash\bigl\langle$  эквивалентно  $\backslash\bigl\langle\langle$ , а  $\backslash\right\rangle$  эквивалентно  $\backslash\right\rangle\rangle$ . Конечно, ‘<’ и ‘>’ обычно производят отношения “меньше чем” и “больше чем” <>, которые заметно отличаются от угловых скобок.

*Увеличенные ограничители.* Для того, чтобы получить несколько увеличенную версию любого из этих символов, просто поставьте перед ним  $\backslash\bigl$  (для открывающего ограничителя) или  $\backslash\bigl$

(для закрывающего ограничителя). Это облегчит чтение формул, которые содержат ограничители внутри ограничителей:

<i>Вход</i>	<i>Выход</i>
<code>\bigl(x-s(x)\bigr)\bigl(y-s(y)\bigr)</code>	$(x - s(x))(y - s(y))$
<code>\bigl[x-s[x]\bigr]\bigl[y-s[y]\bigr]</code>	$[x - s[x]][y - s[y]]$
<code>\bigl\vert\vert x\vert-\vert y\vert \bigr\vert</code>	$  x  -  y  $
<code>\bigl\lfloor\sqrt A\bigr\rfloor</code>	$[\sqrt A]$

Ограничители `\big` настолько больше обычных, что можно почувствовать различие, но еще достаточно малы, так что их можно использовать в тексте абзаца. Приведем все 22 ограничителя в обычном размере и в размере `\big`:

`() [] {} [] [] \langle \rangle / \backslash ||| \uparrow \uparrow \downarrow \downarrow \updownarrow \updownarrow`

`() [] {} [] [] \langle \rangle / \backslash ||| \uparrow \uparrow \downarrow \downarrow \updownarrow \updownarrow`

Можно также использовать `\Bigl` и `\Bigr` для символов в выключенных формулах:

`() [] {} [] [] \langle \rangle / \backslash ||| \uparrow \uparrow \downarrow \downarrow \updownarrow \updownarrow`

Они на 50% выше их `\big`-двойников. Выключенные формулы чаще используют ограничители, которые еще выше (в два раза выше, чем `\big`). Такие ограничители создаются при помощи `\biggl` и `\biggr` и выглядят так:

`() [] {} [] [] \langle \rangle / \backslash ||| \uparrow \uparrow \downarrow \downarrow \updownarrow \updownarrow`

Наконец, есть версия `\Biggl` и `\Biggr`, которая в 2.5 раза выше ограничителей `\bigl` и `\bigr`:

`() [] {} [] [] \langle \rangle / \backslash ||| \uparrow \uparrow \downarrow \downarrow \updownarrow \updownarrow`

Например, чтобы напечатать выключенную формулу

$$\left(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}\right)|\varphi(x + iy)|^2 = 0$$

можно ввести

```

\biggl(\frac{\partial^2}{\partial x^2} +
\frac{\partial^2}{\partial y^2}\biggr)\bigl\vert\varphi(x+iy)
\bigl\vert^2=0

```

Ограничители `\bigl`, `\Bigl`, `\biggl` и `\Biggl` являются открывающими, как левая круглая скобка, а ограничители `\bigr`, `\Bigr`, `\biggr`, `\Biggr` — закрывающими, как правая круглая скобка.  $\text{T}\mathcal{E}\mathcal{X}$  также предусматривает ограничители `\bigm`, `\Bigm`, `\biggm` и `\Biggm` для использования в середине формул. Такой ограничитель играет роль отношения, типа знака равенства, поэтому  $\text{T}\mathcal{E}\mathcal{X}$  помещает с обеих сторон от него маленькие пробелы. Например:

$$\bigl(x \in A(n) \bigr) \bigm x \in B(n) \bigr) \quad (x \in A(n) \mid x \in B(n))$$

$$\bigcup_n X_n \bigm \bigcap_n Y_n$$

Можно также сказать просто `\big`, `\Big`, `\bigg`, `\Bigg`, что дает ограничитель, который действует как обычная переменная. Это используется преимущественно при наклонных чертах и обратных наклонных чертах, как показано в следующем примере:

$$\frac{a+1}{b} \bigg/ \frac{c+1}{d}$$

*Автоматическая установка размера ограничителей.* Т<sub>Э</sub>X имеет встроенный механизм, который вычисляет, насколько высокой должна быть пара ограничителей для того, чтобы охватить данную подформулу, поэтому можно использовать этот метод вместо того, чтобы решать, должен ли быть ограничитель `\big`, `\bigg` или какой-нибудь еще. Единственное, что надо сделать — это сказать

$$\left\langle \text{ограничитель}_1 \right\rangle \langle \text{подформула} \rangle \right\langle \text{ограничитель}_2 \rangle$$

и Т<sub>Э</sub>X напечатает подформулу, вставляя слева и справа заданные ограничители. Размер ограничителей будет как раз такой величины, чтобы охватить подформулу. Например, в выключенной подформуле

$$1 + \left( \frac{1}{1 - x^2} \right)^3$$

Т<sub>Э</sub>X выбрал `\biggl` (и `\biggr`), поскольку меньшие ограничители для этой дроби слишком малы. Простая формула типа `\left(x\right)` дает  $(x)$ , таким образом, `\left` и `\right` иногда выбирают ограничители, которые меньше, чем `\bigl` и `\bigr`.

Операция `\over` в примере выше не включает в себя “+” в начале формулы. Это получилось потому, что `\left` и `\right` в дополнение к функции создания ограничителя выполняют функцию *группирования*: любые определения, которые оказываются между `\left` и `\right`, будут локальными, как если бы заключенная в них подформула была в фигурных скобках.

Всякий раз, когда вы используете `\left` и `\right`, они должны быть в паре друг с другом, как и фигурные скобки в группах. Не может быть `\left` в одной формуле, а `\right` в другой, а также не позволяют конструкции типа

$$\left( \dots \{ \dots \right) \dots \}.$$

Эти ограничения понятны, поскольку Т<sub>Э</sub>X’у прежде чем он может решить, насколько большими делать ограничители, надо набрать подформулу между `\left` и `\right`. Но это стоит ясно упомянуть, потому что вы *не обязаны* употреблять парами круглые, квадратные и тому подобные скобки, если не используете `\left` и `\right`: Т<sub>Э</sub>X не будет возражать против формулы `$(0,1)$`, `$( )` (`$` или даже `)$`). Даже когда используется `\left` и `\right`, Т<sub>Э</sub>X не рассматривает подробно, какие конкретные ограничители выбраны. Так, можно ввести такую странную штуку как “`\left`”, что, в отличие от просто `)`, даст *левый* ограничитель, и/или “`\right`”, что даст *правый* ограничитель. Это можно использовать в интересном примере. Сравните две формулы:

$$\begin{aligned} x \in ]\frac{a}{b}, \frac{c}{d}[ & \quad x \in ]\frac{a}{b}, \frac{c}{d}[ \\ x \in \left] \frac{a}{b}, \frac{c}{d} \right[ & \quad x \in \left] \frac{a}{b}, \frac{c}{d} \right[ \end{aligned}$$

Т<sub>Э</sub>X не оставляет дополнительные пробелы между бинарным отношением и правым ограничителем, и в первом случае после `∈` получился слишком маленький пробел, поскольку Т<sub>Э</sub>X ошибочно посчитал `] правый` ограничителем. Правильным решением будет вторая формула.

В описанном выше примере мы видели еще одну функцию конструкции `\left ... \right` — функцию “распределителя пробелов”. Покажем это еще на одном примере. Ограничители `|` и `\|` довольно специальные, поскольку один и тот же символ служит и левым, и правым ограничителем. Когда перед этими ограничителями не стоит `\left` или `\right` Т<sub>Э</sub>X не может понять, в каком смысле они используются, поэтому считает их ординарными символами со всеми вытекающими отсюда последствиями. Это не вызывает протеста в формулах типа `|x|`, но если в формулу входят бинарные операторы, то даже для получения ограничителей обычного размера могут потребоваться `\left` и `\right`, которые дают правильное распределение пробелов. Сравните, например, две формулы:

$$\begin{aligned} |-x| = |x| & \quad | -x | = | +x | \\ \left| -x \right| = \left| +x \right| & \quad | -x | = | +x | \end{aligned}$$

Сейчас вы, вероятно, удивляетесь, зачем надо тратить силы, изучая `\bigl`, `\bigr` и им подобные, когда `\left` и `\right` должны автоматически вычислять размеры. Да, это правда, `\left` и `\right` достаточно удобны, но есть как минимум три ситуации, когда для выбора размеров ограничителей вам придется воспользоваться собственной мудростью.

- (1) Иногда `\left` и `\right` выбирают размеры меньше, чем вы хотите.

- (2) Иногда `\left` и `\right` выбирают размеры больше, чем вы хотите. Это наиболее часто случается, когда они ограничивают большой оператор в выключной формуле.
- (3) Иногда надо разбить огромную выключенную формулу на две или более отдельные строки, и вы хотите быть уверенными, что открывающиеся и закрывающиеся ограничители имеют одинаковую величину. Но вы не можете использовать на первой строке `\left`, а на последней — `\right`, поскольку `\left` и `\right` должны встречаться парами. Решением будет использовать `\Biggl`, скажем, на первой строке и `\Biggr` — на последней.

Конечно, одним из преимуществ `\left` и `\right` является то, что они могут сделать произвольно большие ограничители — намного больше, чем `\bigggggg!` Однако слэши и угловые скобки имеют максимальный размер. Если вы потребуете более крупные варианты этих символов, то получите наибольший из возможных.

*“Нулевой” ограничитель.* Если ввести ‘.’ после `\left` или `\right` вместо одного из основных ограничителей, то получится так называемый нулевой ограничитель (который равен пробелу). Это может понадобиться, когда нужны формулы, которые содержат только один большой ограничитель, например, выключенная формула

$$\left. \frac{dx^2}{dx} \right|_{x=a} = 2a$$

может быть получена конструкцией вида

`$$\left.\frac{dx^2}{dx}\right|_{x=a}=2a$$`

Здесь `\left.` даст возможность получить невидимый левый ограничитель, соответствующий правой скобке `\right|`.

*Дополнительные ограничители.*  $\TeX$  имеет и дополнительные ограничители, которые не были перечислены в основном наборе 22 ограничителей, потому что они особого сорта. Управляющие последовательности `\arrowvert`, `\Arrowvert` и `\bracevert` производят ограничители, сделанные из повторяющихся частей вертикальных стрелок, двойных вертикальных стрелок и больших фигурных скобок, соответственно, без верхушки стрелок и без закрученной части фигурных скобок. Результат аналогичен `\vert` или `\Vert`, но у них большие пробелы и другая ширина. Также можно использовать `\lgroup` и `\rgroup`, которые сконструированы из фигурных скобок без их средней части, и `\lmoustache` и `\rmoustache`, которые дают верхние и нижние половины фигурных скобок. Например, приведем `\Big` и `\bigg` версии от `\vert`, `\Vert` и этих семи специальных ограничителей:

$$\begin{array}{c} \dots \left| \dots \right| \dots \left| \dots \right| \dots \left| \dots \left( \dots \right) \dots \left. \dots \right. \dots ; \\ \dots \left| \dots \right| \dots \left| \dots \right| \dots \left| \dots \left( \dots \right) \dots \left. \dots \right. \dots \end{array}$$

Заметим, что `\lgroup` и `\rgroup` довольно похожи на жирные круглые скобки с острыми изгибами по углам. Это делает их заманчивыми для некоторых больших выключенных формул. Но их нельзя использовать точно так же, как круглые скобки, потому что они доступны только в больших размерах (`\Big` и больше).

*Теоретико-множественные обозначения.* Особо следует отметить применение ограничителей в теоретико-множественных обозначениях. Простые формулы типа  $\{a, b, c\}$  набираются очень просто:

<code>\$\$\{a,b,c\}\$\$</code>	$\{a, b, c\}$
<code>\$\$\{1,2,\dots,n\}\$\$</code>	$\{1, 2, \dots, n\}$

Но когда посреди такой формулы встречается вертикальная черта `|` или двоеточие `:`, то для получения черты лучше использовать команду `\mid`, которая проставляет вокруг `|` дополнительные пробелы, а фигурные скобки отделить тонкими пробелами (вокруг двоеточия пробелы устанавливаются автоматически):

<code>\$\$\{\,z\mid z&gt;2\,\}\$\$</code>	$\{ z \mid z > 2 \}$
<code>\$\$\{\,z:z&gt;2\,\}\$\$</code>	$\{ z : z > 2 \}$

Когда элементы множества заключаются в увеличенные ограничители `\bigl` и `\bigr`, как в случае

$$\{ (x, f(x)) \mid x \in D \},$$

то для получения `|` следует использовать увеличенную версию `\bigm|`.

`$$\bigl\lbrace\,,\bigl(x,f(x)\bigr)\bigm|x\in D\,,\bigr\rbrace,$$`

#### 4.11. Составные символы

Математики любят создавать новые сложные символы, помещая нечто над или под символом. Для этого у  $\mathcal{A}\mathcal{M}\mathcal{S}\text{-}\mathcal{T}\mathcal{E}\mathcal{X}$  есть команды `\overset` и `\underset`:

<code>\$\$\underset X\to A\$</code>	$\underset{X}{A}$
<code>\$\$\underset\alpha\beta\to X\$</code>	$\underset{\alpha\beta}{X}$
<code>\$\$\overset\alpha\beta\to\longrightarrow\$</code>	$\overset{\alpha\beta}{\longrightarrow}$
<code>\$\$\overset\text{def}\to=\$</code>	$\overset{\text{def}}{=}$
<code>\$\$\overset s\to\{\underset A\to X\}\$</code>	$\overset{s}{X}_A$

Обратите внимание, что скобки вокруг двойной надписи `\alpha\beta` не нужны, потому что надпись всегда ограничена командами `\underset` и `\to`. Разумеется, в конструкции `\underset ... \to` элемент `\to` — это лишь часть “синтаксиса”, а не правая стрелка  $\rightarrow$ , которая тоже имеет имя `\to`.

`\underset` и `\overset` не так тщательно располагают надписи, как это имеет место в случае математических акцентов. Иногда взаимное расположение символов надо скорректировать с помощью дополнительных пробельчиков. К счастью, эти конструкции используются не так уж часто, и, немного поэкспериментировав, вы можете получить приемлемый результат. Сравните, например, последнюю строку в приведенных выше примерах и

`$$\overset \,,\, s\to\{\underset A\to X\}$`  $\overset{s}{X}_A$

Когда вы применяете `\overset` или `\underset` к бинарной операции или отношению, в результате получается бинарная операция или отношение.

Когда вы применяете `\overset` или `\underset` к ординарному символу, то любые верхние или нижние индексы будут располагаться на правильной высоте:

`$$\overset \,,\alpha\to X_i^j$`  $\overset{\alpha}{X}_i^j$

Верхние и нижние индексы выставляются на высоте, соответствующей основной литере  $X$ , а не всей конструкции  $X$ . Но если вы используете `\overset` и `\underset` для получения новой бинарной операции, то это уже так хорошо не работает:

`$$\overset+\to=_j$`  $\overset{+}{=}^j$

Вы можете улучшить результат, набрав

`$$\overset+\to=\{}_j$`  $\overset{+}{=}^j$

но тогда после символа  $\overset{+}{=}$  появится дополнительный пробел.

Хотя для получения

$$\overbrace{x + \dots + x}^{k \text{ раз}} \quad \text{и} \quad \underbrace{x + y + z}_{> 0}$$

нет необходимости использовать команды `\overset` и `\underset`, поскольку для этой цели имеются команды `\overbrace` и `\underbrace` (см. в этом руководстве раздел **Черта, стрелка или скобка над или под формулой**),  $\mathcal{A}\mathcal{M}\mathcal{S}\text{-}\mathcal{T}\mathcal{E}\mathcal{X}$  все-таки предоставляет для этих целей конструкции `\undersetbrace ... \to` и `\oversetbrace ... \to`. Так, например, если ввести



`\text` — это управляющая последовательность с аргументом в виде текста. То, что следует после аргумента, опять будет представлено в математической моде. Например, выключная формула

$$f(x) = x^{17} + \text{члены низшего порядка} + e^x$$

была получена следующей конструкцией:

```
$$f(x)=x^17+\text{члены низшего порядка}
+e^x$$
```

Внутри `\text`, как и в обычном тексте, можно менять шрифты. Так, формула

$$f(x) = x^{17} + \text{члены } \textit{другого} \text{ порядка.}$$

получается командами

```
$$f(x)=x^17+\text{члены {\it другого}порядка.}$$
```

В тексте, получаемом командой `\text` можно использовать сноски. Правда, обычная команда для получения сносок `\footnote` внутри математической моды не работает (она там просто исчезает). Вместо этого следует использовать пару команд `\footnotetext ... \footnotemark`. Описание этих команд вместе с примером их использования приводится в разделе **Сноски**.

Когда вы перемежаете формулы с текстовыми вставками, важно помнить, что в математической моде пробелы всегда игнорируются. Так что выключную формулу

$$\Gamma(n) = (n - 1)! \quad \text{когда } n \text{ целое}$$

надо набирать так:

```
$$\Gamma(n)=(n-1)!\quad\text{когда }n\text{целое}$$
```

Пробелы после **когда** и перед **целое** сохранятся, поскольку они были набраны *внутри* фигурных скобок команды `\text`.

Но есть и более естественный способ получать правильные пробелы в подобных ситуациях. Внутри `\text` можно вернуться в математическую моду, так что можно получать математику, внутри которой находится текст, внутри которого опять математика. Таким образом допускается такой способ набора предыдущей формулы:

```
$$\Gamma(n)=(n-1)! \quad \text{когда } n\$ \text{целое}$$
```

Между математической формулой `$.$.`, включенной в текст абзаца, и формулой, которая находится внутри `\text`'а в выключной формуле, имеется одно важное отличие. В последнем случае математическая формула внутри вставленного текста автоматически будет набрана в размере `d-size` (размере выключенных формул). Таким образом, если вы введете

```
$$
f(a)>f(b)\quad\text{при условии, что }\frac{a}{b+1}>\sqrt{3}\$,
$$
```

то получите

$$f(a) > f(b) \quad \text{при условии, что } \frac{a}{b+1} > \sqrt{3}.$$

Команда `\text`, в отличие от обычного текста, создает из своего аргумента только одну строку текста, неразрывный элемент, который нельзя оформить в виде абзаца. Как правило, именно это и нужно в выключной формуле, но иногда дополнительное условие бывает таким длинным, что из него приходится делать небольшой абзац:

$$\begin{aligned} \sqrt{k+1} - \sqrt{k} &= f(k+1) - f(k) \\ &= f'(x) = \frac{1}{2\sqrt{x}} \quad \text{для некоторого } x \text{ из } (k, k+1), \text{ по} \\ &\quad \text{теореме о среднем} \\ &< \frac{1}{2\sqrt{k}}. \end{aligned}$$

Для подобных ситуаций  $\mathcal{A}\mathcal{M}\mathcal{S}\text{-}\mathcal{T}\mathcal{E}\mathcal{X}$  предоставляет средство `\foldedtext`. Эта выключная формула была набрана так:

```

$$
\align
\sqrt{k+1}-\sqrt k &=f(k+1)-f(k)\!
&=f(x) = \frac{1}{2\sqrt x}\!
\foldedtext\foldedwidth{2in}{для некоторого $x$ из $(k, k+1)$,
по теореме о среднем}\!
&<\frac{1}{2\sqrt k}.
\endalign
$$

```

Если набрать

```
\foldedtext{ ... }
```

то текст “...” оформляется в виде абзаца (но без абзацного отступа в первой строке). По умолчанию ширина такого абзаца определяется форматом, но его можно задать и самостоятельно командой `\foldedwidth`, набрав

```
\foldedtext\foldedwidth{<размер>}{ ... }
```

Обратите внимание, что `\foldedtext` истолковывается как новый символ, и его центральная линия совпадает с центральной линией других символов. Если нужно, чтобы верхний край абзаца находился на одном уровне с верхним краем других символов, то можно воспользоваться `\topfoldedtext`, тогда как управляющая последовательность `\botfoldedtext` даст совпадение нижней кромки.

В конструкциях `\foldedtext` для перехода к следующему абзацу нельзя пользоваться `\par` или пустой строкой. Вместо этого используется `\endgraf`.

Если `\text` находится внутри не выключной формулы, любые математические формулы  $...\$$  внутри `\text` окажутся размера `t-size` (текстового размера). К тому же образуется неразрывная строка текста, в которой с большой вероятностью получится `Overfull box`. Поэтому лучше не вставлять текст в не выключную формулу, а входить и выходить из математической моды. Но в индексах и в индексах второго порядка средство `\text` очень удобно, потому что прямые буквы в индексах меняют свой размер точно так же, как и курсивные буквы математических формул.

Когда мы сказали, что `\text` всегда представляет свой аргумент романским шрифтом, это было не совсем верно. Если `\text` используется в выключных формулах или для однострочных формул, аргумент `\text`’а печатается “текущим шрифтом”. Если `\text` появляется в конструкциях, которые могут изменить текущий шрифт, а почему-либо требуются именно романские буквы, можно набрать `\text{\rm...}`. Но когда `\text` стоит в индексах, романский шрифт выбирается автоматически, поскольку в этом случае вы, вероятно, потому и обращаетесь к `\text`, чтобы явным образом получить романские буквы в математической моде.

#### 4.13. Коррекция математических формул с помощью дополнительных пробелов

Обычно в формулах  $\mathcal{T}\mathcal{E}\mathcal{X}$  задает правильные расстояния между символами различных категорий, автоматически вставляя, где надо, пробелы различной величины, но иногда может потребоваться некоторая поправка. Например, в формуле

$$\int_a^b f(x) dx$$

$dx$  следует отделять от других символов чуть-чуть больше, на маленький интервал, который полиграфисты называют “тонкой шпацией”, а  $\mathcal{T}\mathcal{E}\mathcal{X}$  — тонким пробелом. Для него имеется простая команда `\,`.

Тонкие пробелы `\,` должны также вставляться после восклицательного знака (который в математических выражениях имеет особый смысл, обозначая так называемый “факториал”), если за ним следует число, буква или левый ограничитель:

$$(2n)!/\bigr(n!\,,(n+1)!\bigr) \qquad (2n)!/(n!(n+1)!)$$

$$\frac{52!}{13!13!26!}$$

Тонкие пробелы часто используют после знака квадратного корня, когда первый символ подкоренного выражения слишком близко примыкает к знаку корня:

$$\sqrt{2}x \quad O(1/\sqrt{n})$$

Тонкие пробелы в математической моде также надо ставить между числом и единицей измерения:

$$1 \text{ pc} = 12 \text{ pt}$$

Есть еще группа случаев, когда требуется ручная коррекция формул тонкими пробелами. В индексах  $\TeX$  автоматически не отделяет пробелами, например, бинарные отношения:

$$a^{x+y>z}$$

Для улучшения внешнего вида формулы приходится вокруг  $>$  их ставить вручную:

$$a^{x+y\ ,>\ ,z}$$

Как уже упоминалось (см. **Составные символы**),  $\underset$  и  $\overset$  не так тщательно располагают надписи, как это имеет место в случае математических акцентов. Иногда взаимное расположение символов надо скорректировать с помощью дополнительных пробельчиков.

Если рядом с многоточием стоит точка, то их также следует разделить тонким пробелом:

$$x_1 \cdot x_2 \cdot \dots \cdot x_n$$

(Тонкие пробелы здесь отделяют центрированное многоточие от соседних центрированных точек, обозначающих умножение).

$\TeX$  имеет и отрицательный тонкий пробел  $\!$ , какой удаляет такой же пробел, который  $\!$  добавляет. Приведем несколько примеров того, как  $\!$  и  $\!$  позволяют несколько улучшить внешний вид формул:

$$\begin{aligned} \sqrt{\log x} & \quad \sqrt{\log x} \\ [0, 1) & \quad [0, 1) \\ \log n (\log \log n)^2 & \quad \log n (\log \log n)^2 \\ x^2/2 & \quad x^2/2 \\ n/\log n & \quad n/\log n \\ \Gamma_2 + \Delta^2 & \quad \Gamma_2 + \Delta^2 \\ R_i^{j_{kl}} & \quad R_i^{j_{kl}} \\ \int_1^b \int_a^b & \quad \int_1^b \int_a^b \end{aligned}$$

Кроме тонкой шпации, полиграфисты имеют дело с большими пробелами, называемыми “квадрат” (quad). Например, в случае выключной формулы с дополнительным условием, обычно считается, что между основной формулой и дополнительным условием должен быть пробел в 2 квадрата. Например, формулу

$$F_n = F_{n-1} + F_{n-2}, \quad n > 1.$$

следует получать командами

$$F_n = F_{n-1} + F_{n-2}, \quad n > 1.$$

Пробелы в математической моде подробно описаны в специальном подразделе раздела 4.1.

*Фантомы.* Если вы где-нибудь говорите  $\backslash\text{phantom}\{\langle\text{выражение}\rangle\}$ ,  $\text{T}\text{E}\text{X}$  сделает все пробелы так, как если бы вы просто сказали  $\{\langle\text{выражение}\rangle\}$ , но само выражение будет невидимым. Так, например,  $\backslash\text{phantom}\{0\}2$  занимает в точности столько же места, сколько ‘02’ в текущем шрифте, но в выходном документе даст только 2. Если вы хотите оставить пустое место для нового символа, который имеет в точности такой же размер, как  $X$ , но по каким-то причинам вынуждены вставить этот символ вручную, то  $\backslash\text{phantom}\{X\}$  оставит пустое место в точности нужной величины.

Если набрать  $\backslash\text{hphantom}\{\dots\}$ , то получится “горизонтальный фантом”, ширина которого точно совпадает со строкой в фигурных скобках, а высота равна нулю. Так что это эффективно действует как пробел требуемой ширины.

Еще более полезным, чем  $\backslash\text{hphantom}$ , является  $\backslash\text{vphantom}$ , который создает такой невидимый бокс, высота и глубина которого равны высоте и глубине соответствующего  $\backslash\text{phantom}$ , а ширина равна нулю. Таким образом  $\backslash\text{vphantom}$  создает вертикальную подпорку, которая может увеличить реальную высоту или глубину формулы.

Plain  $\text{T}\text{E}\text{X}$  определяет  $\backslash\text{mathstrut}$  как сокращение для  $\backslash\text{vphantom}(\cdot)$ . Мы уже встречались с применением  $\backslash\text{mathstrut}$  в разделе **Корни**, где с помощью этой команды вместо  $\sqrt{a} + \sqrt{d} + \sqrt{y}$  получалось более симметричное выражение  $\sqrt{a} + \sqrt{d} + \sqrt{y}$ . Напомним, что это делалось так:

```


$$\backslash\sqrt{\backslash\text{mathstrut } a} + \backslash\sqrt{\backslash\text{mathstrut } d} + \backslash\sqrt{\backslash\text{mathstrut } y}$$


```

*Стяжка.*  $\text{T}\text{E}\text{X}$  также предоставляет  $\backslash\text{smash}\{\langle\text{подформула}\rangle\}$ , макрокоманду, которая дает тот же результат, что и  $\{\langle\text{подформула}\rangle\}$ , но делает высоту и глубину равными нулю. Используя как  $\backslash\text{smash}$ , так и  $\backslash\text{vphantom}$ , можно напечатать любую подформулу и задать ей любые желаемые неотрицательные высоту и глубину. Например,

```


$$\backslash\mathop{\backslash\text{smash}\backslash\limsup\backslash\text{vphantom}\backslash\liminf}$$


```

дает большой оператор, который говорит  $\limsup$ , но его высота и глубина такие же, как у  $\backslash\liminf$  (т. е. глубина равна нулю).

Если набрать  $\backslash\text{smash}\{\dots\}$ , то можно убедить  $\text{T}\text{E}\text{X}$ , что ‘...’ не выдается над строкой и не провисает под ней. У команды  $\backslash\text{smash}$  имеются также две разновидности:  $\backslash\text{topsmash}$  и  $\backslash\text{botsmash}$ , в зависимости от того, хотите ли вы, чтобы  $\text{T}\text{E}\text{X}$  проигнорировал часть текста над строкой, или под ней.

#### 4.14. Матрицы

Математики любят рисовать прямоугольные наборы формул, которые устроены из строк и столбцов. Такие наборы называются *матрицами*. В  $\mathcal{A}\mathcal{M}\mathcal{S}$ - $\text{T}\text{E}\text{X}$ ’е есть конструкция  $\backslash\text{matrix} \dots \backslash\text{endmatrix}$ , с помощью которой удобно задавать наиболее общие типы матриц.

Например, предположим, вы хотите задать выключную формулу

$$A = \begin{pmatrix} x - \lambda & 1 & 0 \\ 0 & x - \lambda & 1 \\ 0 & 0 & x - \lambda \end{pmatrix}.$$

Все, что вам надо сделать — это ввести

```


$$\backslash\left( \backslash\text{matrix} \right. \\
x - \backslash\lambda & 1 & 0 \\
0 & x - \backslash\lambda & 1 \\
0 & 0 & x - \backslash\lambda \\
\left. \backslash\text{endmatrix} \right)$$


```

Элементы в строке разделяются знаками  $\&$ , а строки — знаками  $\backslash\backslash$ . Элементы каждого столбца в  $\backslash\text{matrix}$  центрируются, а расстояние между столбцами устанавливается равным  $\backslash\text{quad}$ . Элементы печатаются тем же размером, что и обычный текст, расстояние же между строками тоже равно обычному межстрочному расстоянию.

Вокруг матрицы надо ставить свои собственные ограничители  $\backslash\text{left}$  и  $\backslash\text{right}$ , поскольку различные конструкции матриц используют различные ограничители. С другой стороны, круглые

скобки используются чаще других ограничителей, поэтому если вы хотите, чтобы  $\mathcal{A}\mathcal{M}\mathcal{S}\text{-T}\mathcal{E}\mathcal{X}$  вставил вокруг матрицы круглые скобки, можно использовать `\pmatrix`. Тогда приведенный выше пример сокращается:

```


$$A = \begin{pmatrix} x - \lambda & 1 & 0 \\ 0 & x - \lambda & 1 \\ 0 & 0 & x - \lambda \end{pmatrix}$$


```

Для получения квадратных скобок `\left[ ... \right]` вокруг матрицы в  $\mathcal{A}\mathcal{M}\mathcal{S}\text{-T}\mathcal{E}\mathcal{X}$ е имеется конструкция `\bmatrix ... \endbmatrix`, для вертикальных черточек `\left| ... \right|` — `\vmatrix ... \endvmatrix`, а для двойных вертикальных черточек `\left|| ... \right||` — `\Vmatrix ... \endVmatrix`. Нельзя начать формулу с `\pmatrix`, а закончить ее `\endmatrix`. Это приведет к сообщению об ошибке.

Число столбцов в матрице равно максимальному количеству знаков `&` в ее строках. Строки, содержащие меньше элементов, чем это максимальное число, имеют в остальных столбцах пробелы. Так, если набрать

```


$$\begin{pmatrix} 0 \\ 0 & 1 \\ 0 & 1 & 2 \\ 0 & 1 & 2 & 3 \end{pmatrix}$$


```

то получится “треугольная” матрица

$$\begin{pmatrix} 0 \\ 0 & 1 \\ 0 & 1 & 2 \\ 0 & 1 & 2 & 3 \end{pmatrix}$$

Элементами матрицы могут быть даже матрицы. Например, не трудно получить такую матрицу:

$$\begin{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} & \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} \\ \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix} & \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} \end{pmatrix}$$

где между столбцами заложено по два квадрата, а между строками — по две пробельные строки. Для этого можно ввести

```


$$\begin{matrix} \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} & \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} \\ & \\ & \\ \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix} & \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} \end{matrix}$$


```

снабжая каждый элемент второго столбца пробелом в один квадрат слева (чуть позже мы укажем и другой способ изменять расстояние между столбцами).



Можно изменять расстояние между точками, получаемыми `\hdotsfor` и `\innerhdotsfor`, с помощью команд

```
\spacehdots <число>\for ...
\spaceinnerhdots <число>\for ... \after ...
```

где `<число>` — это десятичное число. Обычно в `\hdotsfor` применяется `<число>= 1.5`. Большее `<число>` даст более редкие точки, а меньшее — более близкие.

Иногда центрирование элементов может быть нежелательным. Например, в матрице

$$\begin{array}{ccc} a & .1 & 1 \\ a + b & .11 & 11 \\ a + b + c & .111 & 111 \end{array}$$

формулы в первом столбце центрированы, второй столбец выровнен по левому краю, а третий — по левому. Для изменения формата матриц в  $\mathcal{A}\mathcal{M}\mathcal{S}\text{-}\mathcal{T}\mathcal{E}\mathcal{X}$ е имеется конструкция `\format ... \`. Приведенная выше матрица была набрана так:

```
$$\matrix \format\c&\quad\l&\quad\r\\
a&.1&1\\
a+b&.11&11\\
a+b+c&.111&111\\
\endmatrix
$$
```

Строка `\format\c&\quad\l&\quad\r` задает новый формат: `\c` говорит, что первый столбец центрируется, `\quad\l` — что второй столбец выровнен по левому краю, и перед ним вставлен пробел в один квадрат, `\quad\r` — что третий столбец выровнен по правому краю, и перед ним также вставлен пробел в один квадрат. Если в строке `\format ...` вместо `&` набрать `&&`, то часть “шаблона”, который идет дальше, будет повторяться столько раз, сколько понадобится. Например,

```
\format \l&&\quad\l \
```

задает формат матрицы, у которой первый столбец выровнен слева, а за ним следует произвольное количество столбцов, каждой из которых также выровнен слева и перед ним стоит пробел в один квадрат. Аналогично, знак `&` в самом начале

```
\format &\quad\l \
```

задает периодическую структуру, определяющую произвольное количество выровненных слева и разделенных пробелом в `\quad` столбцов. Команду `\format` можно использовать также и с `\pmatrix` и т. д.

Можно изменять и расстояние между строками в матрице. Чтобы вставить между *двумя* строками пробел величиной `<размер>`, можно просто сразу после `\` набрать команду

```
\vspace{<размер>}
```

Чтобы увеличить расстояние между *всеми* строками матрицы, нет необходимости после каждой строки помещать `\vspace`, поскольку  $\mathcal{A}\mathcal{M}\mathcal{S}\text{-}\mathcal{T}\mathcal{E}\mathcal{X}$  имеет команду `\spreadmatrixlines`. Если набрать в выключной формуле

```
\spreadmatrixlines{<размер>}
```

то во всех `\matrix` этой формулы междустрочное расстояние увеличится на величину `<размер>`. Если нужно увеличить междустрочное расстояние только для одной матрицы, наберите для этой матрицы

```
{\spreadmatrixlines{<размер>}\matrix
... \endmatrix}
```

До сих пор речь шла о выключных матрицах, но их можно использовать и в тексте абзаца. Использование `\matrix` в тексте не приводит к изменению ее размера, так что просто поместив эту

команду в тексте абзаца, получится что-то вроде  $\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}$ . Это не совсем красиво, и вам, скорее всего, больше понравится такой вариант:  $\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}$ , который получается командами

```
\left(\smallmatrix 1&2\\
3&4\endsmallmatrix\right)
```

Вариантов `\smallmatrix` и им подобных не существует, поэтому необходимые ограничители следует задавать явно. Для `\smallmatrix` можно задавать `\format` и `\vspace`, но нельзя использовать `\spreadmatrixline`.

Следует помнить, что `\matrix` в Plain  $\text{T}_\text{E}\text{X}$ 'е и  $\mathcal{A}\mathcal{M}\mathcal{S}$ - $\text{T}_\text{E}\text{X}$ 'е имеет совершенно разный синтаксис.

#### 4.15. Определения перечислением случаев

Выключные формулы часто используют фигурные скобки, чтобы указать выбор между различными альтернативами, как в конструкции

$$|x| = \begin{cases} x, & \text{если } x \geq 0; \\ -x, & \text{иначе.} \end{cases}$$

Такие конструкции называются *определениями перечислением случаев* и задаются командами `\cases` ... `\endcases`:

```
$$
\vert x\vert=\cases x,&\text{если } \$x\ge0$;\\
-x,&\text{иначе}.\endcases
$$
```

Каждый из случаев имеет две части, которые разделяются символом `&`. Все элементы обрабатываются в математической моде, поэтому для включения текста следует использовать команду `\text`. Пробелы после `&` игнорируются. Случаев может быть любое количество, но не меньше двух. За каждым случаем, кроме последнего, должен следовать `\`. Не следует забывать и о знаках препинания. Заметим, что конструкция `\cases` печатает свою собственную `{` без парной ей `}`.

Приведем еще один пример, состоящий из трех случаев

$$f(x) = \begin{cases} 1/3 & \text{если } 0 \leq x \leq 1; \\ 2/3 & \text{если } 3 \leq x \leq 4; \\ 0 & \text{в других случаях.} \end{cases}$$

который был получен командами

```
$$
f(x)=\cases 1/3&\text{если } \$0\le x\le 1$;\\
2/3&\text{если } \$3\le x\le 4$;\\
0&\text{в других случаях.}
\endcases
$$
```

Так же, как и в матрицах, в `\cases` можно изменять и расстояние между строками. Чтобы вставить между *двумя* строками пробел величиной  $\langle \text{размер} \rangle$ , можно просто сразу после `\` набрать команду

```
\vspace{\langle размер \rangle}
```

Чтобы увеличить расстояние между *всеми* строками `\cases`, можно использовать команду `\spreadmatrixlines`. Если набрать в выключной формуле

```
\spreadmatrixlines{\langle размер \rangle}
```

то во всех `\cases` этой формулы междустрочное расстояние увеличится на величину  $\langle \text{размер} \rangle$ . Если нужно увеличить межстрочное расстояние только для одного `\cases`, наберите для него

```
{\spreadmatrixlines{⟨размер⟩}\cases
... \endcases}
```

До сих пор речь шла о выключных формулах, но `\cases` не запрещается использовать и в обычной математической моде в тексте абзаца. Использование `\cases` в тексте не приводит к изменению ее размера, так что там получится что-то вроде  $|x| = \begin{cases} x, & \dots \\ -x, & \dots \end{cases}$ . Вряд ли это кому-нибудь понравится, но вариантов, аналогичных `\smallmatrix`, не существует,

Следует помнить, что, как и `\matrix`, `\cases` в Plain TeX'e и  $\mathcal{A}\mathcal{M}\mathcal{S}$ -TeX'e имеет совершенно разный синтаксис.

#### 4.16. Коммутативные диаграммы

$\mathcal{A}\mathcal{M}\mathcal{S}$ -TeX прекрасно справляется с простыми коммутативными диаграммами (без диагональных стрелок). Для этой цели используется конструкция `\CD ... \endCD`. В командных скобках `\CD` команды `@>>`, `@<<<`, `@VVV` и `@AAA` дают, соответственно, правую стрелку, левую стрелку, стрелку вниз и стрелку вверх минимального размера. Для горизонтальных стрелок, материал между первым и вторым символами `>` или `<` печатается в виде верхнего индекса, а материал между вторым и третьим символами — в виде нижнего индекса. Если выражение, помещаемое над или под горизонтальными стрелками, достаточно длинное, эти горизонтальные стрелки автоматически увеличиваются. Аналогично, материал между первым и вторым или вторым и третьим символами `A` или `V` вертикальных стрелок будет печататься как левый или правый “боковой индекс”. Например, коммутативная диаграмма

$$(*) \quad \begin{array}{ccc} A & \xrightarrow{\alpha} & B \\ \gamma \downarrow & & \uparrow \delta \\ A' & \xrightarrow{\beta} & B' \end{array}$$

была получена следующими командами

```
$$
\CD
A @>\alpha>> B\\
@V\gamma VV @AA\delta A\\
A' @>>\beta> B'
\endCD \tag{**}
$$
```

Если нужна только часть коммутативной диаграммы (без какого-нибудь “угла” или стрелки, то вместо отсутствующего угла можно оставить пустое место (или поместить `{}`), а вместо отсутствующей стрелки поставить `@.`. Так, получить “усеченный” вариант диаграммы  $(*)$

$$(**) \quad \begin{array}{ccc} A & \xrightarrow{\alpha} & B \\ & & \uparrow \delta \\ & & B' \end{array}$$

можно командами

```
$$
\CD
A @>\alpha>> B\\
@. @AA\delta A\\
\ . B'
\endCD \tag{**}
$$
```

Иногда в коммутативных диаграммах некоторые стрелки заменяются на пары горизонтальных и вертикальных прямых, которые получаются, соответственно, командами `@=` и `@|` (или `@\vert`). Например, преобразуем уже знакомую нам диаграмму

$$\begin{array}{ccc}
 A & \xrightarrow{\alpha} & B \\
 \parallel & & \uparrow \delta \\
 A' & \xlongequal{\quad} & B'
 \end{array}$$

следующими командами:

```

 $\mathcal{A}\mathcal{M}\mathcal{S}$ 
 $\mathcal{T}\mathcal{E}\mathcal{X}$ 
\CD
A @>\alpha>> B\\
@|@AA\delta A\\
A' @= B'
\endCD \tag{\mathcal{A}\mathcal{M}\mathcal{S}}
 $\mathcal{A}\mathcal{M}\mathcal{S}$ 
 $\mathcal{T}\mathcal{E}\mathcal{X}$ 

```

Конечно же, совсем не обязательно диаграмма должна иметь только четыре угловых элемента, соединенных стрелками и прямыми. Можно получать диаграммы, состоящие из множества строк и столбцов. Для примера несколько расширим нашу хорошо знакомую диаграмму:

$$\begin{array}{ccccc}
 A & \xrightarrow{\alpha} & B & \xrightarrow{\beta} & C \\
 \gamma \downarrow & & \parallel & & \uparrow \delta \\
 A' & \xlongequal{\quad} & B' & \xlongequal{\quad} & C' \\
 \epsilon \downarrow & & \parallel & & \uparrow \zeta \\
 E' & \xrightarrow{\iota} & F' & \xrightarrow{\kappa} & G'
 \end{array}$$

что было получено такими командами:

```

 $\mathcal{A}\mathcal{M}\mathcal{S}$ 
 $\mathcal{T}\mathcal{E}\mathcal{X}$ 
\CD
A @>\alpha>> B @>\beta>> C\\
@V\gamma VV @\vert @AA\delta A\\
A' @= B' @= C'\\
@V\epsilon VV @\vert @AA\zeta A\\
E' @>>\iota> F' @>>\kappa> G'
\endCD
 $\mathcal{A}\mathcal{M}\mathcal{S}$ 
 $\mathcal{T}\mathcal{E}\mathcal{X}$ 

```

К сожалению, в настоящей версии `\CD` горизонтальные стрелки одного столбца не устанавливаются в одну длину, и то, что одна из стрелок была увеличена из-за длинного текста над ней, вовсе не означает, что будут увеличены и другие. Так, например, если в диаграмме (\*) над верхней стрелкой вместо  $\alpha$  поставить длинную надпись, то получится:

$$\begin{array}{ccc}
 A & \xrightarrow{\text{длинная надпись}} & B \\
 \gamma \downarrow & & \uparrow \delta \\
 A' & \xrightarrow{\beta} & B'
 \end{array}$$

Чтобы  $\mathcal{A}\mathcal{M}\mathcal{S}$ - $\mathcal{T}\mathcal{E}\mathcal{X}$  растянул и нижнюю стрелку, надо сделать так, чтобы текст  $\beta$  был той же ширины, что и длинная надпись. Это можно сделать, используя конструкцию `\pretend ... \haswidth ...` (“сделай ... имеющим ширину ...”). Если определить новую команду `\bottomarrow`, которая дает горизонтальную стрелку, помещая под ней  $\beta$ , но при этом считая, что ширина  $\beta$  равна ширине длинная надпись и ее использовать вместо традиционной нижней стрелки

```

$$
\define\bottomarrow{@>>\pretend\beta\haswidth
{\text{длинная надпись}}>}
\CD
A @>\text{длинная надпись}>> B \\
@V\gamma VV @AA\delta A \\
A' \bottomarrow B'
\endCD
$$

```

то получится правильный результат

$$\begin{array}{ccc}
 A & \xrightarrow{\text{длинная надпись}} & B \\
 \gamma \downarrow & & \uparrow \delta \\
 A' & \xrightarrow{\beta} & B'
 \end{array}$$

Иногда бывает нужно укоротить стрелки в длинной диаграмме, скажем, чтобы поместить ее на странице. Если набрать

```
\minCDarrowwidth<размер>
```

то минимальная длина стрелок будет равна  $\langle \text{размер} \rangle$ . Команду `\minCDarrowwidth` можно использовать только в выключенной математической моде.

#### 4.17. Формулы в рамках

Формулу (как в тексте абзаца, так и выключную) можно заключить в рамку. Это выполняет команда `\boxed`. Например, если прямо в этом абзаце поместить  $\boxed{x+y}$ , то получится  $\boxed{x+y}$ , а если ввести  $\boxed{\boxed{x+y}}$ , то получится

$$\boxed{x+y}$$

Можно получать и двойные рамочки, дважды используя команду `\boxed`. Например, если ввести

```

\text{Фундаментальный результат:}
\quad \boxed{\boxed{2 \times 2 = 4.}}

```

то получится

Фундаментальный результат:  $\boxed{\boxed{2 \times 2 = 4.}}$

Кроме того, формула в рамке может иметь метку `\tag`, но эту метку ни в коем случае нельзя помещать внутрь `\boxed`, а следует ставить перед закрывающими знаками  $\boxed{\tag{...}}$ .

#### 4.18. Многоточия

Для получения многоточий в математической моде в  $\mathcal{A}\mathcal{M}\mathcal{S}\text{-}\mathcal{T}\mathcal{E}\mathcal{X}$ 'е есть команды `\ldots` для точек на основании строки (...) и `\cdots` для получения точек ( $\cdots$ ), вертикально центрированных на строке. Можно также использовать универсальную команду `\dots`. Эта команда работает как управляющая последовательность с одним аргументом — на основе следующего символа в формуле решает, какого рода точки надо использовать. Команда `\dots` автоматически преобразуется в одну из четырех команд:

```

\dotsc   точки перед запятой (или точкой с запятой),
\dotscb  точки между бинарными операторами и отношениями,
\dotsci  точки между знаками интеграла,
\dotsco  точки, используемые во всех других случаях.

```

Например, когда вы используете `\dots` в математической моде,  $\mathcal{A}\mathcal{M}\mathcal{S}\text{-}\mathcal{T}\mathcal{E}\mathcal{X}$  транслирует это в `\dotsc`, а то, какими эти точки в результате получатся, будет определять тот стиль, который вы используете. В стиле `amsrpt` получится многоточие на основании строки, в то время как другой стиль

может дать многоточие в ее середине, таким образом, конкретный вид точек зависит от стиля и следующего символа формулы. Но бывают ситуации, когда следующего символа нет. В этих случаях `\dots` просто выбирает `\dotso`. Поэтому, если формула заканчивается точками, лучше явно указать  $\mathcal{A}\mathcal{M}\mathcal{S}\text{-}\mathcal{T}\mathcal{E}\mathcal{X}$ 'у, какого именно вида точки нужны.

Есть одна ситуация, когда `\dots` сам справиться не может. Рассмотрим предложение

Объем такого объекта, характеризуемого точками  $A_1A_2A_3A_4$ , очевидно равен  $x_1x_2x_3x_4$ , а в общем случае объем объекта, характеризуемого точками  $A_1A_2A_3 \dots A_n$ , равен  $x_1x_2x_3 \dots x_n$ .

В формуле  $A_1A_2A_3A_4$  буквы  $A_1$ ,  $A_2$ ,  $A_3$  и  $A_4$  просто перечисляются одна за другой. Но в формуле  $x_1x_2x_3x_4$  такое расположение символов означает умножение — эта формула на самом деле представляет собой сокращенную запись  $x_1 \cdot x_2 \cdot x_3 \cdot x_4$  или  $x_1 \times x_2 \times x_3 \times x_4$ . Аналогично, в последней формуле этого предложения расположение символов  $x_1, x_2, \dots$  представляет собой бинарную операцию умножения, которую некоторые авторы любят набирать как  `$\$x_1x_2x_3 \dots x_n\$$` , чтобы получить  $x_1x_2x_3 \dots x_n$ . В идеале, стиль должен сам определять, нужно ли использовать в таких случаях `\dotsb` вместо `\dotso`, но  $\mathcal{A}\mathcal{M}\mathcal{S}\text{-}\mathcal{T}\mathcal{E}\mathcal{X}$  не может понять, имеется ли в виду умножение, или это простое перечисление. Однако  $\mathcal{A}\mathcal{M}\mathcal{S}\text{-}\mathcal{T}\mathcal{E}\mathcal{X}$  может предоставить вам другое средство — `\dotsm`, используемое между перемножаемыми символами. Если набрать

`$\$x_1x_2x_3 \dotsm x_n\$$`

то стиль будет определять, превратить ли `\dotsm` в `\dotsb` или в `\dotso`.

Перед бинарной операцией `\times` команда `\dots` автоматически дает `\dotsb`:

`$\$x_1 \times x_2 \times \dots \times x_n\$$`   $x_1 \times x_2 \times \dots \times x_n$

Но в следующем случае  $\mathcal{A}\mathcal{M}\mathcal{S}\text{-}\mathcal{T}\mathcal{E}\mathcal{X}$  делает исключение из того правила, что `\dots`, за которым следует бинарная операция, дает `\dotsb`:

`$\$x_1 \cdot x_2 \cdot \dots \cdot x_n\$$`   $x_1 \cdot x_2 \cdot \dots \cdot x_n$

Если вы на самом деле хотите отцентрировать здесь точки, вам следует набирать

`$\$x_1 \cdot x_2 \cdot \dots \cdot x_n\$$`   $x_1 \cdot x_2 \cdot \dots \cdot x_n$

(Тонкие пробелы здесь отделяют центрированное многоточие от соседних центрированных точек, обозначающих умножение).

Получение многоточий в матрицах — это тема отдельного обсуждения (см. раздел **Матрицы**).

#### 4.19. Нумерация выключных формул

В математических работах для удобства ссылок на формулы их принято “нумеровать”, т. е. помещать сбоку от них различные метки. Для нумерации формул  $\mathcal{A}\mathcal{M}\mathcal{S}\text{-}\mathcal{T}\mathcal{E}\mathcal{X}$  использует команду `\tag` ..., которая помещается непосредственно перед закрывающими `$$`. Достаточно набрать

`$\$x=y \tag{3-1} \$$`

и получится

$$(3-1) \quad x = y$$

В зависимости от стиля  $\mathcal{A}\mathcal{M}\mathcal{S}\text{-}\mathcal{T}\mathcal{E}\mathcal{X}$  автоматически выберет подходящее место для метки, оформит ее соответствующим образом (в данном случае заключит в круглые скобки), а если формула окажется слишком длинной, поместит метку на отдельной строке:

$$(3-1) \quad \text{очень длинная формула } a + b + c + d + e + \dots + x + y + z \text{ оканчивающаяся здесь}$$

В стиле `amsrpt` метки помещаются слева от формулы.

Нет необходимости набирать `$$ \dots \tag{3-1} $$` с фигурными скобками вокруг метки:  $\mathcal{A}\mathcal{M}\mathcal{S}\text{-}\mathcal{T}\mathcal{E}\mathcal{X}$  знает, что метка — это все, что располагается между `\tag` и закрывающими знаками `$$`. Скобки вокруг метки ставятся автоматически; в некоторых стилях применяются иные способы оформления метки, например `[3-1]` или **3-1** и т. д. Метка обрабатывается как обычный текст, а не как формулы в математической моде, так что `-` и `--` дают дефис и короткое тире, а не знаки минуса.

В стиле `amsppt` нумерацию формул слева можно заменить нумерацией справа, введя команду `\TagsOnRight`. Чтобы вернуть нумерацию слева, надо ввести `\TagsOnLeft`. Эти команды “глобальные”: они действуют на весь последующий текст, даже если используются внутри группы `{ ... }` или между знаками `$`. Журнальные стили такие инструкции обычно игнорируют и оставляют свой способ размещения метки.

Вам следует выбрать либо `\TagsOnLeft` (что задается автоматически в стиле `amsppt`), либо `\TagsOnRight`. Не пытайтесь использовать эти управляющие последовательности для того, чтобы на четных страницах получить нумерацию слева, а на нечетных — справа! Это совершенно иной стиль.

Если у вас масса формул, метками которых являются математические выражения типа  $(A_1)$  и  $(A')$  или  $(*)$  и  $(**)$ , а чисто текстовые символы (типа дефисов и тире) в метках используются редко, то вы можете предложить  $\mathcal{A}\mathcal{M}\mathcal{S}\text{-}\mathcal{T}\mathcal{E}\mathcal{X}$  трактовать номера формул как математические формулы `\tag{A_2$}`, а не как текст. Для этого надо ввести команду `\TagsAsMath`; а `\TagsAsText` вернет противоположное соглашение. Эти команды также “глобальные”. Журнальные стили такие инструкции не игнорируют.

В некоторых стилях номера формул вертикально центрируются даже для формул, разбитых на несколько строк (см. **Многострочные формулы**). Это можно сделать и в стиле `amsppt`, если ввести команду

```
\CenteredTagsOnSplit
```

А затем можете вернуться к обычному позиционированию номеров:

```
\TopOrBottomTagsOnSplit
```

Это также “глобальные” команды. Если вам нужно получить вертикально центрированный номер только для одной разбитой на несколько строк формулы, эти управляющие последовательности использовать не нужно; просто наберите такую формулу как `\aligned`.

Вам может понадобиться отказаться от принятого в данном стиле способа оформления метки для какой-нибудь конкретной формулы. Например, нужно получить для данной формулы жирный номер в жирных скобках **(3)**. Для этого можно использовать `\tag` с “аргументом в виде литеры”. Наберите

```
$$ ... \tag"\bf(3)" $$
```

и номер появится точно в том виде, как вы его набрали между двойными кавычками.

Если внутри текста нужно сослаться на формулу (17), то в разных стилях это может делаться по-разному, например, [17] или **17**. По этому поводу  $\mathcal{A}\mathcal{M}\mathcal{S}\text{-}\mathcal{T}\mathcal{E}\mathcal{X}$  предлагает вам набирать `\thetag{17}`; тогда вы получите именно то, что принято в данном стиле.

#### 4.20. Выравнивание выключных формул

Если в математической работе встречаются несколько выключенных формул подряд, то их часто надо выравнивать по какому-нибудь символу. Например

$$(1) \quad \max(f, g) = \frac{f + g + |f - g|}{2},$$

$$(2) \quad \max(f, -g) = \frac{f - g + |f + g|}{2}.$$

Здесь сделано выравнивание по знакам  $=$ , а затем обе формулы расположены по центру как одно целое. Это достигается при помощи конструкции  $\mathcal{A}\mathcal{M}\mathcal{S}\text{-}\mathcal{T}\mathcal{E}\mathcal{X}$ 'а `\align`:

```
$$\align
\max(f,g) &=\frac{f+g+|f-g|}{2}, \tag1 \\
\max(f,-g) &=\frac{f-g+|f+g|}{2}. \tag2
\endalign$$
```

Нельзя ничего помещать между открывающими `$$` и `\align`, а также между `\endalign` и закрывающими `$$`. Формулы в такой группе разделяются знаками `\\` (по после последней формулы эти

знаки не ставятся). Любая из формул может иметь номер, получаемый, как обычно, командой `\tag`, которая помещается в самом конце формулы (перед `\\` или перед `\endalign`). Выравнивание происходит, конечно же, не обязательно по знакам равенства, а по тем символам, непосредственно перед которыми стоит знак `&`.

Для получения текста, располагающегося между строками группы выровненных формулы и не нарушающего их выравнивания, используется команда

```
\intertext{ ... }
```

которая помещается между двумя строками. Например, чтобы получить

Мы имеем

$$X = (-1)^{i+j-k/3+*[\alpha,\beta]} Z_1 + (-1)^{\alpha/\beta-*[i+j/2,i+k/3]} Z_2$$

по свойствам (a)–(d) из \*, учитывая коммутативность кольца,

$$= \alpha Z_1 + \beta Z_2,$$

что дает требуемую формулу.

с прекрасно выровненными знаками =, надо ввести

Мы имеем

```
$$
```

```
\align X&= (-1)^{i+j-k/3+*[\alpha,\beta]}Z_1
```

```
+(-1)^{\alpha/\beta-*[i+j/2,i+k/3]}Z_2\\
```

```
\intertext{по свойствам (a)--(d) из **$,}
```

```
учитывая коммутативность кольца,}
```

```
&=\alpha Z_1+\beta Z_2,
```

```
\endalign
```

```
$$
```

что дает требуемую формулу.

Для перехода на новый абзац в `\intertext` нельзя применять `\par` или пустую строку, потому что в математической моде `\par` недопустим. Вместо этого используется команда `\endgraf`. В первой строке в `\intertext` абзацный отступ не делается (если только перед ней не стоит `\endgraf`). За исключением редких случаев, использование `\intertext` считается дурным тоном (во всяком случае, так считает Спивак).

Конструкция `\align` получает выровненные формулы, ширина которых считается равной ширине страницы. Иногда же надо просто собрать несколько выровненных формул в один формульный массив, и в одной выключке использовать несколько таких массивов:

$$\left\{ \begin{array}{l} \alpha = f(z) \\ \beta = f(z^2) \\ \gamma = f(z^3) \end{array} \right\} \quad \left\{ \begin{array}{l} x = \alpha^2 - \beta \\ y = 2\gamma \end{array} \right\}.$$

Для обработки такого выключного материала  $\mathcal{A}\mathcal{M}\mathcal{S}$ - $\text{T}\text{E}\text{X}$  предоставляет `\aligned ... \endaligned`. Конструкция `\aligned ... \endaligned` очень похожа на `\align ... \endalign` с тем отличием, что последняя указывает  $\text{T}\text{E}\text{X}$ у выровнять последовательность строк по всей ширине страницы, а первая создает один выровненный массив, который рассчитывается на ширину, достаточную для всех входящих в него формул, и с которым затем можно обращаться, как с одним символом. Приведенная выше выключная формула была получена так:

```
$$
```

```
\left\{
```

```
\aligned \alpha&=f(z)\ \beta&=f(z^2)\
```

```
\gamma&=f(z^3)\endaligned
```

```
\right\}\quad\left\{
```

```
\aligned x&=\alpha^2-\beta\ y&=2\gamma\endaligned \right\}.
```

\$\$

Когда `\aligned ... \endaligned` занимает всю выключную формулу, результат выглядит точно так же, как если бы было использовано `\align ... \endalign`. Но в этих двух конструкциях совершенно по-разному работает `\tag`. В случае `\align`, метки `\tag` можно ставить после каждой формулы, и нельзя поставить `\tag` после `\endalign`. В случае `\aligned`, ситуация прямо противоположная: поскольку все это один объект, нельзя снабдить меткой `\tag` отдельные строки, но можно поставить `\tag` после `\endaligned`. Например, ввод

```

\aligned \alpha&=f(z)\
\beta&=f(z^2)\
\gamma&=f(z^3)\endaligned\tag 22

```

дает

$$(22) \quad \begin{aligned} \alpha &= f(z) \\ \beta &= f(z^2) \\ \gamma &= f(z^3) \end{aligned}$$

В конструкции `\aligned`, нельзя использовать ни команды разрыва страницы `\allowdisplaybreak[s]` и `\displaybreak`, описанные ниже (см. *Разрыв страниц в выключенных формулах*), ни `\intertext`.

Как показано в предыдущем примере, различные `\aligned`, расположенные в одной выключке на одной строке, выравниваются по их центру. Имеются также `\topaligned` и `\botaligned`, в которых выравнивание происходит по верхним или нижним строкам.

Иногда бывает нужно выровнять формулы более чем по одной позиции. Например, в выключной формуле

$$(23) \quad V_i = v_i - q_i v_j, \quad X_i = x_i - q_i x_j, \quad U_i = u_i, \quad \text{для } i \neq j;$$

$$(24) \quad V_j = v_j, \quad X_j = x_j, \quad U_j = u_j + \sum_{i \neq j} q_i u_i.$$

выравнивание по знаку `=` проведено в трех местах. Это достигается при помощи средства  $\mathcal{A}\mathcal{M}\mathcal{S}$ -TeX'a `\alignat`. Приведенная выше формула была набрана так:

```

\alignat 3
V_i & =v_i-q_iv_j, & \quad X_i & =x_i-q_ix_j, \\
& \quad U_i & =u_i, \quad \text{для } i \neq j; \tag 23 \\
V_j & =v_j, & \quad X_j & =x_j, \\
& \quad U_j & =u_j+\sum_{i \neq j} q_i u_i. \tag 24 \\
\endalignat

```

Сразу после `\alignat` нужно указать количество пар формул, которые вы хотите выровнять. Поскольку у нас `\alignat 3`, каждая строка содержит 3 пары формул со знаком `&` между этими формулами в каждой паре. Кроме того, знак `&` необходим между первой и второй парами формул, а также между второй и третьей. Так что всего на такой строке нужно 5 знаков `&`. Пробел `\quad` между различными столбцами указывается явно.

Чаще всего `\alignat` используется для формул вроде

$$\begin{aligned} x &= y && \text{по (1)} \\ x' &= y' && \text{по Аксиоме 2} \\ x + x' &= y + y' && \text{по Теореме 1.} \end{aligned}$$

Это было набрано так:

\$\$

```
\alignat2
x&=y &&\quad\text{по (1)}\\
x'&=y'&&\quad\text{по Аксиоме 2}\\
x+x'&=y+y'&&\quad\text{по Теореме 1.}
\endalignat
$$
```

Мы использовали `&&`, потому что хотели, чтобы условия в правой части рассматривались как вторая формула второй пары и чтобы они были выровнены слева.

Как и в `\align`, внутри `\alignat` можно использовать `\allowdisplaybreak` (см. ниже *Разрыв строки в выключенных формулах*), `\intertext` и т. д.

У `\alignat` есть также “расширенный” вариант `\xalignat`, при котором различные столбцы располагаются по странице равномерно, так что нет необходимости указывать между ними конкретный пробел. Например, в приведенном выше примере с тремя столбцами уберем все вставленные между столбцами пробелы и используем `\xalignat`:

```
$$
\xalignat 3
V_i & =v_i-q_{iv}_j, & X_i & =x_i-q_{ix}_j,
& U_i & =u_i, \text{для } i \neq j; \tag 23 \\
V_j & =v_j, & X_j & =x_j,
& U_j & =u_j+\sum_{i \neq j} q_{iu}_i. \tag 24
\endxalignat
$$
```

Введя такую конструкцию, мы получим

$$(23) \quad V_i = v_i - q_{iv}_j, \quad X_i = x_i - q_{ix}_j, \quad U_i = u_i, \text{ для } i \neq j;$$

$$(24) \quad V_j = v_j, \quad X_j = x_j, \quad U_j = u_j + \sum_{i \neq j} q_{iu}_i.$$

Средство `\xalignat` еще шире — первый и последний столбцы располагаются вдоль самых краев полосы набора:

$$V_i = v_i - q_{iv}_j, \quad X_i = x_i - q_{ix}_j, \quad U_i = u_i, \text{ для } i \neq j;$$

$$V_j = v_j, \quad X_j = x_j, \quad U_j = u_j + \sum_{i \neq j} q_{iu}_i.$$

Здесь `\tag` не имеет смысла и в этой конструкции недопустим.

Существует также средство `\alignedat`, которое относится к `\alignat` точно так же, как `\aligned` относится к `\align`, т. е. создается единый массив формул, который потом можно на одной строке с другими формулами. Внутри `\alignedat` можно использовать те же управляющие последовательности, что и в `\aligned`. Средства `\xalignedat` нет, поскольку в нем нет смысла.

Есть одно важное отличие `\alignat` и `\xalignat` от `\align` и аналогичных конструкций. Если в `\alignat` на строке не хватает места для размещения и формулы, и метки, метка не будет автоматически помещаться на отдельной строке. Метка может даже наложиться на формулу без выдачи сообщения `Overfull box`. Если возникает такая проблема, нужно перенести метку на другую строку, введя пустую формулу.

Иногда автору хочется объединить вместе несколько формул и не выравнивать их, а отцентрировать:

$$a = b + c$$

$$d = e$$

$$f + g = h$$

Если вы наберете их как отдельные выключные формулы

```
$$a=b+c$$ $$d=e$$ $$f=g+h$$,
```

то между ними останется слишком много места. Так что лучше пользоваться командой `\gather`:

```
$$
\gather a=b+c\ d=e\
f=g+h\endgather
$$
```

Как и в случае `\align`, каждой формуле можно дать свою метку `\tag`.

Наконец, имеется одна особая проблема выравнивания, для которой у  $\mathcal{A}\mathcal{M}\mathcal{S}$ - $\mathcal{T}\mathcal{E}\mathcal{X}$ 'а есть особое средство. Иногда вам может понадобиться что-то вроде

$$\begin{aligned} a + b &= c \\ f(a) + f(b) &= f(c) \\ \alpha &= \beta + \delta \\ \alpha' &= \beta' + \delta' \\ A + B &= C + D + E \end{aligned}$$

где несколько формул оформлены посредством `\gather`, но отдельные группы должны быть выровнены (и каждая формула может иметь метку). По идее, нельзя вкладывать `\align` или `\gather` внутрь другой конструкции, но  $\mathcal{A}\mathcal{M}\mathcal{S}$ - $\mathcal{T}\mathcal{E}\mathcal{X}$  допускает такой набор:

```
$$
\gather a+b=c\
f(a)+f(b)=f(c)\
{\align \alpha&=\beta+\delta\
\alpha'&=\beta'+\delta' \endalign}\
A+B=C+D+E\endgather
$$
```

с метками у любых формул, где это нужно. В этом особом случае каждая подгруппа, оформленная посредством `\align ... \endalign`, должна быть заключена в фигурные скобки.

$\mathcal{A}\mathcal{M}\mathcal{S}$ - $\mathcal{T}\mathcal{E}\mathcal{X}$  предоставляет также средство `\gathered ... \endgathered`, которое так же относится к `\gather`, как `\aligned` относится к `\align`. Иными словами, конструкция `\gathered ... \endgathered` порождает объект, который может обрабатываться как новый символ — он может использоваться внутри других формул, и ему может быть присвоена метка `\tag`, располагающаяся по центру относительно всего массива.

*Изменение расстояния между строками.* Расстояние между строками во всех только что описанных математических конструкциях, использующих `\` для разбиения на строки, можно изменять. Для этого в  $\mathcal{A}\mathcal{M}\mathcal{S}$ - $\mathcal{T}\mathcal{E}\mathcal{X}$ 'е есть два способа.

Для вставки после какой-нибудь строки дополнительного пробела величиной  $\langle \text{размер} \rangle$ , можно после `\`, оканчивающих эту строку, поставить

```
\vspace{\langle \text{размер} \rangle}
```

Хотя можно указать любой допустимый в  $\mathcal{T}\mathcal{E}\mathcal{X}$ 'е  $\langle \text{размер} \rangle$ , обычно его выражают в особых  $\mathcal{T}\mathcal{E}\mathcal{X}$ 'овских единицах `\jot` в виде `1\jot` или `.5\jot`, и тогда такая корректировка даст нужный эффект в любом формате (в формате Plain, например, `\jot` равно 3pt).

Если надо добавить вертикальные пробелы  $\langle \text{размер} \rangle$  между *всеми* строками, то вместо того, чтобы после всех `\` ставить `\vspace\langle \text{размер} \rangle`, что, безусловно, очень скучно и утомительно, можно использовать команду  $\mathcal{A}\mathcal{M}\mathcal{S}$ - $\mathcal{T}\mathcal{E}\mathcal{X}$ 'а `\spreadlines`. Так, если в некоторой выключной формуле набрать

```
$$
\spreadlines{\langle \text{размер} \rangle}
...
```

...  
 $\$ \$$

то все строки этой формулы раздвинутся на  $\langle \text{размер} \rangle$ . Команду `\spreadlines` можно использовать только в выключных формулах и ее действие распространяется только на эту формулу.

*Разрыв страниц в выключных формулах.* Выравниваемые формулы обычно воспринимаются как одно целое, так что  $\mathcal{A}\mathcal{M}\mathcal{S}\text{-}\mathcal{T}\mathcal{E}\mathcal{X}$ , как правило, не позволяет переносить на другую страницу часть формулы между `\align ... \endalign`. Но набрав `\allowdisplaybreak` после какого-нибудь `\`, вы получите возможность разорвать формулу после этой строки и перенести ее на другую страницу. Можно также *заставить*  $\mathcal{T}\mathcal{E}\mathcal{X}$  сделать разрыв после какой-нибудь строки, применив после `\` `\displaybreak`. Если поставить *перед* `\align` команду `\allowdisplaybreaks`, это даст тот же эффект, как если бы после каждой строки стояло `\allowdisplaybreak`. Команда `\allowdisplaybreaks` разрешена только между знаками  $\$ \$$ , и ее действие распространяется лишь на ту группу формул, в которой она присутствует.

Кроме того, как уже упоминалось в разделе **Разрыв страницы**, в выключенной математической моде можно использовать команду `\pagebreak`. Для сохранения целостности изложения повторим здесь, что чтобы вызвать разрыв страницы после выключной формулы, следует поместить `\pagebreak` внутри этой формулы, поскольку, если поставить `\pagebreak` сразу после формулы, то на странице после нее появится дополнительное пустое пространство.

#### 4.21. Многострочные формулы

Иногда выключную формулу невозможно напечатать в одну строку, потому что она на ней не помещается, несмотря на все усилия  $\mathcal{T}\mathcal{E}\mathcal{X}$ 'а сжать входящие в нее члены:

$$(a + b)^{n+1} = (a + b)(a + b)^n = (a + b) \sum_{j=0}^n \binom{n}{j} a^{n-1} b^j = \sum_{j=0}^n \binom{n}{j} a^{n+1-j} b^j + \sum_{j=1}^n \binom{n}{j-1} a^{n-j} b^j$$

Эту формулу, составленную из нескольких связанных между собой более мелких формул, можно разбить на несколько строк, используя команду `\align` с выравниванием по символам бинарных операций (см. **Выравнивание выключных формул**). Таким образом, даже еще более длинные формулы можно представить в виде одной выключной:

$$\begin{aligned} (a + b)^{n+1} &= (a + b)(a + b)^n = (a + b) \sum_{j=0}^n \binom{n}{j} a^{n-1} b^j \\ &= \sum_{j=0}^n \binom{n}{j} a^{n+1-j} b^j + \sum_{j=1}^n \binom{n}{j-1} a^{n-j} b^j \\ &= \sum_{j=0}^n \binom{n+1}{j} a^{n+1-j} b^j \end{aligned}$$

что было получено так:

```

 $\$ \$$ 
\align
(a+b)^{n+1}
&=(a+b)(a+b)^n=(a+b)
\sum_{j=0}^n\binom{n}{j}a^{n-1}b^j\
&=\sum_{j=0}^n\binom{n}{j}a^{n+1-j}b^j+
\sum_{j=1}^n\binom{n}{j-1}a^{n-j}b^j\
&=\sum_{j=0}^n\binom{n+1}{j}a^{n+1-j}b^j
\endalign
 $\$ \$$ 

```

Дело усложняется, как только разбитая на части формула снабжается меткой (см. раздел **Нумерация выключных формул**). Если метки помещаются слева, то они размещаются около первой

строки формулы:

$$\begin{aligned}
 (1-2) \quad (a+b)^{n+1} &= (a+b)(a+b)^n = (a+b) \sum_{j=0}^n \binom{n}{j} a^{n-1} b^j \\
 &= \sum_{j=0}^n \binom{n}{j} a^{n+1-j} b^j + \sum_{j=1}^n \binom{n}{j-1} a^{n-j} b^j \\
 &= \sum_{j=0}^n \binom{n+1}{j} a^{n+1-j} b^j
 \end{aligned}$$

Но если метка справа — то около нижней строки:

$$\begin{aligned}
 (a+b)^{n+1} &= (a+b)(a+b)^n = (a+b) \sum_{j=0}^n \binom{n}{j} a^{n-1} b^j \\
 &= \sum_{j=0}^n \binom{n}{j} a^{n+1-j} b^j + \sum_{j=1}^n \binom{n}{j-1} a^{n-j} b^j \\
 &= \sum_{j=0}^n \binom{n+1}{j} a^{n+1-j} b^j \tag{1-2}
 \end{aligned}$$

Для этой цели у  $\mathcal{A}\mathcal{M}\mathcal{S}\text{-}\mathcal{T}\mathcal{E}\mathcal{X}$ 'а есть команда `\split`, которая предоставляет возможность набирать разбиваемые на части формулы, не заботясь о том, как обстоит дело с метками. Если вы наберете

```


$$\begin{aligned}
 & (a+b)^{n+1} \\
 &= (a+b)(a+b)^n = (a+b) \\
 & \sum_{j=0}^n \binom{n}{j} a^{n-1} b^j \\
 &= \sum_{j=0}^n \binom{n}{j} a^{n+1-j} b^j + \\
 & \sum_{j=1}^n \binom{n}{j-1} a^{n-j} b^j \\
 &= \sum_{j=0}^n \binom{n+1}{j} a^{n+1-j} b^j
 \end{aligned}$$


```

то  $\mathcal{A}\mathcal{M}\mathcal{S}\text{-}\mathcal{T}\mathcal{E}\mathcal{X}$  автоматически выдаст результат, соответствующий используемому формату.

Команду `\tag` надо помещать после всей конструкции `\split ... \endsplit`; если попытаете пометить отдельную строку, то получите таинственное сообщение об ошибке, так как  $\mathcal{A}\mathcal{M}\mathcal{S}\text{-}\mathcal{T}\mathcal{E}\mathcal{X}$  воспринимает конструкцию `\split ... \endsplit` как единое целое. Обратите внимание также, что знак `\\` перехода на другую строку помещается *перед* бинарной операцией, тогда как формулы в тексте, как упоминалось ранее, при необходимости разрываются *после* бинарных операций.

Иногда формулу разбивают так, что, казалось бы, нет выравнивания:

$$\begin{aligned}
 (f \circ g)'''(x) &= [f'''(g(x)) \cdot g'(x)^3 + 2f''(g(x)) \cdot g'(x)g''(x)] \\
 &+ [f''(g(x)) \cdot g'(x)g''(x) + f'(g(x)) \cdot g'''(x)]
 \end{aligned}$$

В таких случаях принято оставлять по крайней мере два квадрата пробела перед второй частью формулы. Приведенная выше формула была набрана так:

```


$$\begin{aligned}
 & (f \circ g)'''(x) = \bigl[ f'''(g(x)) \cdot g'(x)^3 + \\
 & 2f''(g(x)) \cdot g'(x)g''(x) \bigr] \\
 & + \bigl[ f''(g(x)) \cdot g'(x)g''(x) + \\
 & f'(g(x)) \cdot g'''(x) \bigr] \\
 & \endsplit
 \end{aligned}$$


```

так что `=\bigl[` первой строки выровнено с невидимым `\qqquad` второй строки.

Для формул, состоящих из нескольких строк, иногда достаточно воспользоваться конструкцией `\multline ... \endmultline`, разделяя строки с помощью `\\`. В этом случае первая строка формулы придвигается почти вплотную влево, последняя — почти вплотную вправо:

$$\int_a^b \left\{ \int_a^b [f(x)^2 g(y)^2 + f(y)^2 g(x)^2 - 2f(x)g(x)f(y)g(y) dx] dy \right\} \\ = \int_a^b \left\{ g(y)^2 \int_a^b f^2 + f(y)^2 \int_a^b g^2 - 2f(y)g(y) \int_a^b fg \right\} dy$$

Эта формула была получена так:

```


$$\int_a^b \left\{ \int_a^b [f(x)^2 g(y)^2 + f(y)^2 g(x)^2 - 2f(x)g(x)f(y)g(y) dx] dy \right\} \\ = \int_a^b \left\{ g(y)^2 \int_a^b f^2 + f(y)^2 \int_a^b g^2 - 2f(y)g(y) \int_a^b fg \right\} dy$$


```

Если в формуле более двух строк, все строки между первой и последней располагаются по центру. Однако, любую из этих строк можно отодвинуть влево или вправо, набрав

```
\shoveleft{ ... }\\
```

или

```
\shoveright{ ... }\\
```

`\shoveleft` и `\shoveright` представляют собой управляющие последовательности с одним аргументом; вся строка, которую вы хотите сдвинуть, заключается в фигурные скобки, и в конце все так же ставится `\\`.

Точное расстояние от полей определяется форматом. В стиле `amspp` закладывается один квадрат; в некоторых форматах пробел может вовсе отсутствовать. Расстояния до левого и правого краев, которые оставляет управляющая последовательность `\multline`, можно менять при помощи команды `\multlinegap{⟨размер⟩}`. Например, если ваша формула `\multline` несколько шире, чем нужно, чтобы уместиться в отведенное ей место, можно удалить пробелы слева и справа, набрав

```


$$\int_a^b \left\{ \int_a^b [f(x)^2 g(y)^2 + f(y)^2 g(x)^2 - 2f(x)g(x)f(y)g(y) dx] dy \right\} \\ = \int_a^b \left\{ g(y)^2 \int_a^b f^2 + f(y)^2 \int_a^b g^2 - 2f(y)g(y) \int_a^b fg \right\} dy$$


```

Когда формула завершится, прежние пробелы будут восстановлены. Вместо команды `\multlinegap{0pt}` можно использовать `\nomultlinegap`, что то же самое.

Управляющая последовательность `\multlinegap` допускается только внутри знаков `$$$ ... $$$`. Но имеется также `\MultLineGap` для изменения расстояния до полей во всех многострочных формулах на все время работы; таким образом создается новый стиль.

После `\multline ... \endmultline` можно задать метку `\tag`, точно так же, как и после `\split ... \endsplit`. Например,

```


$$\int_a^b \left\{ \int_a^b [f(x)^2 g(y)^2 + f(y)^2 g(x)^2 - 2f(x)g(x)f(y)g(y) dx] dy \right\} \\ = \int_a^b \left\{ g(y)^2 \int_a^b f^2 + f(y)^2 \int_a^b g^2 - 2f(y)g(y) \int_a^b fg \right\} dy$$


```

\$\$

в зависимости от формата дает либо

$$(17) \quad \int_a^b \left\{ \int_a^b [f(x)^2 g(y)^2 + f(y)^2 g(x)^2 - 2f(x)g(x)f(y)g(y) dx] dy \right. \\ \left. = \int_a^b \left\{ g(y)^2 \int_a^b f^2 + f(y)^2 \int_a^b g^2 - 2f(y)g(y) \int_a^b fg \right\} dy \right.$$

либо

$$\int_a^b \left\{ \int_a^b [f(x)^2 g(y)^2 + f(y)^2 g(x)^2 - 2f(x)g(x)f(y)g(y) dx] dy \right. \\ \left. = \int_a^b \left\{ g(y)^2 \int_a^b f^2 + f(y)^2 \int_a^b g^2 - 2f(y)g(y) \int_a^b fg \right\} dy \right. \quad (17)$$

Как и в случае `\split`, не следует помещать `\tag` на той или иной строке, метка должна следовать после всей конструкции `\multline ... \endmultline`. Более того, `\multline` создает строки на ширину всей страницы, подобно `\align`, так что вы не сможете поместить это внутрь какой-либо другой конструкции: ни между первым `$$` и `\multline`, ни между `\endmultline` и последним `$$` ничего быть не должно.

Для изменения межстрочных пробелов в `\multline` можете пользоваться управляющими последовательностями `\vspace` и `\spreadlines` (см. в конце предыдущего раздела *Изменение расстояния между строками*).

#### 4.22. Шрифты в математике

*Жирные символы в математической моде.* Жирные буквы в математических формулах получаются командой `\bold`. Команда `\bold` является командой с одним аргументом, а не командой переключения шрифта (как `\bf` в тексте). Это особенно понятно на примере формулы, в которой жирные буквы перемежаются с обычными курсивными:

$$\mathbf{a}\bold{x} + \mathbf{b}\bold{y}^n \qquad ax + by^n$$

Буквы, к которым применена команда `\bold` автоматически ставятся в нужном размере (например, уменьшаются в индексах):

$$\mathbf{a}^{\mathbf{x}} + \mathbf{b}^{\mathbf{y}} \qquad a^x + b^y$$

Если в работе в математических формулах часто встречаются жирные символы `\bold x`, `\bold y`, `\bold z`, то, вероятно, имеет смысл ввести определения

```
\define\x{\bold x}
\define\y{\bold y}
\define\z{\bold z}
```

после чего можно будет вводить

$$\mathbf{x}^{\mathbf{y}+\mathbf{z}} + \mathbf{z}^{\mathbf{x}+\mathbf{y}} \qquad x^{y+z} + z^{x+y}$$

В принципе, для получения `x + y` можно вводить `\bold{x+y}`, но с этим следует быть осторожным. Так, например, `\bold{ff}` даст жирную лигатуру **ff**, а совсем не два жирных символа **f****f**, как вы, вероятно, предполагали.

Команду `\bold` следует применять только к буквам. Если в ее аргумент попадает, например, знак `+`, то он не меняет свой вид. Но после того, как во входном файле встретилась команда `\loadbold`, становятся доступными и другие жирные символы (в последних версиях  $\mathcal{A}\mathcal{M}\mathcal{S}$ - $\mathcal{T}\mathcal{E}\mathcal{X}$  и `AMSFonTS`). Для различных видов жирных символов используются две управляющие последовательности:

<code>\boldkey</code>	для символов, имеющих на клавиатуре
<code>\boldsymbol</code>	для символов, задаваемых одной командой

Например,

`\bold x \boldsymbol{\in} \boldsymbol{\varGamma}`

дает

$$\mathbf{x} \in \mathbf{\Gamma},$$

`a \boldsymbol{\lbrack} a \boldsymbol{\rbrack}` дает  $[a]$ , если вам приходится использовать `\lbrack` и `\rbrack` вместо клавиш `[` и `]`.

Более точно, `\boldkey` можно использовать в математических формулах в следующих комбинациях:

- С любым из символов

`+ - = < > ( ) [ ] | / * . , : ; ! ?`

чтобы получить

$\mathbf{+ - = < > ( ) [ ] | / * . , : ; ! ?}$

Как уже упоминалось, для получения жирных вариантов этих символов нельзя использовать `\bold`: `\bold+` даст только обычный `+`, и т. д.

Жирные `+` и `-` остаются бинарными операциями, так же, как и обычные символы `+` и `-`; жирное `=` будет бинарным отношением, как и обычный `=`, и т. д.

- С буквами:

`\boldkey a$, ... , \boldkey z$`  $\mathbf{a, \dots, z}$   
`\boldkey A$, ... , \boldkey Z$`  $\mathbf{A, \dots, Z}$

Обратите внимание, что это буквы шрифта *bold math italic*, в отличие от жирных текстовых букв  $\mathbf{a, \dots, z, A, \dots, Z}$ , которые вы получите, используя в математической моде команду `\bold`.

- С цифрами:

`\boldkey 0$, ... , \boldkey 9$`  $\mathbf{0, \dots, 9}$

Однако, это дает те же самые числа, которые получаются командами `\bold0$, ... , \bold9$`.

Конструкцию `\boldsymbol` можно использовать в любой из следующих комбинаций:

- С прописными и строчными греческими буквами

`\boldsymbol{\Gamma}$, ...`  $\mathbf{\Gamma, \dots}$   
`\boldsymbol{\varGamma}$, ...`  $\mathbf{\Gamma, \dots}$   
`\boldsymbol{\alpha}$, ...`  $\mathbf{\alpha, \dots}$

Прямые прописные жирные греческие буквы являются частью обычного жирного шрифта, поэтому для их получения не нужно дополнительной команды загрузки шрифта. Однако, строчные и наклонные прописные жирные греческие буквы автоматически не загружаются, так что перед тем, как их использовать, следует указать `\loadbold`. В версиях  $\mathcal{A}\mathcal{M}\mathcal{S}$ - $\mathcal{T}\mathcal{E}\mathcal{X}$ 'а, ранних, чем 2.0, жирные прямые прописные греческие буквы  $\mathbf{\Gamma, \dots, \Omega}$  получались командами `\boldGamma, ... , \boldOmega`; теперь эти управляющие последовательности отсутствуют.

- Для удобства за `\boldsymbol` также может следовать буква (но не цифра или другой символ); это дает тот же результат, что и `\boldkey`.
- Также можно применять `\boldsymbol` ко всем другим стандартным символам, которые задаются одной управляющей последовательностью. например, получить жирный “прим”:

`\boldsymbol{\prime}$`  $\mathbf{'}$   
`\boldsymbol A^{\boldsymbol{\prime}}$`  $\mathbf{A'}$

(Но `\boldsymbol'`, использующий сокращенную запись для `\prime`, работать не будет.)

- Можно применять `\boldsymbol` “ограничителям”:

<code>\boldsymbol\{... \boldsymbol\}</code>	$\{\dots\}$
<code>\boldsymbol\langle ... \boldsymbol\rangle</code>	$\langle\dots\rangle$
<code>\boldsymbol , \boldsymbol , \boldsymbol , \boldsymbol </code>	$ ,  ,   ,   $
<code>\boldsymbol\vert, \boldsymbol\vert, \boldsymbol\Vert, \boldsymbol\Vert</code>	$ ,  ,   ,   $

Однако, после `\left` and `\right` использовать `\boldsymbol` нельзя. В частности, введя `\left\boldsymbol| ... \right\boldsymbol|`, вы получите только сообщение об ошибке.

- Некоторые символы из жирных шрифтов через `\boldkey` и `\boldsymbol` вообще получить нельзя. К ним относятся жирные варианты  $\mathcal{A}, \dots, \mathcal{Z}$  “каллиграфических” (или “рукописных”) букв  $\mathcal{A}, \dots, \mathcal{Z}$ , которые задаются как `\Cal A, \dots, \Cal Z`, и жирные варианты  $\mathbf{o}, \dots, \mathbf{9}$  чисел в старом стиле  $\mathbf{o}, \dots, \mathbf{9}$ , которые можно получить командой `\oldnos`. Если вам действительно необходимы такие символы, вам следует призвать на помощь Т<sub>Е</sub>Xнолога или использовать “жирный шрифт для бедных” `\pmb` (см. ниже).

*Рукописный шрифт.* Как уже упоминалось, имеется специальное семейство “каллиграфических” (или “рукописных”) букв  $\mathcal{A}, \dots, \mathcal{Z}$ , которые задаются как `\Cal A, \dots, \Cal Z`. В этом семействе присутствуют только прописные буквы (об этом напоминает прописная  $\mathcal{C}$  в `\Cal`). Для получения единственной строчной рукописной буквы  $\ell$  используется команда `\ell`. Рукописный шрифт можно использовать только в математической моде.

*Шрифт Fraktur.* Доступ к готическому шрифту Fraktur, который создан для использования только в математической моде, можно получить, поместив в области преамбулы документа команду `\loadeufr`. Если вы пользуетесь стилем `preprint`, то Fraktur средней толщины загружается автоматически. Чтобы получить буквы из шрифта Fraktur, надо ввести

<code>\frac g</code>	$\mathfrak{g}$
<code>\frac A\$, \dots, \frac Z\$</code>	$\mathfrak{A}, \dots, \mathfrak{Z}$

*Жирный ажурный шрифт.*  $\mathcal{A}\mathcal{M}\mathcal{S}$ -Т<sub>Е</sub>X имеет “жирный ажурный” шрифт `\Bbb`. Как и `\Cal`, он работает только в математической моде и только с прописными буквами. Его буквы являются частью шрифта `msbm`, и доступ к ним можно получить, поместив в начало входного файла команду `\loadmsbm`. (В стиле `preprint` он загружается автоматически.)

<code>\Bbb A, \Bbb C, \Bbb R\$, ...</code>	$\mathbb{A}, \mathbb{C}, \mathbb{R}, \dots$
--	---

*Жирный шрифт “для бедных”.* Сейчас у  $\mathcal{A}\mathcal{M}\mathcal{S}$ -Т<sub>Е</sub>X’а есть жирные версии большинства математических символов. Если, однако, вам нужен только несколько жирных символов и загрузка новых шрифтов или их семейств выводит вас за рамки возможностей Т<sub>Е</sub>X’а, командой `\pmb` можно получить “бедняцкие” варианты жирных символов:

<code>\pmb{\Bbb A}\pmb{\geq}\bold x\$</code>	$\mathbb{A} \geq x$
--	---------------------

Версия `\pmb` бинарной операции или отношения также является, соответственно, бинарной операцией или отношением. Но для получения более “толстого” большого оператора придется обратиться к Т<sub>Е</sub>Xнологу.

*Символы с акцентами.* Над жирными буквами обычно ставятся обычные акценты:

<code>\hat{\bold x}\$</code>	$\hat{x}$
------------------------------	-----------

Но если все же хочется получить жирный акцент, то следует ввести

<code>\bold{\hat x}\$</code>	$\hat{x}$
------------------------------	-----------

Большинство специальных математических шрифтов не имеют своих собственных акцентов, так что в результате только что описанной конструкции получится обычный акцент. Так, например

<code>\Cal{\hat x}\$</code>	$\hat{\mathcal{x}}$
-----------------------------	---------------------

поскольку шрифт `\Cal` не имеет своего собственного акцента `\hat`.

*Другие шрифты в математической моде.* Кроме `\bold`, в математической моде можно использовать `\roman` (для того, чтобы буква в формуле получилась в прямом начертании), `\slanted` и `\italic` (позволяющий получать буквы текстовым, а не математическим курсивом). Но, в отличие от `\bold`, буквы, к которым применены `\slanted` или `\italic`, не меняют своей величины в индексах.

## 5. ИМЕНА ДОПОЛНИТЕЛЬНЫХ СИМВОЛОВ

Как показано ниже, символам из шрифтов `msam` и `msbm` присвоены “стандартные” имена командных последовательностей. В стиле препринт все эти имена символов загружаются автоматически. Если вы не используете стиль препринт, тот же эффект достигается командой `\UseAMSsymbols`. Это добавляет во внутреннюю таблицу `TeX`’а около 200 новых командных последовательностей. Если у вас ограничена память или вам нужны только несколько таких символов, то для их получения есть различные способы. См. ниже раздел **Команда `\newsymbol`**.

### 5.1. Специальные символы и жирные ажурные буквы

Некоторые символы из семейства `msam` можно задавать командными последовательностями, которые становятся определенными после появления в файле команды `\loadmsam`.

Во-первых, это четыре символа, которые обычно используются вне математической моды:

✓ <code>\checkmark</code>	® <code>\circledR</code>
✕ <code>\maltese</code>	¥ <code>\yen</code>

Такие символы, как ¶, §, †, and ‡, можно использовать и в математической моде, причем их размер в индексах первого и второго порядка будет изменяться.

Следующие четыре символа являются “ограничителями” (хотя у них и нет увеличенных версий, получаемых командами `\left` и `\right`), поэтому они должны быть использованы в математической моде:

┌ <code>\ulcorner</code>	┐ <code>\urcorner</code>
└ <code>\llcorner</code>	┘ <code>\lrcorner</code>

И наконец, из символов этого семейства сконструированы две пунктирные стрелки. Заметим, что одна из них имеет два имени и получить ее можно по любому из них:

<code>--&gt; \dashrightarrow, \dasharrow</code>	<code>&lt;-- \dashleftarrow</code>
---	------------------------------------

Жирные ажурные буквы  $\mathbb{A}, \dots, \mathbb{Z}$  имеются в семействе `msbm`. После того, как в файле встретилась команда `\loadmsbm`, их можно задавать (в математической моде) как `\Bbb A, \dots, \Bbb Z`.

Семейство `msbm` также содержит широкие версии `\widehat` и `\widetilde`.

### 5.2. Команда `\newsymbol`

Всем символам шрифтов `msam` и `msbm` должны соответствовать имена командных последовательностей, с помощью которых (после загрузки шрифтов) их можно задавать (только в математической моде). Это делается инструкцией `\UseAMSsymbols`, которая находится в файле `AMSSYM.TEX`. Эта инструкция включена в стиль препринт, так что имена присваиваются автоматически, и это требует около 200 командных последовательностей.

Если вам мало памяти для имен командных последовательностей, и нужны только несколько из этих символов, вы можете опустить `\UseAMSsymbols`. Вместо этого присвойте имена только тем символам, которые вам нужны. Для создания командной последовательности, которая производит нужный вам символ, используйте новую командную последовательность `AMS-TeX`’а `\newsymbol`. Имя этой командной последовательности может быть либо “стандартным”, из перечисленных ниже, либо любым по вашему выбору.

В приведенном ниже списке указан сам символ, его четырехзначный код “ID,” и “стандартное” имя символа. (Первый символ в ID указывает на семейство шрифтов, в котором этот символ проживает. Символы из семейства `msam` первым символом имеют 1, а символы из семейства `msbm` имеют первым символом 2.) Например, символ  $\not\leq$  представлен как

$\not\leq$  230A `\nleqslant`

Чтобы получить командную последовательность с этим именем, в `AMSSYM.TEX` надо поместить такую инструкцию:

```
\newsymbol\nleqslant 230A
```

Такая же инструкция может быть введена пользователем, который не использует стиль препринт и решил не загружать все имена символов командой `\UseAMSsymbols`. После этого командная последовательность `\nleqslant` будет производить символ  $\nleqslant$  (в математической моде), который будет действовать как “бинарное отношение”.

Несколько символов из этих шрифтов заменяют символы, которые в `PLAIN.TEX` определены, как комбинация символов из шрифтов Computer Modern. Это символы `\angle` ( $\angle$ ) и `\hbar` ( $\hbar$ ) из группы “Смешанные символы”, а также `\rightleftharpoons` ( $\rightleftharpoons$ ) из группы “Стрелки” ниже. Новые символы будут правильно изменять размер в индексах, при условии, что вы используете подходящее определение. Для того, чтобы с помощью `\newsymbol` заменить существующее определение, имя сначала должно быть сделано “неопределенным”. Ниже приведены строки, которые вы должны поместить в свой входной файл, если вы не используете стиль препринт или команду `\UseAMSsymbols` (которая делает переопределения автоматически):

```
\undefine\angle
\newsymbol\angle 105C
\undefine\hbar
\newsymbol\hbar 207E
\undefine\rightleftharpoons
\newsymbol\rightleftharpoons 130A
```

Такие символы в приведенной ниже таблице помечены значком “(U)”, что означает, что они сначала должны быть сделаны неопределенными.

Обратите внимание, что в этой таблице некоторые символы приведены с двумя именами. В таких случаях символ можно получать с помощью любого из этих имен.

### 5.3. Таблица символов

#### • Строчные греческие буквы

$\digamma$  207A `\digamma`  $\varkappa$  207B `\varkappa`

#### • Еврейские буквы

$\beth$  2069 `\beth`  $\gimel$  206A `\gimel`  
 $\daleth$  206B `\daleth`

#### • Смешанные символы

$\hbar$ 207E <code>\hbar</code> (U)	$\backprime$ 2038 <code>\backprime</code>
$\hslash$ 207D <code>\hslash</code>	$\varnothing$ 203F <code>\varnothing</code>
$\vartriangle$ 234D <code>\vartriangle</code>	$\blacktriangle$ 204E <code>\blacktriangle</code>
$\triangledown$ 204F <code>\triangledown</code>	$\blacktriangledown$ 2048 <code>\blacktriangledown</code>
$\square$ 2003 <code>\square</code>	$\blacksquare$ 2004 <code>\blacksquare</code>
$\lozenge$ 2006 <code>\lozenge</code>	$\blacklozenge$ 2007 <code>\blacklozenge</code>
$\circledS$ 2073 <code>\circledS</code>	$\bigstar$ 2046 <code>\bigstar</code>
$\angle$ 205C <code>\angle</code> (U)	$\sphericalangle$ 205E <code>\sphericalangle</code>
$\measuredangle$ 205D <code>\measuredangle</code>	
$\nexists$ 2040 <code>\nexists</code>	$\complement$ 207B <code>\complement</code>
$\mho$ 2066 <code>\mho</code>	$\eth$ 2067 <code>\eth</code>
$\Finv$ 2060 <code>\Finv</code>	$\diagup$ 201E <code>\diagup</code>
$\Game$ 2061 <code>\Game</code>	$\diagdown$ 201F <code>\diagdown</code>
$\Bbbk$ 207C <code>\Bbbk</code>	

#### • Бинарные операторы

$\dotplus$ 2275 <code>\dotplus</code>	$\ltimes$ 226E <code>\ltimes</code>
$\smallsetminus$ 2272 <code>\smallsetminus</code>	$\rtimes$ 226F <code>\rtimes</code>
$\Cap$ 2265 <code>\Cap</code> , <code>\doublecap</code>	$\leftthreetimes$ 2268 <code>\leftthreetimes</code>

☉	2264	\Cup, \doublecup	∠	2269	\rightthreetimes
∧	225A	\barwedge	∩	2266	\curlywedge
∨	2259	\veebar	∪	2267	\curlyvee
∧	225B	\doublebarwedge	⊖	227F	\circleddash
⊖	220C	\boxminus	⊗	227E	\circledast
⊗	2202	\boxtimes	⊙	227D	\circledcirc
⊙	2200	\boxdot	•	2205	\centerdot
⊕	2201	\boxplus	‡	227C	\intercal
*	223E	\divideontimes			

### • Бинарные отношения

≠	2335	\leqq	≧	233D	\geqq
≠	2336	\leqslant	≧	233E	\geqslant
≠	2330	\eqslantless	≧	2331	\eqslantgtr
≠	232E	\lesssim	≧	2326	\gtrsim
≠	232F	\lessapprox	≧	2327	\gtrapprox
≠	2375	\approxeq	▷	236D	\gtrdot
≠	236C	\lessdot	≧	236F	\ggg, \gggtr
≠	236E	\lll, \llless	≧	233F	\gtrless
≠	2337	\lessgtr	≧	2352	\gtreqless
≠	2351	\lesseqgtr	≧	2354	\gtreqqless
≠	2353	\lesseqqgtr	≧	2350	\eqcirc
≠	232B	\doteqdot, \Doteq	≧	2324	\circeq
≠	233A	\risingdotseq	≧	232C	\triangleq
≠	233B	\fallingdotseq	~	2373	\thicksim
∩	2376	\backsim	≈	2374	\thickapprox
∩	2377	\backsimeq	⊃	236B	\supseteq
⊂	236A	\subseteq	⊃	2363	\Supset
⊂	2362	\Subset	⊂	2341	\sqsupset
⊂	2340	\sqsubset	∩	233C	\succcurlyeq
∩	2334	\preccurlyeq	∩	2333	\curlyeqsucc
∩	2332	\curlyeqprec	∩	2325	\succsim
∩	232D	\prec	∩	2376	\succapprox
∩	2377	\precapprox	▷	2342	\vartriangleright
∩	2343	\vartriangleleft	▷	2344	\trianglerighteq
∩	2345	\trianglelefteq	∩	230D	\Vdash
∩	230F	\vdash			
∩	230E	\Vdash		2370	\shortmid
∩	2360	\smallsmile		2371	\shortparallel
∩	2361	\smallfrown	∩	2347	\between
∩	236C	\bumpeq	∩	2374	\pitchfork
∩	236D	\Bumpeq	∩	237F	\backepsilon
∩	235F	\varpropto	▶	2349	\blacktriangleright
◀	234A	\blacktriangleleft	∴	232A	\because
∴	2329	\therefore			

### • Отрицательные отношения

∩	2304	\nless	∩	2305	\ngtr
∩	2302	\nleq	∩	2303	\ngeq
∩	230A	\nleqslant	∩	230B	\ngeqslant
∩	2314	\nleqq	∩	2315	\ngeqq
∩	230C	\lneq	∩	230D	\gneq
∩	2308	\lneqq	∩	2309	\gneqq

$\nabla$	2300	<code>\lvertneqq</code>	$\nabla$	2301	<code>\gvertneqq</code>
$\simeq$	2312	<code>\lnsim</code>	$\simeq$	2313	<code>\gnsim</code>
$\approx$	231A	<code>\lnapprox</code>	$\approx$	231B	<code>\gnapprox</code>
$\prec$	2306	<code>\nprec</code>	$\succ$	2307	<code>\nsucc</code>
$\preceq$	230E	<code>\npreceq</code>	$\succeq$	230F	<code>\nsucceq</code>
$\precneqq$	2316	<code>\precneqq</code>	$\succneqq$	2317	<code>\succneqq</code>
$\precnsim$	2310	<code>\precnsim</code>	$\succnsim$	2311	<code>\succnsim</code>
$\precnapprox$	2318	<code>\precnapprox</code>	$\succnapprox$	2319	<code>\succnapprox</code>
$\sim$	231C	<code>\nsim</code>	$\cong$	231D	<code>\ncong</code>
$\shortmid$	232E	<code>\nshortmid</code>	$\parallel$	232F	<code>\nshortparallel</code>
$\mid$	232D	<code>\nmid</code>	$\parallel$	232C	<code>\nparallel</code>
$\dashv$	2330	<code>\nvdash</code>	$\dashv$	2332	<code>\nvDash</code>
$\Vdash$	2331	<code>\nVDash</code>	$\Vdash$	2333	<code>\nVDash</code>
$\triangleleft$	2336	<code>\ntriangleleft</code>	$\triangleright$	2337	<code>\ntriangleright</code>
$\trianglelefteq$	2335	<code>\ntrianglelefteq</code>	$\trianglerighteq$	2334	<code>\ntrianglerighteq</code>
$\subseteq$	232A	<code>\nsubseteq</code>	$\supseteq$	232B	<code>\nsupseteq</code>
$\subseteqq$	2322	<code>\nsubseteqq</code>	$\supseteqq$	2323	<code>\nsupseteqq</code>
$\subsetneq$	2328	<code>\subsetneq</code>	$\supsetneq$	2329	<code>\supsetneq</code>
$\varsubsetneq$	2320	<code>\varsubsetneq</code>	$\varsupsetneq$	2321	<code>\varsupsetneq</code>
$\subsetneqq$	2324	<code>\subsetneqq</code>	$\supsetneqq$	2325	<code>\supsetneqq</code>
$\varsubsetneqq$	2326	<code>\varsubsetneqq</code>	$\varsupsetneqq$	2327	<code>\varsupsetneqq</code>

• Стрелки

$\Leftrightarrow$	2312	<code>\leftleftarrows</code>	$\Rrightarrow$	2313	<code>\rightrightarrows</code>
$\Leftrightarrow$	231C	<code>\leftrightharpoons</code>	$\Leftrightarrow$	231D	<code>\rightleftarrows</code>
$\Leftarrow$	2357	<code>\Lleftarrow</code>	$\Rightarrow$	2356	<code>\Rrightarrow</code>
$\twoheadleftarrow$	2311	<code>\twoheadleftarrow</code>	$\twoheadrightarrow$	2310	<code>\twoheadrightarrow</code>
$\leftarrowtail$	231B	<code>\leftarrowtail</code>	$\rightarrowtail$	231A	<code>\rightarrowtail</code>
$\looparrowleft$	2322	<code>\looparrowleft</code>	$\looparrowright$	2323	<code>\looparrowright</code>
$\leftrightharpoons$	230B	<code>\leftrightharpoons</code>	$\rightleftharpoons$	230A	<code>\rightleftharpoons</code> (U)
$\curvearrowleft$	2378	<code>\curvearrowleft</code>	$\curvearrowright$	2379	<code>\curvearrowright</code>
$\circlearrowleft$	2309	<code>\circlearrowleft</code>	$\circlearrowright$	2308	<code>\circlearrowright</code>
$\Lsh$	231E	<code>\Lsh</code>	$\Rsh$	231F	<code>\Rsh</code>
$\upuparrows$	2314	<code>\upuparrows</code>	$\downdownarrows$	2315	<code>\downdownarrows</code>
$\upharpoonleft$	2318	<code>\upharpoonleft</code>	$\upharpoonright$	2316	<code>\upharpoonright</code> , <code>\restriction</code>
$\downharpoonleft$	2319	<code>\downharpoonleft</code>	$\downharpoonright$	2317	<code>\downharpoonright</code>
$\multimap$	2328	<code>\multimap</code>	$\rightsquigarrow$	2320	<code>\rightsquigarrow</code>
$\leftrightsquigarrow$	2321	<code>\leftrightsquigarrow</code>			

• Отрицающие стрелки

$\nleftarrow$	2338	<code>\nleftarrow</code>	$\nrightarrow$	2339	<code>\nrightarrow</code>
$\nLeftarrow$	233A	<code>\nLeftarrow</code>	$\nRightarrow$	233B	<code>\nRightarrow</code>
$\nleftrightarrow$	233D	<code>\nleftrightarrow</code>	$\nLeftrightarrow$	233C	<code>\nLeftrightarrow</code>

## 6. СРЕДСТВА ФОРМАТИРОВАНИЯ

Форматирование документов, подготовленных в  $\mathcal{A}\mathcal{M}\mathcal{S}$ - $\mathcal{T}\mathcal{E}\mathcal{X}$ , выполняется “стилевым файлом”. Представленные здесь средства являются частью стиля “препринт” (или, что то же самое, `amsprt`), разработанного Американским математическим обществом (AMS). Стиль  $\mathcal{A}\mathcal{M}\mathcal{S}$ - $\mathcal{T}\mathcal{E}\mathcal{X}$ ’а “препринт”, версия 2.1, будет форматировать входной файл в виде, принятом для статьи в журнале *Journal of the American Mathematical Society*, полиграфические особенности которого он наиболее полно отражает. Предполагается, что входной файл содержит статью, а не главу книги. При форматировании книги следует явно указать стиль `\Monograph` (см. ниже раздел **Форматирование книги**). Далее описывается журнальный стиль, в противном случае об этом будет сказано особо.

### 6.1. Структура входного файла

Входной файл в  $\mathcal{A}\mathcal{M}\mathcal{S}$ - $\mathcal{T}\mathcal{E}\mathcal{X}$ ’е выглядит как-то так:

```
\input amstex
\documentstyle{...}

<команды преамбулы, такие как \define, \pageno, \Monograph,
  \NoRunningHeads, \loadbold, и т.д.>

\topmatter
...
\endtopmatter
\document
<тело статьи>
\enddocument
```

Если вы собираетесь использовать стиль препринт, то это надо указать так: `\documentstyle{amsprt}`. Этот стиль задает длину строки равной 30 pica, а длину полосы набора — 47 pica. Предполагается, что у каждой страницы, за исключением первой, будет колонтитул: на четных страницах будет номер страницы и название статьи, а на нечетных — номер страницы и авторы.

Некоторые команды влияют на внешний вид всего документа. Такие команды должны находиться в самом начале входного файла сразу после строки `\documentstyle` и перед строкой `\topmatter`. Эту область обычно называют “преамбулой”. В преамбуле располагаются команды, определяющие новые управляющие последовательности, типа `\define`, команды, меняющие стиль нумерации формул типа `\TagsOnRight`, такие команды, как `\NoPageNumbers`, `\NoRunningHeads`, `\Monograph`, команды, меняющие размер страниц и их нумерацию `\pagewidth`, `\pageheight`, `\pageno`, а также команды загрузки шрифтов.

Если у вас есть документы, подготовленные на версиях  $\mathcal{A}\mathcal{M}\mathcal{S}$ - $\mathcal{T}\mathcal{E}\mathcal{X}$ , более ранних, чем версия 2.0, вам пригодится файл `amsprt1.tex`. Если сразу после строки `\documentstyle` поместить строку `\input amsppt1`, команды из области `\topmatter` и команды рубрикации будут преобразованы в такой вид, чтобы они работали первоначальным способом. Кроме этой, должно быть не особенно много проблем с несовместимостью предыдущих версий. Замечание: не следует использовать `amsprt1.tex`, кроме как для обработки ранее созданных файлов.

Преамбула может быть у любого документа — как у статьи, так и у монографии.

### 6.2. Область `\topmatter`

После преамбулы во входном файле следует область, в которой содержатся название работы, авторы, дата написания, адреса и т.д., то есть сведения об издании — это так называемая область `\topmatter ... \endtopmatter`. Данные из этой области могут появиться в начале титульной страницы, в конце ее или даже в конце всей работы. Область `\topmatter` выглядит как-то так:

```
\topmatter
```

```

\title...\endtitle
\author...\endauthor
\affil...\endaffil
\address...\endaddress
\curraddr...\endcurraddr
\email...\endemail
\dedicatory...\enddedicatory
\date...\enddate
\thanks...\endthanks
\translator...\endtranslator
\keywords...\endkeywords
\subjclass...\endsubjclass
\abstract...\endabstract
\endtopmatter
\document

```

Все пары управляющих последовательностей, стоящих в области `\topmatter`, являются необязательными и могут набираться в любом порядке. То, как именно должно задаваться то или иное вхождение в `\topmatter`, зависит от конкретного издательства. AMS, например, требует, чтобы в материалах для публикации были представлены заголовок (`\title`), автор (`\author`), адрес (`\address`), тематика (`\subjclass`), а для публикаций в журнале — аннотация (`\abstract`).

Если пропущена (или введена с ошибкой) какая-нибудь из команд `\end...`, то на строку `\endtopmatter` или следующую пустую строку будет выдано сообщение об ошибке: например, если вы неправильно ввели `\endtitle`, будет выдано сообщение вроде “! Paragraph ended before \title was complete.” Если вы пропустили `\endtopmatter`, сообщение об ошибке не появится, но никаких данных из области `\topmatter` напечатано не будет.

В многострочных заголовках, в указаниях членства, в авторах или в посвящениях (т. е. везде, где строки центрируются, а не оформляются в виде абзацев), конец строки обозначается командой `\`. В других частях области `\topmatter`, которые оформляются в виде абзацев, конец строки указывается командой `\linebreak`.

*Заголовок статьи.* Заголовок будет набираться прописными буквами. Чтобы отключить автоматический переход на прописные буквы, надо использовать опцию `\nofrills: \title\nofrills...\endtitle`. Авторам следует самим позаботиться, чтобы заголовок был красиво разбит на строки, используя для обозначения конца строки команду `\`.

*Адреса.* Адреса электронной почты включаются командами `\email`. Перед каждым `\email`-адресом должен стоять обычный (почтовый) `\address`, в противном случае e-mail-адрес напечатан не будет. Можно использовать несколько `\email`-адресов, но каждый `\email ... \endemail` должен быть в паре с `\address` того же автора. `\email`-адрес будет напечатан в конце документа после парного ему почтового адреса в виде “*E-mail address: <сетевой адрес>*”. При вводе почтового адреса не допускается вводить `\` для указания разбивок на строки. Следует просто вводить его обычной строкой, а разбивка на строки будет сделана автоматически.

Обычно адрес, заданный в `\address` — это адрес автора в тот момент, когда им выполнена научная работа; если во время публикации адрес автора изменился, текущий адрес должен быть указан в `\curraddr`. В файле документа эта команда должна помещаться между `\address` и `\email`. Как и в случае `\email`, можно использовать несколько `\curraddr`, если каждый из них включается после `\address` автора, которому он принадлежит. Если нет предшествующего `\address`, текущий адрес напечатан не будет.

*Дополнительная информация.* Команда `\dedicatory` используется для посвящений типа “Посвящается профессору X. в связи с его восьмидесятилетием.” Посвящение будет напечатано курсивом перед аннотацией.

Команда `\thanks` предназначена для благодарностей за финансовую и другую поддержку научной работы автора или другой общей информации, которая не была включена в другие команды, такие,

как `\keywords` или `\subclass`. Информация будет напечатана внизу первой страницы в виде сноски без номера. Как `\address`, так и `\thanks` можно использовать более одного раза.

В случае статьи предусмотрен не только автор, но и переводчик, который задается в `\translator`. Эта информация будет напечатана в конце статьи романским шрифтом в 8 пунктов в виде “Translated by” и следующим за ним именем переводчика, набранного прописными буквами.

Информация из `\keywords` и `\subclass` появляется внизу первой страницы в виде нумерованной сноски, так же, как в журналах AMS. В главах монографий она вовсе не будет напечатана, поскольку должна быть обработана отдельно как часть фронтального материала всей монографии.

*Аннотация.* Аннотация печатается капителью в размере восемь пунктов с отступами от левого и правого поля. Заголовок аннотации набирать не надо. Он печатается автоматически тем же шрифтом в виде “АБСТРАКТ.” Если вас это не устраивает, то, как всегда, такую автоматику можно отменить командой `\nofrills`. Например, если вам нужно, чтобы вместо “АБСТРАКТ.” было напечатано “АННОТАЦИЯ”, то нужно ввести

```
\abstract\nofrills АННОТАЦИЯ. Текст
аннотации.\endabstract
```

и тогда получится

```
АННОТАЦИЯ. Текст аннотации.
```

Обратите внимание, что если вы примените `\nofrills`, то по умолчанию будет использован обычный романский шрифт размером в 8 пунктов. Если нужен другой шрифт, его следует указать явно.

*Оглавление.* Можно получать простое оглавление. Оглавление статьи задается в области `\topmatter` наряду со всем остальным посредством управляющих последовательностей `\toc... \endtoc`. (Оглавление книги получается несколько иначе — см. ниже раздел **Форматирование книги**). Команда `\toc` автоматически выдаст центрированный заголовок “CONTENTS”. Если вам нужен другой заголовок, то, как всегда, можно применить команду `\nofrills`. Например, если вам нужно, чтобы вместо “CONTENTS” было напечатано “Содержание”, то нужно ввести

```
\toc\nofrills{\bf Содержание}
...
\endtoc
```

Далее в содержании приводятся разделы и подразделы статьи:

```
\toc
\specialhead... \endspecialhead
\head... \endhead
\subhead... \endsubhead
\subsubhead... \endsubsubhead
\endtoc
```

Синтаксис названий разделов идентичен синтаксису, используемому для заголовков внутри документа (см. ниже разделы **Заголовки** и **Форматирование книги**), так что те, кто хочет этого и имеет подходящий текстовый редактор, могут создавать оглавление, выбирая из основного текста соответствующие строки.<sup>4</sup>

В коротком оглавлении для журнальной статьи обычно не требуются номера страниц, но по желанию их можно вставить так же, как и в оглавлении монографии; см. ниже раздел **Форматирование книги**.

В оглавлении в начале строк `\head` и `\subhead` делается отступ, величина которого достаточна, чтобы поместились номера, соответственно, в виде “1.” и “1.1.”; величину отступа можно регулировать, используя `\widestnumber`:

<sup>4</sup>Только обратите внимание, что оригинальное разбиение строк в многострочных заголовках может оказаться не подходящим для оглавления, так что, может быть, вам следует убрать некоторые из присутствующих `\`.

```

\toc
\widestnumber\head{10}
\widestnumber\subhead{10.1}
...

```

По желанию это можно делать несколько раз внутри различных разделов оглавления.

Если оказывается, что “номер раздела ” из `\head` включает в себя что-то вроде “Приложение” (как в этом руководстве), перед ним должна стоять пара “пустых” фигурных скобок:

```
\head {} Приложение. Пример библиографии --- входные ...\endhead
```

Вставка в начале текста заголовка `{}` с последующим пробелом указывает, что заголовок в целом будет прижат вплотную влево. Если в содержание входит несколько разных по уровню значимости заголовков, можно их внешне несколько отделить друг от друга, если сразу после `\toc` поместить команду

```
\openup.5\baselineskip
```

Размер отделяющего пространства может быть не только `.5\baselineskip`, но также любым допустимым в  $\TeX$ 'е размером.

Например, оглавление этого руководства было получено командами:

```

\toc\nofrills{\smc Содержание}
\openup.5\baselineskip
\widestnumber\head{100}
\widestnumber\subhead{10.10}
\head 1. Введение \endhead
\subhead 1.1. Файлы, входящие в пакет  $\LaTeX$  версии~2.1
\endsubhead
\subhead 1.2. Об этом руководстве \endsubhead
\head 2. Подготовка текста в  $\TeX$ 'е \endhead
...
\head 10. Выражение признательности за использование  $\LaTeX$ 
\endhead
\head {} Приложение А. Пример получения библиографии\endhead
\head {} Приложение Б. Макеты шрифтов коллекции AMSFonts
\endhead
\head {} Библиография \endhead
\endtoc

```

Если вы готовите библиографию, формат и содержание `\topmatter` будет другим. Подробности приведены ниже в разделе **Форматирование книги**.

### 6.3. Форматирование книги

Если вы готовите монографию, в стиле препринт есть средства, благодаря которым напечатанное вами будет выглядеть, как глава книги, а не как отдельная статья. Каждая глава готовится отдельно и по отдельности прогоняется через  $\LaTeX$ .

Прежде всего, вам следует объявить о своих намерениях, задав в преамбуле сразу после строки `\documentstyle` команду `\Monograph`.

*Заголовок главы.* В главе монографии типичный раздел `\topmatter` должен быть набран таким образом:

```

\documentstyle{amspt}
\Monograph
\topmatter
\title\chapter{4}Алгебра матриц\endtitle
\endtopmatter

```

что выдаст следующий заголовок главы:

CHAPTER IV

## АЛГЕБРА МАТРИЦ

Обратите внимание что автоматически номер главы преобразуется в римское число и добавляется слово “CHAPTER”. Если заголовок главы следует печатать в другом виде, надо использовать команду `\nofrills`:

```
\topmatter
\title\chapter\nofrills {ПРИЛОЖЕНИЕ D} Интеграл Пуассона\endtitle
\endtopmatter
```

Это даст

ПРИЛОЖЕНИЕ D

## ИНТЕГРАЛ ПУАССОНА

Замененный текст `\chapter` появится точно в том виде, как набран.

И наконец, для таких разделов, как введение или предисловие, не имеющих предваряющего текста, опустите команду `\chapter`:

```
\topmatter
\title Предисловие\endtitle
\endtopmatter
```

*Оглавление книги.* В монографиях оглавление обычно рассматривается, как отдельная глава. Ее надо начать заголовком “Оглавление”, как для предисловия или введения, а затем использовать структуру `\toc... \endtoc`, как тело документа (а не помещать ее в область `\topmatter`, как вы бы сделали в журнальной статье).

```
\topmatter
\title Оглавление\endtitle
\endtopmatter

\document
\toc
\title Предисловие\page{vii}\endtitle
\title\chapter{1}Алгебра матриц\page{1}\endtitle
\head {} Непрерывные комплексные функции\page{1}\endhead
...
\title Библиография\page{307}\endtitle
\endtoc
\enddocument
```

Заголовки глав, перечисленные в оглавлении, набираются в том самом виде, в котором они набраны в основном тексте книги. Как показано выше, чтобы получить номера страниц, непосредственно перед концом элемента следует использовать `\page`. Эта возможность имеется на всех уровнях заголовков.

*Колонтитулы.* В монографии, подготовленной в стиле препринт, для левого колонтитула используется название главы, а для правого — текст из заголовка секции (из `\head`). Нередко текст заголовка оказывается слишком длинным, чтобы поместиться в область, отведенную для колонтитула; в этом случае для задания сокращенной формы заголовка, которая будет использоваться в колонтитулах, следует использовать `\rightheadtext`:

```
\head Коэффициенты Фурье непрерывных периодических функций
```

```
с ограниченной энтропией\endhead
\righttheadtext{Коэффициенты Фурье периодических функций}
```

Сокращенный заголовок должен следовать сразу после `\head`, для гарантии того, что оба будут действовать на одной и той же странице. Если слишком длинным, чтобы поместиться в левом колонтитуле, оказывается название главы, то точно так же можно использовать его сокращенную форму, помещая `\lefttheadtext` сразу после `\title`. См. также раздел **Колонтитулы**.

*Стиль главы.* Стиль для главы монографии несколько отличается от стиля статьи. Текст из `\head` печатается не капителью, а жирным шрифтом, заголовки теорем, утверждений, определений и т.д. печатается не жирным шрифтом, а капителью, причем с абзацным отступом, а не прижатым вплотную к левому краю.

#### 6.4. Номера страниц

Если вы используете стиль препринт, то номера страниц печатаются в колонтитуле на внешнем поле, за исключением первой страница, у которой колонтитул пустой, а номер страницы печатается внизу в центре.

Если нужно опустить нумерацию страниц, поместите в начале документа команду `\NoPageNumbers` (после строки `\documentstyle`). Текст колонтитулов сохранится; см. также **Колонтитулы**.

Можно задать нумерацию страниц римскими цифрами, например, для оглавления и предисловия, используя для этого обычный прием  $\TeX$ 'а — команду `\pageno` вместе с отрицательным номером. Напомним, что вы можете сказать `\pageno=100`, если надо, чтобы следующая страница результата имела номер 100, а если рукопись начинается с `\pageno=-1`, номера страниц будут i, ii, iii, iv, v и т. д.

#### 6.5. Размер страницы

В стиле препринт по умолчанию приняты ширина страницы, равная 30pc, и высота страницы, равная 47.5pc. В высоту страницы включается и строка для номера. Можно изменить размеры страницы командами

```
\pagewidth{⟨размер⟩}
\pageheight{⟨размер⟩}
```

используя подходящие *⟨размер⟩*ы, где под этим обозначением мы подразумеваем допустимые размеры  $\TeX$ 'а, описанные в разделе **Размеры**.

Можно изменить и расположение страницы с вновь определенными размерами на стандартном листе бумаги. Команда

```
\hcorrection{⟨размер⟩}
```

передвинет страницу целиком по горизонтали вправо на *⟨размер⟩*, а команда

```
\vcorrection{⟨размер⟩}
```

опустит ее вниз на *⟨размер⟩*.

#### 6.6. Колонтитулы

Если вы используете стиль препринт, колонтитулы выглядят так, как в *Воспитательный  $\TeX$* : текст располагается в центре, а номер страницы на полях. (Как обычно, на первой странице колонтитул пустой, а номер страницы помещается внизу.)

Если вы ничего не предпринимали, чтобы определить текст колонтитулов, в колонтитулах левосторонних страниц будет печататься фамилия автора, а в колонтитулах правосторонних страниц — название. (Это касается стиля для статей; о монографиях см. ниже.) Если вам нужны какие-нибудь другие значения, скажем, сокращенное название, можно переопределить текст колонтитулов левосторонних и правосторонних страниц, введя

```
\lefttheadtext{⟨текст левого колонтитула⟩}
\righttheadtext{⟨текст правого колонтитула⟩}
```

Эти инструкции могут появиться в любом месте после команды `\documentstyle`, но чаще всего их помещают сразу после `\title`, `\author` или `\head`, чтобы перекрыть автоматически сформированные колонтитулы. Если `\rightheadtext` или `\leftheadtext` заданы над областью `topmatter`, то `\title` и `\author` их не перекрывают.

Если вы готовите не журнальную статью, а монографию и используете переключатель `\Monograph`, то это следующим образом действует на колонтитулы: в левом колонтитуле печатается название главы, а в правом — текст заголовка секции (из `\head`). В главах, которые не содержат никаких `\head`'ов — например, в предисловиях — как левый, так и правый колонтитулы будут содержать название главы.

По умолчанию колонтитулы печатаются прописными буквами. Это украшение можно отменить командой `\nofrills`:

```
\rightheadtext\nofrills {Текст колонтитула}
```

Если по каким-то причинам вы не хотите, чтобы печатались колонтитулы, поместите в начале документа `\NoRunningHeads` (после строки `\documentstyle`). Если колонтитулы опущены, номера страниц помещаются в середине строки внизу страницы (и даже могут быть отменены командой `\NoPageNumbers`.)

В монографии, если вы не хотите, чтобы текст из секции `\head` появился в колонтитулах, то вам следует переопределить внутреннюю команду `\headmark`, которая используется командой `\head` для задания правых колонтитулов. Чтобы сделать это, поместите в свой входной документ после `\Monograph` и перед `\topmatter` следующую строку:

```
\redefine\headmark#1{}
```

(где `#1` — это номер аргумента, как объясняется в описании `\define` и связанных с ней команд).

## 6.7. Библиографии

Библиографический раздел документа начинается с `\Refs`, а в конце должен иметь `\endRefs`. Каждый элемент библиографии начинается с `\ref` и оканчивается `\endref`. Между `\ref` и `\endref` конкретные элементы можно указывать в любом порядке. Однако, сразу после `\ref` обычно следует номер или другая метка, идентифицирующая этот элемент. Эта метка получается с помощью команды `\key`. Формат метки определяется текущим *стилем ссылок*, который задается командой `\refstyle`. Стиль документа препринт предусматривает три стиля ссылок, обозначаемых буквами А, В и С и соответствующих буквенным меткам, отсутствию меток и числовым меткам. Ниже указаны форма команд `\cite` и `\key` для каждого стиля и их результат:

<code>\refstyle{A}</code>		<code>\refstyle{B}</code>		<code>\refstyle{C}</code>	
<code>\cite{DK}</code>	[DK]	<code>\cite{Smith 1989}</code>	[Smith 1989]	<code>\cite{19}</code>	[19]
<code>\key DK</code>	[DK]	(нет ключа)	(нет метки)	<code>\key 19</code>	19.

Команда `\refstyle` обычно помещается в преамбуле документа.

Элементы библиографии обычно печатаются с подвешенным отступом. В общем случае, величина этого отступа такова, чтобы там поместилось двузначное число. Его можно увеличить (или уменьшить), указав самую широкую метку, используемую в списке библиографии. Например,

```
\widestnumber\key{GHMR}% стиль А
\widestnumber\key{999}% стиль С --- 3 цифры
```

увеличит отступ, чтобы поместить, соответственно, ключ [GHMR] или трехзначное число. Если библиография состоит из менее десяти элементов, можно уменьшить отступ, указав `\widestnumber\key{9}`. Как показано в примерах, при использовании `\widestnumber` не следует ставить квадратные скобки, точки, команды переключения шрифта или другие форматирующие команды. При создании отступа эти элементы учитываются автоматически. Если в библиографии идут подряд несколько работ одного автора/авторов, в первом из таких элементов используется `\by` со следующим за ним полностью указанным именем (или именами) автора, а в каждом последующем элементе ставится команда `\bysame` — только `\bysame` без повторения имени (имен). Команда `\bysame` производит горизонтальную черту фиксированной длины 3 em.

Для указания редакторов имеются два варианта, `\ed` и `\eds`, так же, как `\page` и `\pages`, поскольку “ed.” или “eds.” являются частью автоматического форматирования. Если `\by` отсутствует, вместо имени автора будет помещено имя редактора.

Для трудов конференции в поле `\procinfo` записывается место и дата ее проведения. При этом будут добавлены круглые скобки.

В конце элемента библиографии имеются две возможности для указания разных примечаний — это `\finalinfo` `\miscnote`. `\miscnote` отличается только тем, что автоматически добавляет круглые скобки; обычно используется для таких примечаний, как “(препринт)”, “(представлено)” или “(to appear)”. Так как последнее встречается чаще всего, для него имеется специальная команда `\toappear`, которая эквивалентна последовательности `\miscnote to appear`.

Если библиографическая информация представлена в переводе или есть сомнения в том, что будет правильно понято, на каком языке подготовлен документ, для указания языка оригинала следует использовать команду `\lang`.

Иногда несколько элементов библиографии группируются в один — например, части длинного документа, которые были изданы отдельно. Другой тип составных элементов — это работа, цитируемая как в оригинале, так и в переводе. Для обработки таких ситуаций имеются команды `\moreref` и `\transl`. После `\moreref` и `\transl` можно опять использовать любую из форматизирующих библиографию команд.

`\moreref` используется для цитирования, например, “часть II” статьи. После команды `\moreref` по желанию можно ставить другие библиографические команды и данные. Например:

```
... \moreref \paper \rom{II}
\jour Comm. Pure Appl. Math. \vol 36
\yr 1983 \pages 571--594 \endref
```

При использовании `\transl`, текст, который описывает перевод, обычно помещается между `\transl` и следующей командой. Далее следуют команды и данные для переведенной работы. Например:

```
... \transl English transl. \publ Birkh \auser
\publaddr Basel \yr 1985 \endref
```

Автоматическая пунктуация печататься не будет, если соответствующее поле включено, но оставлено пустым. В противном случае, чтобы автоматическая пунктуация не появлялась, следует использовать команду `\nofrills`. Например, `\bookinfo \nofrills...` подавляет запятую или другие знаки пунктуации, обычно добавляемые в конце информации `\bookinfo`. `\nofrills` также подавляет другое автоматическое форматирование, такое, как слово “eds.” для `\eds`, слово “vol.” для томов книги или круглые скобки вокруг года издания журнальной статьи. Конечную точку можно подавить командами `\finalinfo \nofrills`.

Использование этих команд проиллюстрировано на примерах: образец входных данных и результата приведен в Приложении А. Более подробно о библиографии написано в Приложении С книги *Воспитательный  $\TeX$* .

## 7. ОПРЕДЕЛЕНИЕ НОВЫХ КОМАНД

Для определения новых макрокоманд  $\mathcal{A}\mathcal{M}\mathcal{S}$ -TeX имеет команду `\define`, хотя по прежнему позволяет использовать и команду Plain TeX'a `\def` (см. *Все про TeX*). Например, новую управляющую последовательность `\ab` можно определить так:

```
\define\ab{\alpha^2+\beta^2}
```

после чего TeX станет подставлять `\alpha^2+\beta^2` всякий раз, как только встретит `\ab`, так что `\ab` выдаст формулу  $\alpha^2 + \beta^2$ , а `$$\ab.$$` — выключную формулу

$$\alpha^2 + \beta^2.$$

После `\define` должно стоять имя управляющей последовательности, как обычно начинающееся с бэкслэша, а далее в фигурных скобках следовать определению этой последовательности. Заключать имя определяемой последовательности в фигурные скобки нельзя! Если же определяется управляющий символ, то следует соблюдать осторожность и не оставлять между ним и открывающей фигурной скобкой никаких пробелов:

```
\define\1{\alpha^2+\beta^2}
```

С помощью `\define` присвоить уже известной команде с длинным именем новое имя, которое вам нравится больше. Так, если определить новую команду `\dra` так:

```
\define\dra{\dashrightarrow}
```

то команда `\dra` напечатает правую пунктирную стрелку  $\dashrightarrow$ .

Указанным образом можно определять только новые команды. Так, например, попытка определения

```
\define\d{\dashrightarrow}
```

приведет к сообщению об ошибке

```
! AmS-TeX error: \d is already define
```

поскольку уже существует команда `\d` (она печатает точку под буквой). Для того, чтобы справиться с такой ситуацией,  $\mathcal{A}\mathcal{M}\mathcal{S}$ -TeX предоставляет команду `\redefine`. Если поместить во входной файл такое определение:

```
\redefine\d{\dashrightarrow}
```

то все последующие `\d` будут заменены на `\dashrightarrow`, несмотря на то, что изначально `\d` имело другой смысл. Если при этом все еще хочется иногда ставить точку под буквой, то следует использовать такую конструкцию:

```
\predefine\dotunder{\d}
\redefine\d{\dashrightarrow}
```

Тогда управляющая последовательность, позволяющая ставить точку под буквой, получит имя `\dotunder`, а `\d` станет новым именем для `\dashrightarrow`. Обратите внимание, что в `\predefine` в фигурных скобках не должно быть ничего, кроме единственной управляющей последовательности.

При определении команд категорически запрещена рекурсия: попытка определить команду через саму себя приведет к сообщению о переполнении памяти TeX'a или к чему-нибудь еще худшему!

Определение новой команды необходимо поместить до того, как вы ею воспользуетесь. Часто все глобальные (т.е. действующие во всем документе) `\define` помещают в преамбуле входного файла, где-нибудь между строками `\documentstyle` и `\topmatter` или в начале текста сразу после строки `\document`.

Можно также вводить `\define` прямо в том месте, где это нужно. Однако следует помнить, что если поместить `\define` внутри группы или математической формулы, то с окончанием этой группы или формулы ее действие закончится, а определенная ею команда станет неопределенной. Если же поместить `\define` внутри абзаца, то есть вероятность получить лишние пробелы. Спивак

советует помещать такие `\define` на отдельной строке с `%` в конце ее, чтобы отменить действие `\carriage-return`.

Команда `\define` может иметь аргументы. Так, можно определить управляющую последовательность с одним аргументом `\deriv` для получения производных по  $x$ :

```
\define\deriv#1{\dfrac{d#1}{dx}}
```

Здесь `#1` обозначает аргумент, так что `\deriv f` заменяется на `\dfrac{df}{dx}` и `\deriv f` дает  $\frac{df}{dx}$ . Аналогично, `\deriv g` дает  $\frac{dg}{dx}$ , `\deriv z` дает  $\frac{dz}{dx}$  и т. д.

Приведем пример определения управляющей последовательности с двумя аргументами. Если в работе часто встречаются частные производные

$$\frac{\partial f}{\partial x}, \quad \frac{\partial f}{\partial y}, \quad \frac{\partial g}{\partial x}, \quad \frac{\partial h}{\partial z},$$

то для упрощения их набора можно определить управляющую последовательность `\pd` следующим образом:

```
\define\pd#1#2{\dfrac{\partial#1}{\partial#2}}
```

и приведенные выше частные производные будут легко получаться командами `$$\pd fx$$`, `$$\pd fy$$`, `$$\pd gx$$` и `$$\pd hz$$`. Обратите внимание, что пробел перед `#` в `\partial#1` и `\partial#2` не нужен, не смотря на то, что вместо `#1` и `#2` можно подставить буквы. После `\define`  $\LaTeX$  воспримет `\partial` и `#1` постоянно разделенными.

Можно определить управляющую последовательность, имеющую вплоть до девяти аргументов: `#1, #2 #3 ... #9`, причем если вы ввели `\define\cs`, где `\cs` есть управляющая последовательность, то аргументы после `\define\cs` должны следовать в строго указанном выше порядке, иначе  $\TeX$  выдаст сообщение об ошибке. Более того, между параметрами `#1` и `#2`, `#2` и `#3` и т. д. *не должно быть пробелов*, поскольку, например, если набрать

```
\define\pd#1 #2{\dfrac{\partial#1}{\partial#2}}
```

то  $\TeX$  подумает, что между первый и второй аргумент разделяются пробелами, и в команде

```
\pd fx =3 +x
```

первым аргументом будет считать `fx`, а вторым — `=3`! Представляете, что вы получите? Если же правильно определить `\pd` (без всяких пробелов между аргументами), то для получения частных производных можно пользоваться набирать как `$$\pd fx$$`, так и `$$\pd f x$$`, поскольку  $\TeX$  всегда игнорирует пробелы, когда ищет аргументы управляющей последовательности.

Если же вам нужен сложный аргумент, состоящий из более чем одного символа, просто заключите его в фигурные скобки.

## 8. ИСПРАВЛЕНИЕ ОШИБОК

## 8.1. Сообщения об ошибках

Ну вот, вы, как мы надеемся, внимательно прочитали наше руководство и можете начать самостоятельную работу. Вам может крупно повезти и вы никогда не будете делать никаких ошибок, но согласитесь, что вероятность этого крайне невысока и скорее всего вы уже видели реакцию Т<sub>Э</sub>X'а на свои ошибки — машина останавливается, а на экране появляется что-нибудь вроде:

```
! Undefined control sequence.
1.2 \vship
      1in
?
```

Т<sub>Э</sub>X начинает свои сообщения об ошибках с `—!` и, печатая две строки контекста, показывает, что было прочитано во время обнаружения ошибки. Верхняя часть этой пары (в данном случае `“\vship”`) показывает то, что Т<sub>Э</sub>X просмотрел до сих пор (1.2—это номер файла и номер строки), нижняя часть (в данном случае `“1in”`) показывает остаток входной строки.

Знак `?`, который появляется после контекста, означает, что Т<sub>Э</sub>X хочет получить совет, что делать дальше. Если вы перед этим никогда не видели сообщения об ошибках, или если вы забыли, какой ответ ожидается, то можете ввести `‘?’`; Т<sub>Э</sub>X откликнется следующим образом:

```
Type <return> to proceed, S to scroll future error messages,
R to run without stopping, Q to run quietly,
I to insert something, E to edit your file,
1 or ... or 9 to ignore the next 1 to 9 tokens of input,
H for help, X to quit.
```

Это перечень возможностей, а именно:

- (1) `<return>`: продолжай работу, попытавшись, насколько это можно, исправить ошибку.
- (2) `S`: продолжай работу, не спрашивая указаний, даже если далее встретятся ошибки. Последующие сообщения об ошибках промелькнут на вашем терминале, возможно, быстрее, чем вы успеете прочитать их, и появятся в протокольном файле, где на досуге их можно тщательно изучить. Таким образом, `S` — это то же самое, что отвечать `<return>` на каждое сообщение.
- (3) `R`: почти то же самое, что `S`, но сильнее, поскольку приказывает Т<sub>Э</sub>X'у не останавливаться ни по какой причине, даже если не найдено имя файла.
- (4) `Q`: похоже на `R`, но еще сильнее, поскольку приказывает Т<sub>Э</sub>X'у не только продолжать работу без остановки, но также подавляет весь дальнейший вывод на терминал. Это быстрый, но несколько безрассудный способ работы (предназначенный для работы Т<sub>Э</sub>X'а без присутствия человека).
- (5) `I`: вставь текст, следующий за `I`. Т<sub>Э</sub>X прочтет его, а затем продолжит чтение входной строки. Строки, вставленные таким способом, не должны оканчиваться пробелом.
- (6) Небольшое число (меньше 100): удали это количество символов и командных последовательностей из оставшейся части входной информации и остановись снова, чтобы дать еще один шанс посмотреть на то, что получилось.
- (7) `H`: помоги! Именно это вы должны сделать сейчас и делать всякий раз, когда сталкиваетесь с сообщением об ошибке, которое вы пока еще не видели. У Т<sub>Э</sub>X'а есть два сообщения, встроенных для того, чтобы ошибку мог понять каждый: одно формальное и одно неформальное. Формальное сообщение печатается первым (например, `‘!Undefined control sequence.’`); неформальное печатается, если вы, введя `‘H’`, запрашиваете дополнительную помощь, а также оно появляется в протокольном файле, если вы записываете туда сообщения об ошибках. Неформальное сообщение пытается дополнить формальное пояснением того, в чем, по мнению Т<sub>Э</sub>X'а, заключается неприятность, и часто советует стратегию для возмещения потерь.
- (8) `X`: кончай работу, прекратив обработку вашего файла и завершив протокольный файл; `dvi`-файл будет содержать все предыдущие страницы. Текущая (незавершенная) страница выведена не будет.

- (9) E: редактируй — похоже на X, но дополнительно подготавливает компьютер к редактированию файла, который  $\TeX$  в данный момент читает, в текущей позиции, так что вы можете со всеми удобствами сделать изменения перед следующим запуском  $\TeX$ 'а.

$\TeX$  умеет выдавать более сотни различных видов сообщений об ошибках, и вы, вероятно, никогда не столкнетесь со всеми из них, поскольку некоторые типы ошибок очень трудно сделать. Иногда, например, если ошибка встретилась при обработке команд высокого уровня, можно получить более сложный контекст сообщения об ошибке — не одну пару строк, а две.

Подробно работа над ошибками в  $\TeX$ 'е описана в книге Д. Кнута *Все про  $\TeX$* .

## 8.2. Переполнение или недогрузка боксов

Как уже говорилось,  $\TeX$  сам формирует строки и страницы выходного документа. Иногда же он попадает в затруднительное положение — из-за того, что он не может найти удачное место для разрыва строки, либо пробелы в строке получаются слишком широкими, либо “хвост” строки залезает на правое поле. В первом случае  $\TeX$  выдаст на терминал предупреждающее сообщение о недогрузке *горизонтального бокса* `Underfull \hbox` с указанием номера строки входного файла, а во втором — выдаст предупреждающее сообщение о его переполнении `Overfull \hbox` и напечатает на полях выходного документа ярко выделяющийся черный прямоугольник. Для каждой переполненной строки (*горизонтального бокса*) дается ее положение во входном файле, содержание переполненных боксов в сокращенном виде, а также указывается, насколько не помещается содержимое в боксе. Сокращенная запись переполненных боксов показывает переносы, которые  $\TeX$  испробовал перед тем, как прибегнуть к переполнению. Вам не надо доставать карандаш и бумагу, чтобы записать полученные сообщения о переполненных боксах, прежде чем они исчезнут из вида, так как  $\TeX$  всегда ведет “протокол” или “log-файл”, в который записывает все интересное, что произошло во время каждого сеанса работы.

Конечно, вам не нужны такие боксы, поэтому у  $\TeX$ 'а есть несколько способов убрать их. Иногда проблема переполнения бокса может быть решена, если просто намекнуть  $\TeX$ 'у, как лучше переносить слова (см. раздел **Формирование строк; переносы**). Если же этого недостаточно, можно сделать выключку формулы, которая не помещается на этой строке, изменить текст абзаца, или, если мы имеем дело с выключной формулой, которая шире строки всего на один-два пункта, просто не обращать внимания на предупреждение, а черный прямоугольник убрать командой  $\LaTeX$ 'а `\NoBlackBoxes`.

Это глобальная команда, которая действует на весь последующий материал, даже если она заключена в фигурные скобки или в знаки  $\$$ . Имеется также обратная команда `\BlackBoxes` для возврата черных прямоугольников, но в одном и том же файле обе эти команды употреблять нельзя.

Аналогично, могут встретиться сообщения `Underfull \vbox` и `Overfull \vbox`. Они указывают на то, что у  $\TeX$ 'а возникли трудности с формированием страницы и он не может найти хороший разрыв для перехода на следующую страницу.

Переполнение `\vbox` (т. е. *вертикального бокса*) может произойти, если вы задали `\aligned`, который больше, чем страница.  $\TeX$  все-таки разместит материал на странице, но пожалуется на переполнение вертикального бокса. Если даже материал из `\aligned` не занимает всю страницу, его, возможно, придется разместить, начиная со следующей страницы, а на предыдущей странице окажется мало материала. Тогда будет выдано предупреждение о недогрузке вертикального бокса.

## 8.3. Автоматическая проверка синтаксиса

Если вы работаете на медленном компьютере и хотите только проверить свой входной файл на предмет возможных ошибок, не рассчитывая получать распечатку выходного документа, можно отказаться от получения dvi-файла, сократив тем самым время работы  $\TeX$ 'а в 2–4 раза. Для этого во входном файле после строки `\documentstyle` надо поставить команду `\syntax`. В этом случае  $\TeX$  не выдаст никакого результата, кроме сообщения о встретившихся ошибках.

У команды `\syntax` есть один существенный недостаток — проводя синтаксический контроль входного файла, она не отслеживает возможные переполнения боксов. Поэтому после исправления найденных командой `\syntax` ошибок можно заменить ее на команду `\galley`s.  $\LaTeX$  будет имитировать режим типографской гранки, выдавать все сообщения о возможно встретившихся переполненных или незаполненных боксах, но по-прежнему не будет писать dvi-файл.

$\mathcal{A}\mathcal{M}\mathcal{S}$ - $\text{T}\mathcal{E}\mathcal{X}$  имеет также универсальную команду `\printoptions`. Если поместить ее во входной файл, то  $\mathcal{A}\mathcal{M}\mathcal{S}$ - $\text{T}\mathcal{E}\mathcal{X}$  спросит: “Что вы желаете, проверку синтаксиса (S), гранки (G) или распечатку (P)? Введите S, G или P и нажмите `<return>`”:

```
Do you want Syntax check), G(alleys) or P(ages)
Type S, G or P, follow by <return>
```

Если в ответ на этот запрос вы ответите S, G или P, то  $\text{T}\mathcal{E}\mathcal{X}$  будет работать в соответствующем вашему ответу режиме. Если же ответ будет отличаться от указанных трех, то  $\text{T}\mathcal{E}\mathcal{X}$  будет повторять свой вопрос до тех пор, пока не уговорит вас ответить адекватно.

Можно спорить о том, действительно ли так полезен автоматический синтаксический контроль. Мы, например, считаем, что часто для исправления ошибок полезно посмотреть именно `dvi`-файл. Однако, безусловно, надо знать и о такой возможности, чтобы принять наиболее рациональное с вашей точки зрения решение.

9. СРАВНЕНИЕ С PLAIN  $\mathcal{T}\mathcal{E}\mathcal{X}$ 'ОМ

$\mathcal{A}\mathcal{M}\mathcal{S}$ - $\mathcal{T}\mathcal{E}\mathcal{X}$  является макронадстройкой над Plain  $\mathcal{T}\mathcal{E}\mathcal{X}$ 'ом, поэтому в нем можно использовать большинство конструкций из Plain  $\mathcal{T}\mathcal{E}\mathcal{X}$ 'а. Так, например, можно использовать все команды генерации боксов, команды для получения таблиц, команды для разбиения строк и страниц и даже команду `\item` (не забывая при этом, что команда `\item` Plain  $\mathcal{T}\mathcal{E}\mathcal{X}$ 'а абсолютно отлична от той `\item`, которая встречается в конструкции  $\mathcal{A}\mathcal{M}\mathcal{S}$ - $\mathcal{T}\mathcal{E}\mathcal{X}$ 'а `\roster`.)

Однако некоторые средства Plain  $\mathcal{T}\mathcal{E}\mathcal{X}$ 'а не могут быть использованы в  $\mathcal{A}\mathcal{M}\mathcal{S}$ - $\mathcal{T}\mathcal{E}\mathcal{X}$ 'е, а некоторые команды  $\mathcal{A}\mathcal{M}\mathcal{S}$ - $\mathcal{T}\mathcal{E}\mathcal{X}$ 'а перекрывают команды Plain  $\mathcal{T}\mathcal{E}\mathcal{X}$ 'а с тем же именем. Для тех, кто использует Plain  $\mathcal{T}\mathcal{E}\mathcal{X}$  вместе с  $\mathcal{A}\mathcal{M}\mathcal{S}$ - $\mathcal{T}\mathcal{E}\mathcal{X}$ 'ом или предполагает переключаться с одного на другой, может оказаться полезной следующая информация:

Команда `\.` в Plain  $\mathcal{T}\mathcal{E}\mathcal{X}$ 'е дает точку над следующей буквой, тогда как в  $\mathcal{A}\mathcal{M}\mathcal{S}$ - $\mathcal{T}\mathcal{E}\mathcal{X}$ 'е она обозначает точку после аббревиатуры, т. е. точку, которая не стоит в конце предложения. Точка же над буквой  $\mathcal{A}\mathcal{M}\mathcal{S}$ - $\mathcal{T}\mathcal{E}\mathcal{X}$  получает командой `\D` (точка под буквой продолжает получаться командой Plain  $\mathcal{T}\mathcal{E}\mathcal{X}$ 'а `\d`). Аналогично, поскольку в Plain  $\mathcal{T}\mathcal{E}\mathcal{X}$ 'е черта под буквой получается командой `\b`, для черты над буквой в  $\mathcal{A}\mathcal{M}\mathcal{S}$ - $\mathcal{T}\mathcal{E}\mathcal{X}$ 'е выбрана команда `\B`. Старая же команда для получения черты `\='` становится в  $\mathcal{A}\mathcal{M}\mathcal{S}$ - $\mathcal{T}\mathcal{E}\mathcal{X}$ 'е неопределенной.

В Plain  $\mathcal{T}\mathcal{E}\mathcal{X}$ 'е следует заботиться о том, чтобы ни с одной стороны связки `~` не остался пробел, тогда как  $\mathcal{A}\mathcal{M}\mathcal{S}$ - $\mathcal{T}\mathcal{E}\mathcal{X}$  эти пробелы просто игнорирует.

В Plain  $\mathcal{T}\mathcal{E}\mathcal{X}$ 'е команды `\{` и `\}`, а также `\,` и `\!` используются только в математической моде, а в тексте вместо `\,` и `\!` применяются, соответственно, `\thinspace` и `\negthinspace`.  $\mathcal{A}\mathcal{M}\mathcal{S}$ - $\mathcal{T}\mathcal{E}\mathcal{X}$  же все эти шесть команд использует как в тексте, так и в математике.

Аналогично,  $\mathcal{A}\mathcal{M}\mathcal{S}$ - $\mathcal{T}\mathcal{E}\mathcal{X}$  допускает использование как в математической моде, так и в тексте, таких команд, как `\medspace`, `\negmedspace`, `\thickspace` и `\negthickspace`, а в качестве синонима для `\medspace` применяет также `\;`. В Plain  $\mathcal{T}\mathcal{E}\mathcal{X}$ 'е команду `\;` можно использовать только в математической моде. С другой стороны, Plain имеет краткое название `\>` для  $\mathcal{A}\mathcal{M}\mathcal{S}$ - $\mathcal{T}\mathcal{E}\mathcal{X}$ 'овского `\medskip`, которое в  $\mathcal{A}\mathcal{M}\mathcal{S}$ - $\mathcal{T}\mathcal{E}\mathcal{X}$ 'е становится неопределенным.

Акцент `\t` работает в  $\mathcal{A}\mathcal{M}\mathcal{S}$ - $\mathcal{T}\mathcal{E}\mathcal{X}$ 'е и Plain  $\mathcal{T}\mathcal{E}\mathcal{X}$ 'е несколько по-разному. В  $\mathcal{A}\mathcal{M}\mathcal{S}$ - $\mathcal{T}\mathcal{E}\mathcal{X}$ 'е у команды `\t` два аргумента и даже можно перед вторым аргументом оставить пробел, но в Plain  $\mathcal{T}\mathcal{E}\mathcal{X}$ 'е никаких пробелов быть не должно.

$\mathcal{A}\mathcal{M}\mathcal{S}$ - $\mathcal{T}\mathcal{E}\mathcal{X}$  требует, чтобы в математической моде использовались команды замены шрифтов, отличные от команд замены шрифтов в тексте, например, `\bold`, а не `\bf` и т. п. Более того, команды типа `\bold` представляют собой управляющие последовательности с одним аргументом, а не команды изменения шрифта; команды изменения шрифта в математической моде можно использовать только в таком виде:

```
$$ \text{\bf ... }$$
```

При такой конструкции все пробелы из `...` появятся и в выходном результате. Plain  $\mathcal{T}\mathcal{E}\mathcal{X}$  же позволяет использовать команды типа `\bf` и в математической моде, но пробелы внутри математической моды игнорируются.

Для получения рукописных букв  $\mathcal{A}\mathcal{M}\mathcal{S}$ - $\mathcal{T}\mathcal{E}\mathcal{X}$  использует команду с одним аргументом `\cal`, а Plain  $\mathcal{T}\mathcal{E}\mathcal{X}$  — команду переключения шрифта `\cal`, которая в  $\mathcal{A}\mathcal{M}\mathcal{S}$ - $\mathcal{T}\mathcal{E}\mathcal{X}$ 'е становится неопределенной.

Для получения другого написания прописных греческих букв Plain  $\mathcal{T}\mathcal{E}\mathcal{X}$  использует команду замены шрифта на математический курсив `\mit`, тогда как в  $\mathcal{A}\mathcal{M}\mathcal{S}$ - $\mathcal{T}\mathcal{E}\mathcal{X}$ 'е для них просто есть другое название (`\varGamma` и т. п.).

Для старинного написания цифр 1, 2, 3, 4 ...  $\mathcal{A}\mathcal{M}\mathcal{S}$ - $\mathcal{T}\mathcal{E}\mathcal{X}$  использует не команду Plain  $\mathcal{T}\mathcal{E}\mathcal{X}$ 'а `\oldstyle`, которая становится неопределенной, а свою собственную команду `\oldnos`.

Следующее разногласие между Plain  $\mathcal{T}\mathcal{E}\mathcal{X}$ 'ом и  $\mathcal{A}\mathcal{M}\mathcal{S}$ - $\mathcal{T}\mathcal{E}\mathcal{X}$ 'ом касается верхних индексов: в Plain  $\mathcal{T}\mathcal{E}\mathcal{X}$ 'е для получения  $f''^2$  можно набирать следующее:

```
$f''^2$
```

вместо того, чтобы вводить

$\$f^{\{\backslash\prime\backslash\prime2\}}\$$  или  $\$f''^{\{\}^2}\$$

В  $\mathcal{A}\mathcal{M}\mathcal{S}$ - $\text{T}\text{E}\text{X}$ 'е таким приемом пользоваться нельзя, поскольку нет четких правил для получения  $f^{2''}$

Команды `\ldots` и `\cdots` в  $\mathcal{A}\mathcal{M}\mathcal{S}$ - $\text{T}\text{E}\text{X}$ 'е автоматически вставляют дополнительный тонкий пробел в конце формулы или после правого ограничителя, подобно общей команде  $\mathcal{A}\mathcal{M}\mathcal{S}$ - $\text{T}\text{E}\text{X}$ 'а `\dots`. В Plain  $\text{T}\text{E}\text{X}$ 'е такого автоматического соглашения нет.

`\rmod` в  $\mathcal{A}\mathcal{M}\mathcal{S}$ - $\text{T}\text{E}\text{X}$ 'е в формулах в тексте и в выключных формулах применяет разные пробелы.

Большие операторы типа `\sum` и названия операторов типа `\max` в Plain  $\text{T}\text{E}\text{X}$ 'е и  $\mathcal{A}\mathcal{M}\mathcal{S}$ - $\text{T}\text{E}\text{X}$ 'е определяются совершенно по-разному, так что в разных форматах они могут давать разные результаты.

Команды `\matrix` и `\cases` имеют в Plain  $\text{T}\text{E}\text{X}$ 'е и  $\mathcal{A}\mathcal{M}\mathcal{S}$ - $\text{T}\text{E}\text{X}$ 'е совершенно разный синтаксис.

Команды `\proclaim` и `\footnote` используются и в Plain  $\text{T}\text{E}\text{X}$ 'е и в  $\mathcal{A}\mathcal{M}\mathcal{S}$ - $\text{T}\text{E}\text{X}$ 'е, но имеют там совершенно разный синтаксис. Так, например, в Plain  $\text{T}\text{E}\text{X}$ 'е команда `\footnote` имеет два аргумента, первый из которых задает знак сноски, а второй — ее текст, тогда как в  $\mathcal{A}\mathcal{M}\mathcal{S}$ - $\text{T}\text{E}\text{X}$ 'е у этой команды один аргумент (текст сноски), а знак сноски определяется стилем.

10. ВЫРАЖЕНИЕ ПРИЗНАТЕЛЬНОСТИ ЗА ИСПОЛЬЗОВАНИЕ  $\mathcal{A}\mathcal{M}\mathcal{S}$ - $\text{T}\mathcal{E}\mathcal{X}$ 

Далее предлагаются формы выражения признательности за использование  $\mathcal{A}\mathcal{M}\mathcal{S}$ - $\text{T}\mathcal{E}\mathcal{X}$  для подготовки документа к публикации.

Отдельный документ должен внизу первой страницы содержать следующее:

Typeset by  $\mathcal{A}\mathcal{M}\mathcal{S}$ - $\text{T}\mathcal{E}\mathcal{X}$

(такая запись автоматически производится в стиле препринт  $\mathcal{A}\mathcal{M}\mathcal{S}$ - $\text{T}\mathcal{E}\mathcal{X}$ ).

Если с помощью  $\mathcal{A}\mathcal{M}\mathcal{S}$ - $\text{T}\mathcal{E}\mathcal{X}$  был подготовлен целиком журнал или книга, то на странице, где указаны копирайты, должна присутствовать такая запись:

This [journal/book] was typeset by  $\mathcal{A}\mathcal{M}\mathcal{S}$ - $\text{T}\mathcal{E}\mathcal{X}$ , the  $\text{T}\mathcal{E}\mathcal{X}$  macro system of the American Mathematical Society.

Если в журнале или книге на  $\mathcal{A}\mathcal{M}\mathcal{S}$ - $\text{T}\mathcal{E}\mathcal{X}$  подготовлены только некоторые статьи, то эти статьи должны быть отмечены, как указано выше, а на странице с копирайтами должна присутствовать запись:

$\mathcal{A}\mathcal{M}\mathcal{S}$ - $\text{T}\mathcal{E}\mathcal{X}$  is the  $\text{T}\mathcal{E}\mathcal{X}$  macro system of the American Mathematical Society.

## ПРИЛОЖЕНИЕ А. ПРИМЕР БИБЛИОГРАФИИ

```
\Refs
\ref\key 4 % assuming \refstyle{C}
\by V. I. Arnol$\$d, A. N. Varchenko,
  and S. M. Guse\u{i} n-Zade
\book Singularities of differentiable maps.~\rom I
\publ ‘‘Nauka’’ \publaddr Moscow \yr 1982
\lang Russian
\endref

\ref\key 5\by same
\book Singularities of differentiable maps.~\rom{II}
\publ ‘‘Nauka’’ \publaddr Moscow \yr 1984
\lang Russian
\endref

\ref\key 6
\by O. A. Ladyzhenskaya
\book Mathematical problems in the dynamics
  of a viscous incompressible fluid
\bookinfo 2nd rev. aug. ed.
\publ ‘‘Nauka’’ \publaddr Moscow \yr 1970
\lang Russian
\transl English transl. of 1st ed.
\book The mathematical theory of viscous
  incompressible flow
\publ Gordon and Breach \publaddr New York
\yr 1963; rev. 1969
\endref
```

```
\ref\key 7
\by P. D. Lax and C. D. Levermore
\paper The small dispersion limit for the
  KdV equation.~\rom I
\jour Comm. Pure Appl. Math. \vol 36 \yr 1983
\pages 253--290 \finalinfo (overview)
\moreref\paper \rom{II}
\jour Comm. Pure Appl. Math.
\vol 36 \yr 1983 \pages 571--594
\moreref\paper \rom{III}
\jour Comm. Pure Appl. Math.
\vol 36 \yr 1983 \pages 809--829 \endref

\ref\key 10 \by S. Osher
\paper Shock capturing algorithms for equations of
  mixed type
\inbook Numerical Methods for Partial Differential
  Equations \eds S. I. Hariharan and T. H. Moulton
\publ Longman \publaddr New York \yr 1986
\pages 305--322
\endref

\ref\key 17 \by G. S. Petrov
\paper Elliptic integrals and their nonoscillatory
  behavior
\jour Funktsional. Anal. i Prilozhen.
\vol 20 \yr 1986 \pages 46--49
\transl\nofrills English transl. in
\jour Functional Anal. Appl. \vol 20\yr 1986
\endref
```

```
% switch to a different references style
\refstyle{A}
\widestnumber\key{GHMR}

\ref\key C1
\by B. Coomes
\book Polynomial flows, symmetry groups, and
  conditions sufficient for injectivity of maps
\bookinfo Ph.D. thesis, Univ. Nebraska--Lincoln
\yr 1988
\endref

\ref\key C2
\bysame % B. Coomes
\paper The Lorenz system does not have a
  polynomial flow
\jour J. Differential Equations
\toappear
\endref

\ref\key GHMR
\by J. Guckenheimer, P. Holmes, M. Martineau,
  and L. P. Robinson
\book Nonlinear oscillations, dynamical systems,
  and bifurcations of vector fields
\bookinfo % fields can be left blank
\publ Springer-Verlag \publaddr New York
\yr 1983
\endRefs
```

## REFERENCES

4. V. I. Arnol'd, A. N. Varchenko, and S. M. Guseĭn-Zade, *Singularities of differentiable maps*. I, "Nauka", Moscow, 1982. (Russian)
  5. ———, *Singularities of differentiable maps*. II, "Nauka", Moscow, 1984. (Russian)
  6. O. A. Ladyzhenskaya, *Mathematical problems in the dynamics of a viscous incompressible fluid*, 2nd rev. aug. ed., "Nauka", Moscow, 1970 (Russian); English transl. of 1st ed., *The mathematical theory of viscous incompressible flow*, Gordon and Breach, New York, 1963; rev. 1969.
  7. P. D. Lax and C. D. Levermore, *The small dispersion limit for the KdV equation*. I, *Comm. Pure Appl. Math.* **36** (1983), 253–290, (overview); II, *Comm. Pure Appl. Math.* **36** (1983), 571–594; III, *Comm. Pure Appl. Math.* **36** (1983), 809–829.
  10. S. Osher, *Shock capturing algorithms for equations of mixed type*, *Numerical Methods for Partial Differential Equations* (S. I. Hariharan and T. H. Moulton, eds.), Longman, New York, 1986, pp. 305–322.
  17. G. S. Petrov, *Elliptic integrals and their nonoscillatory behavior*, *Funktsional. Anal. i Prilozhen.* **20** (1986), 46–49; English transl. in *Functional Anal. Appl.* **20** (1986).
- [C1] B. Coomes, *Polynomial flows, symmetry groups, and conditions sufficient for injectivity of maps*, Ph.D. thesis, Univ. Nebraska–Lincoln, 1988.
- [C2] ———, *The Lorenz system does not have a polynomial flow*, *J. Differential Equations* (to appear).
- [GHMR] J. Guckenheimer, P. Holmes, M. Martineau, and L. P. Robinson, *Nonlinear oscillations, dynamical systems, and bifurcations of vector fields*, Springer-Verlag, New York, 1983.

ПРИЛОЖЕНИЕ Б. МАКЕТЫ ШРИФТОВ КОЛЛЕКЦИИ AMSFONTS

**Замечание:** Номера рядов и колонок шестнадцатиричные.

Дополнительные символы, группа 1, средней толщины – msam10

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
0	◻	▣	▤	◻	■	▪	◊	◆	↻	↻	⇌	⇌	◻	▮	▮	▮	0
1	→	↑	⇌	⇌	⇌	⇌	⇌	⇌	↓	↓	↔	↔	↔	↔	↔	↔	1
2	↔	↔	↔	↔	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	2
3	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	3
4	◻	◻	◻	◻	◻	◻	★	∅	▼	▶	◀	→	←	△	▲	▽	4
5	#	∞	∞	∞	∞	∞	∞	∞	✓	∞	∞	∞	∞	∞	∞	∞	5
6	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	6
7	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	7
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	

Дополнительные символы, группа 2, средней толщины – msbm10

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
0	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	0
1	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	1
2	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	2
3	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	3
4	#	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	4
5	P	Q	R	S	T	U	V	W	X	Y	Z						5
6	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	6
7	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	7
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	

Euler Fraktur средней толщины – eufm10

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
0	b	d	f	f	g	t	t	u									0
1			'	'													1
2		!					&	'	(	)	*	+	,	–	.	/	2
3	o	1	2	3	4	5	6	7	8	9	:	;		=		?	3
4		Ɱ	Ɱ	Ɱ	Ɱ	Ɱ	Ɱ	Ɱ	Ɱ	Ɱ	Ɱ	Ɱ	Ɱ	Ɱ	Ɱ	Ɱ	4
5	Ɱ	Ɱ	Ɱ	Ɱ	Ɱ	Ɱ	Ɱ	Ɱ	Ɱ	Ɱ	Ɱ	Ɱ	[	]	^		5
6		a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	6
7	p	q	r	s	t	u	v	w	x	y	z			"		l	7
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	

Euler курсив (романский) средней толщины – eurm10

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
0	Г	Δ	Θ	Λ	Ξ	Π	Σ	Υ	Φ	Ψ	Ω	α	β	γ	δ	ε	0
1	ζ	η	θ	ι	κ	λ	μ	ν	ξ	π	ρ	σ	τ	υ	φ	χ	1
2	ψ	ω	ε	ϑ	ω			φ									2
3	0	1	2	3	4	5	6	7	8	9	.	,	<	/	>		3
4	∂	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	4
5	P	Q	R	S	T	U	V	W	X	Y	Z						5
6	ℓ	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	6
7	p	q	r	s	t	u	v	w	x	y	z	ι	Ͽ				7
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	

Euler рукописный средней толщины – eusm10

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
0	–																0
1									~								1
2																	2
3											¬		Ɱ	Ɱ			3
4	Ɱ	A	B	C	D	E	F	G	Ɱ	Ɱ	Ɱ	Ɱ	Ɱ	Ɱ	Ɱ	Ɱ	4
5	P	Q	R	S	T	U	V	W	X	Y	Z				^	∨	5
6							{	}							\		6
7									§								7
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	

Кириллица, средней толщины – wncyr10

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
0	Ь	Ъ	Ц	Э	І	Є	Ђ	Ѓ	њ	љ	џ	ѡ	і	е	ђ	ѓ	0
1	Ю	Ж	Й	Ё	V	Ѡ	S	Я	ю	ж	й	ё	v	ѡ	s	я	1
2	“	!	”	Ђ	˘	%	’	’	(	)	*	Ђ	,	-	.	/	2
3	0	1	2	3	4	5	6	7	8	9	:	;	«	ı	»	?	3
4	˘	A	B	Ц	Д	Е	Ф	Г	X	И	J	K	Л	М	Н	О	4
5	П	Ч	Р	С	Т	У	В	Щ	Ш	Ы	З	[	“	]	Ь	Ъ	5
6	‘	a	b	ц	д	e	ф	г	x	и	j	к	л	м	н	о	6
7	п	ч	р	с	т	у	в	щ	ш	ы	з	–	—	№	ь	ъ	7
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	

Кириллица, курсив, средней толщины – wncyr10

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
0	Ь	Ъ	Ц	Э	І	Є	Ђ	Ѓ	њ	љ	џ	ѡ	і	е	ђ	ѓ	0
1	Ю	Ж	Й	Ё	V	Ѡ	S	Я	ю	ж	й	ё	v	ѡ	s	я	1
2	“	!	”	Ђ	˘	%	’	’	(	)	*	њ	,	-	.	/	2
3	0	1	2	3	4	5	6	7	8	9	:	;	«	ı	»	?	3
4	˘	A	B	Ц	Д	Е	Ф	Г	X	И	J	K	Л	М	Н	О	4
5	П	Ч	Р	С	Т	У	В	Щ	Ш	Ы	З	[	“	]	Ь	Ъ	5
6	‘	a	b	ц	д	e	ф	г	x	и	j	к	л	м	н	о	6
7	п	ч	р	с	т	у	в	щ	ш	ы	з	–	—	№	ь	ъ	7
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	

Computer Modern капитель – cmcsc10

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
0	Γ	Δ	Θ	Λ	Ξ	Π	Σ	Υ	Φ	Ψ	Ω	↑	↓	'	ı	ı	0
1	I	J	`	'	˘	˘	-	˚	,	ss	Æ	Œ	∅	Æ	Œ	∅	1
2	-	!	”	#	\$	%	&	'	(	)	*	+	,	-	.	/	2
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?	3
4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	4
5	P	Q	R	S	T	U	V	W	X	Y	Z	[	“	]	^	.	5
6	'	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	6
7	P	Q	R	S	T	U	V	W	X	Y	Z	-	—	”	~	..	7
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	

Computer Modern жирный математический курсив – cmmb10

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
0	Γ	Δ	Θ	Λ	Ξ	Π	Σ	Υ	Φ	Ψ	Ω	α	β	γ	δ	ε	0
1	ζ	η	θ	ι	κ	λ	μ	ν	ξ	π	ρ	σ	τ	υ	φ	χ	1
2	ψ	ω	ε	ϑ	Ϙ	ϙ	ς	φ	←	↖	→	↗	◁	▷	◃	◅	2
3	o	1	2	3	4	5	6	7	8	9	.	,	<	/	>	*	3
4	∂	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	4
5	P	Q	R	S	T	U	V	W	X	Y	Z	b	h	#	~	~	5
6	l	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	6
7	p	q	r	s	t	u	v	w	x	y	z	i	j	φ	→	~	7
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	

Computer Modern жирные математические символы – cmbsty10

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
0	–	·	×	*	÷	◇	±	≠	⊕	⊖	⊗	⊘	⊙	◯	◦	●	0
1	×	≡	⊂	⊃	≤	≥	≲	≳	≈	≈	⊆	⊇	≪	≫	≲	≳	1
2	←	→	↑	↓	↔	↗	↘	↺	⇐	⇒	↑	↓	↔	↗	↘	∞	2
3	/	∞	∈	∃	△	▽	/	·	∀	∃	¬	∅	ℜ	ℑ	⊤	⊥	3
4	<i><b>ℵ</b></i>	<i><b>ℒ</b></i>	<i><b>ℬ</b></i>	<i><b>ℭ</b></i>	<i><b>ℰ</b></i>	<i><b>ℱ</b></i>	<i><b>ℱ</b></i>		<i><b>ℋ</b></i>	<i><b>ℐ</b></i>	<i><b>ℐ</b></i>	<i><b>℔</b></i>	<i><b>ℒ</b></i>	<i><b>℔</b></i>	<i><b>ℕ</b></i>	<i><b>℔</b></i>	4
5	<i><b>ℙ</b></i>	<i><b>ℚ</b></i>	<i><b>℔</b></i>	∩	⊕	∧	∨	5									
6	⌈	⌋	⌊	⌋	⌈	⌋	{	}	<	>			↕	↕	\	~	6
7	√	∏	∇	∫	⊔	⊔	≡	≡	§	†	‡	¶	♣	◇	♥	♠	7
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	

Computer Modern математический расширенный шрифт – cmex10

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
0	(	)	[	]	⌈	⌋	⌈	⌋	{	}	<	>			/	\	0
1	(	)	(	)	[	]	⌈	⌋	⌈	⌋	{	}	<	>	/	\	1
2	(	)	[	]	⌈	⌋	⌈	⌋	{	}	<	>	/	\	/	\	2
3	(	)	⌈	⌋	⌈	⌋			⌈	⌋	⌈	⌋	{	}	·		3
4	\	/			<	>	⊔	⊔	℔	℔	⊙	⊙	⊕	⊕	⊗	⊗	4
5	Σ	Π	∫	∫	∩	⊕	∧	∨	Σ	Π	∫	∫	∩	⊕	∧	∨	5
6	∏	∏	^	^	~	~	~	~	[	]	⌈	⌋	⌈	⌋	{	}	6
7	√	√	√	√	√		⌈	⌋	↑	↓	↶	↷	↶	↷	↑	↓	7
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	

## БИБЛИОГРАФИЯ

- Дональд Е. Кнут. *Все про  $\mathcal{T}\mathcal{E}\mathcal{X}$* . Пер. с англ., АО РДТЕХ, 1993.  
М. Спивак. *Воспитательный  $\mathcal{T}\mathcal{E}\mathcal{X}$* , Пер. с англ., Москва “Мир”, 1993.  
User’s Guide to  $\mathcal{A}\mathcal{M}\mathcal{S}$ - $\mathcal{T}\mathcal{E}\mathcal{X}$  Version 2.1.  
User’s Guide to AMSFonts Version 2.2.