

М.Спивак

ВОСХИТИТЕЛЬНЫЙ AMS-TEX:

Руководство по комфортному изготовлению научных публикаций в пакете AMS-TeX

Книга американского специалиста, представляющая собой руководство по издательской системе AMS-TeX, предназначенной для подготовки научных публикаций с большим количеством математических формул. Эта система работает как на больших ЭВМ, так и на персональных компьютерах. Приведенных в книге сведений вполне достаточно для практической подготовки научных статей к публикации. Изложение рассчитано на неподготовленного пользователя и сопровождается большим числом примеров и упражнений.

Для научных работников, самостоятельно готовящих к изданию свои публикации, для специалистов по издательским системам, аспирантов и студентов соответствующих специальностей.

ОГЛАВЛЕНИЕ

От переводчика и редактора	5
Предисловие ко второму изданию	7
Слова признательности	8
Прочь половые признаки	9
Введение. О сложных видах набора	10
Часть 1. Закуски	15
0. Первое знакомство; ключевая глава	15
Символы, имеющиеся на клавиатуре	
1. Лингвистические уТеХи	18
Обычный текст и управляющие последовательности	
2. Тиснем любой набор	26
Замена шрифтов	
3. Ваш первый TeX-практикум	31
Обработка файла TeX'ом	
4. Технический контроль ошибок	37
Сообщения об ошибках, и как на них реагировать	
5. Разделяющие пробелы и связывающие тильды	41
Тонкости, касающиеся пробелов и разрыва строк	
6. С каждым знаком знаком	43
Особые символы и акценты	
Часть 2. Дежурные блюда	49
7. MaTEmатический экстаз	49
Математические формулы в тексте	
8. Скверный разрыв? Сделайте "выключку"	58
Выключные формулы	
9. 2-й уровень сложности	63
Верхние и нижние индексы	
10. Наших проблем громадь	70
Дроби, биномиальные коэффициенты и т. п.	

11. Большие дары TeX'a	75
\int , \sum и другие "большие операторы"	
12. Создание своего пространства	79
Корректировка пробелов в математических формулах	
13. Длина меняется сама собой	81
Ограничители и другие символы, меняющие размеры	
14. Прямым текстом	89
Прямой шрифт в формулах	
15. Держите равнение	97
Системы уравнений и многострочные выключные формулы	
16. Столько всяких удобств	102
Разрыв слишком длинных формул	
17. Затруднительное положение	106
Матрицы	
Часть 3. Напитки и приправы	110
18. Практическое самоуправление	110
Определение новых управляющих последовательностей	
19. Технические приемы	123
Технический толковый словарь	
Приложения	179
A. Стиль AMS для журнальных статей	179
B. Ответы ко всем упражнениям	194
C. Библиография	239
D. Сравнение с 'plain' TeX'ом	244
E. Отсутствующие клавиши	247
F. Специальные знаки	248
G. Дополнительные шрифты	252
H. TUG — группа пользователей TeX'a	258
I. На помощь!	259
Предметный указатель	260

От переводчика и редактора

Система Т_РХ подготовки научно-технических публикаций была разработана Д. Кнудом около 15 лет назад. За эти годы она не только не оказалась вытесненной современными настольными издательскими системами, но приобрела еще более широкую популярность во всем мире и теперь завоевывает нашу страну. Столь продолжительное “творческое долголетие” уникально среди компьютерных издательских систем, и по этому параметру Т_РХ стоит в одном ряду с лучшими языками программирования. Но их роднит не только это. Фактически Т_РХ предоставляет в распоряжение пользователя универсальный язык программирования. Это означает, что с помощью Т_РХ’а можно запрограммировать все, что угодно, даже столь экзотические вещи, как интерпретатор языка Бейсик на Т_РХ’е, описанный в одном из недавних выпусков журнала *TUGboat*, издаваемого американским обществом пользователей Т_РХ’а. Однако стандартный путь использования базовых средств Т_РХ’а состоит в создании надстроек, предоставляющих пользователю удобные средства для работы в какой-либо определенной области. К их числу относится и пакет *AMS-T_РХ*, описываемый в этой книге, который ориентирован в первую очередь на подготовку математических текстов.

Как и всякий хороший язык программирования, Т_РХ не только предоставляет средства для записи “программ”, т. е., в данном случае, печатных документов и стилевых файлов, но и предлагает систему понятий, в рамках которой можно рассуждать о различных вариантах оформления документа. Поэтому изучение Т_РХ’а, и данной книги в частности, полезно даже в том случае, если вы не собираетесь использовать Т_РХ в практической работе. Но, конечно, в первую очередь эта книга призвана служить руководством по конкретному пакету *AMS-T_РХ*.

Перевод на русский язык руководства к американскому пакету, предназначенному для подготовки к изданию в американских журналах математических работ на английском языке — задача сама по себе внутренне противоречивая. Как и водится в подобных случаях, мысль эта полностью дошла до нашего сознания, когда отступить было поздно. При написании книги ее автор был настроен довольно игриво, что видно из самого названия, представляющего собой перефразировку скандально известного “The Joy of Sex”. Это настроение не оставило его до последней страницы и доставило нам при подготовке перевода немало хлопот. Увы, далеко не всякая игра с английскими словами превратилась в столь же изящный афоризм на русском языке. Но мы старались... Понятно, что не переводить на русский язык многочисленные примеры, в которых стиль автора отразился наиболее ярко, значило бы иссушить книгу. Мы также не могли себе позволить всюду приводить параллельно английский и русский варианты одного и того же примера — книга бы разбухла и вряд ли доставила бы читателю (и издателю) радость. Давать же везде только русский эквивалент примеров означало бы выплеснуть вместе с водой и младенца — во многих случаях такие примеры становились бессмысленными. Поэтому был избран эклектический путь: вы найдете здесь все варианты! Причем в начале книги в основном даются параллельные тексты или только русский перевод; по мере усложнения материала (а,

значит, и с ростом квалификации читателя) предпочтение отдается примерам на языке оригинала.

Та же эклектика наблюдается и в соблюдении (или несоблюдении) некоторых типографских правил. В соответствии с принятыми в англоязычной литературе правилами увеличивается пробел после точки в конце предложения и употребляются “английские” кавычки, но тире — с отбивками с двух сторон согласно нашим правилам. Что касается формул, то здесь мы строго следовали канонам англоязычной литературы, иначе это руководство потеряло бы всякий смысл.

Следует обратить внимание читателя на одно существенное отличие между принятой у нас типографской системой мер (типометрией) и системой мер полиграфистов США. В основе и той, и другой типометрии лежит типографский пункт, равный $1/72$ дюйма, с той лишь разницей, что у нас это французский дюйм, который равен 27.06 мм, а в США это так называемый королевский дюйм, величина которого составляет 25.4 мм. Поэтому, когда в книге речь идет о шрифте кегля 10, у нас ему соответствует шрифт примерно 9-го кегля.

Перевод на русский язык такой книги естественным образом предполагает и повторение технологии подготовки ее оригинала-макета. Поэтому пришлось по ходу дела решать и сверхзадачу: создавать русифицированную версию пакета \TeX и разрабатывать кириллические шрифты. Работу по русификации выполнил А. Б. Ходулёв; весь труд по разработке шрифтов мы ввалили на О. Г. Лапко. Мы приносим ей сердечную признательность за терпение, с которым она сносила наше руководство, и выражаем восхищение незаурядными способностями, которые раскрылись у нее в процессе этой деятельности. Оригинал-макет книги был выполнен И. А. Маховой на основе \TeX -файлов оригинала, любезно предоставленных Американским математическим обществом.

Это, по-видимому, первое у нас переводное издание книги о системе \TeX , хотя, конечно, было бы естественнее начинать знакомство с \TeX 'ом с книг самого Дональда Кнута. К сожалению, ситуация в наших издательствах сейчас такова, что работа по переводу и изданию научно-технической литературы сродни благотворительной деятельности. Тем не менее хочется надеяться, что найдутся и энтузиасты, и спонсоры, которые будут способствовать распространению \TeX нической литературы. Во всем мире сам пакет \TeX распространяется свободно, и для его популяризации в разных странах создаются группы пользователей \TeX 'а. Самая первая — TUG — была создана в США в 1979 г. В России с 1991 г. функционирует Ассоциация *CyrTUG* пользователей кириллического \TeX 'а, которая ставит перед собой аналогичную задачу.

В заключение мы хотим сказать, что возлагаем на себя всю ответственность за возможные неточности и опечатки, привнесенные в русский вариант книги, и хотим надеяться, что нам удалось хоть в малой степени передать живость стиля автора. Ну а мощь самого пакета \TeX умалить невозможно при всем желании.

И. А. Маховая
А. Б. Ходулёв

Предисловие ко второму изданию

Книга “Восхитительный Т_ЕX” представляет собой учебник по издательской системе $\mathcal{A}\mathcal{M}\mathcal{S}$ -T_ЕX — расширению революционного пакета Дональда Кнута Т_ЕX для набора научно-технических текстов. При разработке $\mathcal{A}\mathcal{M}\mathcal{S}$ -T_ЕX’а ставилась цель облегчить ввод математических текстов и иметь возможность получать оригинал-макет в одном из заранее разработанных стилей. Этот учебник будет полезен как операторам-наборщикам, так и ученым, подготавливающим на компьютере свои рукописи. Упражнения, в большом количестве рассеянные по главам, должны побуждать читателя сесть за терминал и обучаться на практике.

Две первые части книги — “Закуски” и “Дежурные блюда” — научат читателя набирать те тексты и математические формулы, с которыми он сталкивается повседневно. В третьей части — “Напитки и приправы” — описаны более экзотические случаи и представлен шестидесятистраничный глоссарий Т_ЕXнических терминов. Научные работники и опытные операторы-наборщики сразу оценят ту легкость, с которой $\mathcal{A}\mathcal{M}\mathcal{S}$ -T_ЕX управляет внешним видом набранных страниц. Описание общепринятых правил математического набора послужит хорошим подспорьем для новичков в этом деле. В приложениях удачно собраны как часто используемые, так и довольно экзотические символы; кроме того, даны ответы к упражнениям.

Второе издание “Восхитительного Т_ЕX’а” дополнено новым материалом, отражающим те изменения, которые были сделаны в версии 2.0 макро-пакета $\mathcal{A}\mathcal{M}\mathcal{S}$ -T_ЕX. В версии 2.0 д-р Спивак сделал многие сообщения об ошибках и подсказки более понятными и устранил ряд погрешностей, которые вскрылись при работе с особыми случаями. Он ввел ряд усовершенствований в макро и сделал некоторые замены для экономии памяти. Кроме того, был облегчен доступ к шрифтам, отличным от тех, которые определены в plain Т_ЕX’е.

Когда д-р Спивак вносил изменения, относящиеся непосредственно к пакету $\mathcal{A}\mathcal{M}\mathcal{S}$ -T_ЕX, сотрудники Группы поддержки технического и программного обеспечения Американского математического общества существенно переработали стиль `amsrpt` с целью ввести ряд факультативных средств форматирования и дать пользователям больше свобод. Среди прочего — это, помимо стилей для статей, стили для монографий, колонтитулы, меняющийся размер страниц и оформление оглавлений. В первом издании “Восхитительного Т_ЕX’а” особенности стиля `amsrpt` излагались в гл. 7. В настоящем издании содержимое гл. 7 переместилось в приложение А. Сотрудники Группы поддержки технического и программного обеспечения Американского математического общества существенно дополнили приложение А, так что теперь оно может служить полным перечнем всех возможностей, предоставляемых стилем `amsrpt` в версии 2.0.

Американское математическое общество
Июль, 1990

Слова признательности за поддержку первого издания

Примерно в то время, когда я только начинал знакомиться с компьютерами, не испытывая при этом ничего кроме легкого отвращения, мой друг Ричард Пале с энтузиазмом изучал новую программу компьютерного набора, о которой он услышал как член Комиссии по технологии типографского набора при Американском математическом обществе. Он предложил мне немного поучиться этому и потом изложить приобретенный опыт в небольшом учебнике, ориентированном на неподготовленного пользователя, прекрасным образчиком которого я и являлся. Итак, мы отправились в Стэнфорд для двухнедельной работы над проектом, где присоединились к Барбаре Битон, Роберту Моррису и Риле Тедфорд.

Было это шесть лет тому назад. Теперь, когда первое издание этого учебника наконец увидело свет, можно назвать еще многие имена — тех, кто принимал участие в его подготовке.

Первые шаги в осуществлении этого проекта были бы невозможны без помощи сотрудников Американского математического общества, в особенности Барбары Битон, Сэма Уиддена и Рэя Гучера. Я глубоко признателен Дэвиду Фуксу, Джону Хобби, Артуру Келлеру и Джо Уинингу за то, что в течение моего пятидневного пребывания в Стэнфорде в 1983 г. они позволяли докучать им по поводу всяких туманных мест, и чрезвычайно благодарен Лесли Лэмпорту за беседы на тему \LaTeX . Майкл Харрисон любезно помог в моей работе над проектом в университетском колледже в Беркли летом 1984 г. В конце концов все эти шатания по университетам не могли не сказаться, и, несомненно, проект никогда бы не был завершен, если бы компания Personal TeX, Inc., не предоставила мне возможность использовать пакет РС TeX на моем персональном компьютере.

И ПРЕВЫШЕ ВСЕГО, однако, я ценю ту поддержку, которую получил от Дональда Э. Кнута, автора TeX'a, и выражаю ему огромную признательность за интерес к макро-проекту Американского математического общества, за помощь, которую он мне предложил, и, в особенности, за этот удивительный дар, которым он осчастливил все человечество.

Майкл Д. Спивак

Слова признательности за поддержку второго издания

Второе издание книги “Восхитительный TeX” стало возможным благодаря усилиям нескольких специалистов. Мы хотим поблагодарить Майкла Спивака за работу над \LaTeX -макро, которые теперь используются при издании всех публикаций Американского математического общества. Большой вклад внесли сотрудники Группы поддержки технического и программного обеспечения АМО — Ральф Янген (руководитель), Майкл Даунс, Джанин Уинтер, Барбара Битон, Юлия Помрой и Нил Бартоломью — переработавшие стиль препринта (amsrpt) и дополнившие текст книги, в особенности приложение А.

Американское математическое общество

Прочь половые признаки

*Это предоставляется читателю
на Его или Ее вкус*

Так как ТҒХ — в значительной мере революционное явление в наборном деле, автор решил, что в этой книге окажется уместным революционный подход к поп-ҒҒХуальной терминологии. Автор сам, разумеется, совершенно беспристрастен. Как говаривал Марк Твен, или мог бы сказать: “Я хочу лишь знать, что мужчина и женщина — это человеческие существа: для меня этого достаточно; ни та, ни другой ничем не хуже. Но меня бесит, когда надо говорить ‘он или она’, либо ‘его или ее’, либо использовать неуклюжие околичности.” К этой проблеме было предложено немало подходов, но один поразил меня своей удивительной простотой и осмысленностью. Точно так же как ‘Я’ — это первое лицо единственного числа, независимо от рода, в этой книге будем употреблять ‘О’ как третье лицо единственного числа для обоих родов, поскольку это первая буква в ‘он’ и ‘она’. Соответственно, Е будет сокращением от ‘его’, ‘ее’ и т. п., а Н — ‘него’, ‘нее’ и т. п. Вот пример, иллюстрирующий применение всех этих форм:

О пылает к Н страстью только из-за Е тела.

•Упражнение ППП.1¹

Какое количество разных смыслов заложено в этом утверждении?

¹Ответы на все упражнения вы найдете в приложении В.

Введение О сложных видах набора

Название этой книги несколько обманчиво. У вас может возникнуть неправильное представление относительно происхождения слова \TeX , которое на самом деле составлено из трех греческих букв ТАУ ЭПСИЛОН ХИ. По словам создателя этой системы Дональда Э. Кнута, посвященные произносят \TeX так, чтобы это слово рифмовалось со словом “blecchhh” (а мы бы сказали “е-е-эх”! — Перев.). Итак, \TeX — это просто написанное прописными буквами $\tau\epsilon\chi$ — начало греческого слова, обозначающего искусство, а также корень английского слова technology и его производных. Это название, таким образом, объединяет две основные особенности \TeX : это компьютерная издательская система для набора технических текстов с большим количеством математических формул, а, кроме того, это система для получения изысканных текстов, сравнимых по своей красоте с теми, что выходят из-под руки высококлассного мастера набора.

Название \TeX явно демонстрирует и другие свои преимущества: типографскому наборщику пришлось бы, по-видимому, завести специальную отливку для получения слова “ \TeX ”, но \TeX может напечатать свое имя самостоятельно, слегка сдвинув буквы вниз и чуть-чуть приподняв их. На своем терминале, где подобные трюки немыслимы, вы можете, ссылаясь на \TeX , просто набирать ‘ \TeX ’. Это позволит отличать \TeX от множества других компьютерных программ, в название которых входит слог **TEX** (который столь же прозаически произносится, сколь и пишется).

Кроме перечисленных \TeX нических особенностей, надо отметить еще одну: \TeX позволяет делать эти вещи легко: учебник предназначен для тех, кто ничего не знает о компьютерах, и написан тем, кто знает о компьютерах столь же мало.

На самом деле последнее утверждение не вполне корректно: предполагает все же, что вы знаете, как использовать простой текстовый редактор для создания на вашем компьютере исходного файла. Не предполагается никаких специальных знаний из области полиграфии, но по мере изложения будут введены отдельные полиграфические термины. Вы можете оставаться в счастливом неведении относительно сложных правил, которыми руководствуются наборщики при воспроизведении математических формул — всеми этими правилами владеет \TeX . Также не требуется никаких математических познаний. Но нужно иметь общее представление о том, как должна выглядеть формула в полиграфически грамотно подготовленном издании. Математики и наборщики, ранее имевшие дело с математическим набором, увидят, что \TeX позволяет достичь нужного эффекта с меньшими затратами, чем это было раньше, при этом с более скрупулезным контролем качества окончательного продукта. Перед наборщиками-новичками будет поставлена двоякая цель: понять, чего хотят от них математики, и узнать, каким образом \TeX может это осуществить. Относительно второго в этом учебнике рассказано все; что касается первого — вам будет оказана посильная помощь.

Чтобы получить некоторое представление о том, как работает \TeX , рассмотрим новенькую работу из несуществующего журнала, набранную в \TeX 'е.

What Every Young Mathematician Should Know

BY LORD K. ELVIN

Abstract. We evaluate an interesting definite integral.

The purpose of this paper is to call attention to a result of which many mathematicians seem to be ignorant.

THEOREM. *The value of $\int_{-\infty}^{\infty} e^{-x^2} dx$ is*

$$\int_{-\infty}^{\infty} e^{-x^2} dx = \sqrt{\pi}.$$

Proof: We have

$$\begin{aligned} \left(\int_{-\infty}^{\infty} e^{-x^2} dx \right)^2 &= \left(\int_{-\infty}^{\infty} e^{-x^2} dx \right) \left(\int_{-\infty}^{\infty} e^{-y^2} dy \right) \\ &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} e^{-x^2} e^{-y^2} dx dy && \text{by Fubini} \\ &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} e^{-(x^2+y^2)} dx dy \\ &= \int_0^{2\pi} \int_0^{\infty} e^{-r^2} r dr d\theta && \text{using polar coordinates} \\ &= \int_0^{2\pi} \left[\int_0^{\infty} e^{-r^2} r dr \right] d\theta \\ &= \int_0^{2\pi} \left[-\frac{e^{-r^2}}{2} \Big|_{r=0}^{r=\infty} \right] d\theta \\ &= \int_0^{2\pi} \left[\frac{1}{2} \right] d\theta \\ &= \pi. \end{aligned}$$

Remark: A mathematician is one to whom *that* is as obvious as that twice two makes four is to you.

Institute For Haughty Attitudes.

Received by the editors April 1, 2000.

Research supported in part by the National Foundation.

Сначала автор написал эту статью от руки, и будучи от природы ленивым, отдал ее наборщику, чтобы получить исходный файл на компьютере. Часть этого файла представлена ниже в таком виде, который позволит здесь и в дальнейшем отличать исходный файл от того, который Т_РX дает на выходе, или от того, как выглядит основной текст этой книги. Не расстраивайтесь, если на вашей клавиатуре отсутствуют те или иные символы — у Т_РX'a имеется способ обойти это затруднение.

```
\title What Every Young Mathematician Should Know\endtitle
\author Lord K. Elvin\endauthor
```

```
The purpose of this paper is to call attention to a result
of which many mathematicians seem to be ignorant.
```

```
\proclaim{Theorem} The value of
 $\int_{-\infty}^{\infty} e^{-x^2} dx$  is
 $\int_{-\infty}^{\infty} e^{-x^2} dx =$ 
 $\sqrt{\pi}$ .
```

```
We have
```

```
$$\align
\left( \int_{-\infty}^{\infty} e^{-x^2} dx \right)^2
&= \left( \int_{-\infty}^{\infty} e^{-x^2} dx \right)
\left( \int_{-\infty}^{\infty} e^{-y^2} dy \right) \ \ \
&= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty}
e^{-x^2} e^{-y^2} dx dy
```

Даже беглого взгляда на этот файл достаточно, чтобы представить себе, как работает Т_РX. Например, все английские слова, которые появляются в этой статье, имеются в том или ином виде в исходном файле, тогда как формулы заключены либо в знаки \$ (когда эта формула встречается внутри текста), либо в знаки \$\$ (когда она выключная). Кроме английских слов и букв в формулах, имеется множество таинственных комбинаций, называемых *управляющими последовательностями*, которые начинаются с “обратной косой”, или “бэкс-лэша” \. Некоторые управляющие последовательности носят имена соответствующих специальных символов; например, \int устанавливает “знак интеграла” ∫ в выражении $\int_{-\infty}^{\infty}$, а \pi проставляет греческую букву π (пи). Другие управляющие последовательности вроде \title, \author и \proclaim, более сложны, они указывают Т_РX'у, как обработать следующий за ними текст в исходном файле.

Как только этот файл получен, несколько простых инструкций дадут Т_РX'у указание представить данную статью в стиле препринта, корректурный оттиск которой, может и не очень четкий, распечатает принтер (см. с. 14).

Удовлетворившись видом своего препринта, автор пошлет эту статью в журнал (достаточно послать лишь исходный компьютерный файл). Там внесут в этот файл единственное изменение: снабдят командой, задающей соответствующий этому журналу стиль и одновременно позволяющей получить распечатку на принтере с высоким разрешением, т. е. изготовить оригинал-макет, кото-

рый можно тиражировать.² Если журнал не принимает статью, автор может послать ее (т. е. исходный компьютерный файл) в другой журнал³, который воспроизведет ее в своем стиле, изменив для этого только одну строку.

Как видите, широта познаний \TeX 'а не может не вызывать уважения. На самом деле здесь мы будем говорить о специализированной версии \TeX 'а, известной как $\text{AMS-}\TeX$. Она была разработана Американским математическим обществом как чрезвычайно простая система для издания статей в стиле препринта, в некоторых журнальных стилях и даже для издания книг. Настоящая книга целиком посвящена системе $\text{AMS-}\TeX$, хотя зачастую мы будем называть ее просто \TeX и прибегать к полному имени лишь в тех случаях, когда речь идет о свойстве \TeX 'а, которое присуще именно $\text{AMS-}\TeX$ 'у.

Несмотря на узкую специализацию, $\text{AMS-}\TeX$ в состоянии выполнить практически все, что может когда бы то ни было понадобиться математику. Части 1 и 2 этого учебника содержат сведения о том, как набрать практически любой математический текст, с которым вам обычно приходится иметь дело; в части 3, чтение которой можно отложить до лучших времен, описаны возможности $\text{AMS-}\TeX$ 'а по набору более экзотических вещей. В частях 1 и 2 периодически встречаются экскурсы в довольно специальные вопросы; эти ответвления оформлены в отдельный абзац, начинающийся дорожным знаком



Части 1 и 2 данного учебника занимают около 100 страниц, но из этого отнюдь не следует, что $\text{AMS-}\TeX$ чрезвычайно сложен и недоступен для понимания. Чтобы облегчить ваши первые шаги в компьютерном наборе, ставилась цель не сокращать путь познания предмета, а сделать его более приятным. Причем изложение — как, например, это — сопровождалось многочисленными упражнениями, перемежающимися примерами авторского остроумия (во всяком случае ему так казалось), одобренными каламбурами, иносказаниями и иными свидетельствами дурного литературного вкуса. Вы можете пропускать все шутки, которых, к сожалению, предостаточно, но, пожалуйста, не пропускайте упражнения! Ответы к ним приведены в приложении В и касаются они, в основном, важных деталей, о которых не было повода рассказать в другом месте. Убедительно советуем сначала попробовать выполнить упражнение самостоятельно — такой способ изучения значительно эффективнее, чем пассивное чтение.

Как только вы выполните все упражнения, можете считать себя истинным пользователем $\text{AMS-}\TeX$ 'а. После этого вам, по-видимому, захочется узнать больше и о самой системе \TeX .⁴ $\text{AMS-}\TeX$ всего лишь специализированная версия \TeX 'а, но, овладев ею на должном уровне, можете считать себя причастным к \TeX нической революции.

²В настоящее время почти повсюду принята технология офсета, при которой краска переносится на обработанную фотохимическими материалами тонкую металлическую пластину. До введения фотонаборных машин наборную форму получали так: с металлического набора делался единственный оттиск, который потом использовался в процессе фотообработки пластин, а сам набор рассыпали!

³Конечно, если в нем в качестве издательской системы принят \TeX . — Прим. перев.

⁴Все сведения о \TeX 'е вы почерпнете из книги Дональда Э. Кнута *The \TeX book*, совместном издании Американского математического общества и Эдиссон-Уэсли.

WHAT EVERY YOUNG MATHEMATICIAN SHOULD KNOW

LORD K. ELVIN

April 1, 2000

ABSTRACT. We evaluate an interesting definite integral.

The purpose of this paper is to call attention to a result of which many mathematicians seem to be ignorant.

Theorem. *The value of $\int_{-\infty}^{\infty} e^{-x^2} dx$ is*

$$\int_{-\infty}^{\infty} e^{-x^2} dx = \sqrt{\pi}.$$

Proof. We have

$$\begin{aligned} \left(\int_{-\infty}^{\infty} e^{-x^2} dx \right)^2 &= \left(\int_{-\infty}^{\infty} e^{-x^2} dx \right) \left(\int_{-\infty}^{\infty} e^{-y^2} dy \right) \\ &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} e^{-x^2} e^{-y^2} dx dy && \text{by Fubini} \\ &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} e^{-(x^2+y^2)} dx dy \\ &= \int_0^{2\pi} \int_0^{\infty} e^{-r^2} r dr d\theta && \text{using polar coordinates} \\ &= \int_0^{2\pi} \left[\int_0^{\infty} e^{-r^2} r dr \right] d\theta \\ &= \int_0^{2\pi} \left[-\frac{e^{-r^2}}{2} \Big|_{r=0}^{r=\infty} \right] d\theta \\ &= \int_0^{2\pi} \left[\frac{1}{2} \right] d\theta \\ &= \pi. \end{aligned}$$

Remark. A mathematician is one to whom *that* is as obvious as that twice two makes four is to you.

INSTITUTE FOR HAUGHTY ATTITUDES

Research supported in part by the National Foundation.

Typeset by $\mathcal{A}\mathcal{M}\mathcal{S}\text{-}\mathcal{T}\mathcal{E}\mathcal{X}$

Часть 1

ЗАКУСКИ

основные ингредиенты



Глава 0. Первое знакомство; ключевая глава

Прежде чем приступить к изучению подхода к наборным процессам на базе ТрХ'а, неплохо бы взглянуть на клавиатуру вашего терминала, чтобы понять, какие символы уже имеются в вашем распоряжении. На ряде терминалов есть специальные дополнительные клавиши, но, по крайней мере, следующие символы должны присутствовать обязательно — исключения обсуждаются ниже.

Во-первых, у вас есть буквы — прописные и строчные — и цифры:

```
ABCDEFGHIJKLMNOPQRSTUVWXYZ  
abcdefghijklmnopqrstuvwxyz  
0123456789
```

Вы, вероятно, уже усвоили тот факт, что цифра 1 должна отличаться от строчной буквы l, а цифра 0 — от прописной буквы O. На экране 0 и O могут выглядеть почти одинаково, но на клавиатуре цифра 0 располагается рядом с другими цифрами.

Во-вторых, у вас есть символы

```
, . ; : ? !  
' (апостроф, или единичная правая кавычка)  
' (единичная левая кавычка)  
( ) [ ]  
/ (косая) - (дефис) * (звездочка)
```

используемые как знаки препинания, для написания чисел вроде 1,376.0003, для обозначения переносов и дефисов в сложных словах, а также для примечаний в скобках (а может быть — в квадратных скобках); в большинстве шрифтов звездочка располагается как верхний индекс* и может служить обозначением сноски. На вашей клавиатуре и экране кавычки ‘ и ’ могут выглядеть менее эстетично, вроде ` и ` . Но не следует огорчаться: ТРХ представит их должным образом. На всех терминалах имеются также и двойные кавычки " (или "), но они не перечислены в этой группе, поскольку у ТРХ'а есть свой способ формирования двойных кавычек. Обратите внимание, что - (дефис) не следует путать с другой клавишей, которая присутствует на большинстве клавиатур, _ (символ подчеркивания).

Следующая группа — несколько стандартных литер клавиатуры, обычно используемых в математических формулах:

+ = < > |

Символ | может выглядеть прерывисто как |, но ТРХ всегда даст непрерывное изображение. ТРХ обладает управляющими последовательностями для наименования всех других математических символов, включая буквы из иностранных алфавитов. Разумеется, английские буквы и цифры также используются в математических формулах, как и большинство знаков препинания. Например, / применяется для обозначения деления типа a/b , а дефис выступает в качестве минуса в формулах вроде $x - y$. ТРХ, однако, использует другие правила расстановки пробелов в математических выражениях, и знак минуса при печати отличается от дефиса. Более того, обычно в формулах используются курсивные буквы.

Четвертая группа символов, присутствующая обычно на любой клавиатуре, включает следующие:

\ { } \$ & # % @ - ~ _ "

Каждый из этих символов в АМС-ТРХ'е имеет специальное назначение; например, как уже упоминалось во введении, все управляющие последовательности начинаются с "обратной косой" (бэкслэша) \, а знаки \$ и \$\$ используются для математических формул.

Есть еще один символ, который большинство пользователей не замечают: нажатие клавиши пробела приводит к появлению на экране символа, который выглядит как пустой интервал. В тех случаях, когда необходимо подчеркнуть, что имеет место пробел, будем использовать символ

□

Имеются еще две клавиши, которые так или иначе должны быть упомянуты. Первая — клавиша возврата каретки (carriage-return), которая перемещает курсор вниз в начало следующей строки. Если нажать эту клавишу дважды, на экране появится пустая строка. Другая клавиша — клавиша табуляции TAB. Она перемещает курсор по экрану на фиксированное количество позиций, хотя в файле это обозначается одной специальной литерой. ТРХ обычно трактует клавишу TAB в точности как пробел.

*Подобно этому.

Исключения. Некоторые из этих символов бывает трудно или даже невозможно получить на вашем терминале, но Т_ЕX имеет управляющие последовательности, которые можно использовать взамен. Например, левую кавычку ‘ часто нельзя получить с клавиатуры, но вы можете использовать последовательность `\lq` для ‘ (и `\rq` для ’). Имеются также управляющие последовательности для замены символов `[`, `]`, `^`, `_` и `~`. Они вводятся по мере надобности, и их полный обзор приводится в приложении E.

Разумеется, ни одна из этих замен не сослужит вам хорошей службы, если у вас нет также и символа `\!`. Кроме того, Т_ЕX во многом полагается на фигурные скобки `{` и `}`, которым отводится особая роль и которые не могут быть заменены никакими управляющими последовательностями. Фактически, даже если на вашей клавиатуре отсутствуют символы `\`, `{` и `}`, еще не все потеряно, поскольку Т_ЕX допускает такую модификацию, при которой вместо этих символов можно использовать другие. Но подобные изменения требуют некоторых Т_ЕXнических познаний, поэтому только тот, кто сталкивался с аналогичными проблемами, сможет четко выполнить эту задачу и минимизировать путаницу в тех случаях, когда файлы, представленные в Т_ЕX'e, переносятся с машины на машину.

Если вы встречались с подобными проблемами — или даже если нет — вы, вероятно, захотите присоединиться к TUG — Т_ЕX Users Group¹ — чтобы познакомиться с другими пользователями Т_ЕX'a. О том, как установить контакт с Т_ЕX Users Group, вы прочтете в приложении H.

¹Или к CyrilTUG — Cyrillic Т_ЕX Users Group — Ассоциации пользователей кириллического Т_ЕX'a; см. рекламу в конце книги. — *Прим. перев.*

Глава 1. Лингвистические уTeXи

Хотя TeX предназначен в основном для набора книг и статей, изобилующих математическими выражениями, он также великолепно справляется с набором обычного текста. В первой части мы будем иметь дело почти исключительно с текстами, не содержащими математических формул. Это дает благоприятную возможность не спеша познакомиться со специализированным языком TeX'a и изучить способы обработки многочисленных стилевых особенностей, которыми наши друзья-литераторы так любят украшать текст.

Когда вам нужно обработать с помощью TeX'a обычный текст, следует просто набирать то, что вы хотели бы получить на выходе, оставив TeX'у все заботы о деталях. Предположим, вы вводите следующий текст:

```
This is the first paragraph of
a 1,000 word document
that has been set by TeX. The lines are
automatically justified, i.e., they are
all set to the same length.
```

```
To do this, TeX
inserts extra space between words,
and/or hyphenates words that are
too long to fit on the line. ( But it
can't correct your errors.)
```

```
Now let's get down to the nitty-gritty: how
did TeX know that it was supposed
to begin a new paragraph here? Come, come,
you really should be able to figure that
out for yourself!
```

Используя этот файл в качестве входного, TeX на выходе выдаст

```
This is the first paragraph of a 1,000 word document that has
been set by TeX. The lines are automatically justified, i.e., they are
all set to the same length. To do this, TeX inserts extra space between
words, and/or hyphenates words that are too long to fit on the line. (
But it can't correct your errors.)
```

```
Now let's get down to the nitty-gritty: how did TeX know that
it was supposed to begin a new paragraph here? Come, come, you
really should be able to figure that out for yourself!
```

[Перевод:

Это первый абзац текста, содержащего 1,000 слов, который вводится при помощи TeX'a. Строки автоматически выравниваются, т. е. они устанавливаются одинаковой длины. Для этого TeX расставляет дополнительные пробелы между словами и/или делает переносы в словах, которые не умещаются в одну строку. (Но он не исправляет ваших ошибок.)

Но теперь проникнем в суть дела: как ТрХ узнает, что именно здесь начинается новый абзац? Ну же, вы обязательно должны сами догадаться, если постараетесь!]

Заметим, что входные строки могут быть любой удобной длины; ТрХ автоматически отформатирует их в абзац, все строки которого имеют одинаковую длину. Дополнительные пробелы между словами не имеют для ТрХ'a особого значения; если во входном файле появляется последовательность пробелов, ТрХ просто игнорирует все пробелы, кроме первого. Символ возврата каретки (carriage-return) в конце каждой строки также подсказывает ТрХ'у вставить пробел, причем все дополнительные пробелы в начале следующей строки игнорируются. Так, 5 пробелов в начале строки

```
    To do this, TeX
```

были полностью проигнорированы. (Это остается в силе, даже если пробелы были получены клавишей ТАВ, поскольку обычно ТрХ обращается с ТАВ точно так же, как с пробелами.)

Для аса в машинописи эти свойства могут показаться маловажными или даже явным излишеством. Но как только вы начнете серьезно работать с текстовым редактором и использовать его для внесения правки — вставлять или удалять слова и переименовывать целые предложения — вы оцените то обстоятельство, что вам не надо заботиться о дополнительных пробелах, которые могли оказаться в тексте.

Количество пробелов после знаков препинания также несущественно. Машинистки (англоязычные) обычно любят оставлять по два пробела после точки, точки с запятой и восклицательного и вопросительного знаков, но такие украшения ТрХ'у безразличны: он пользуется своими собственными законами при расстановке пробелов после знаков препинания. Однако если вы не оставляете ни одного пробела, то и ТрХ не оставляет пробелов, так что вы можете получить на выходе выражения вроде “1,000” и “i.e.”. С другой стороны, пробел между скобкой (и словом *But* был грубой ошибкой, который вызвал нежелательный пробел на выходе. ТрХ воспринимает “(” как однобуквенное слово, и, поскольку это “слово” оказалось в конце строки, оно было отделено от слова “*But*”, которое перешло в начало следующей строки.

• Упражнение 1.1:

Входной файл английского оригинала книги содержал следующие строки:

```
First we have the upper- and
lower-case letters, and the numerals:
```

Что произошло бы, если бы вместо них было набрано следующее:

```
First we have the upper-
and lower-
case letters, and the numerals:
```

А если бы

First we have the upper- and lower-case letters, and the numerals:

Поскольку несколько пробелов в начале строки не служат указанием на начало нового абзаца, TeX обладает другим средством, на которое вы, вероятно, уже обратили внимание: TeX завершает абзац, когда сталкивается с пустой строкой, и начинает новый абзац, как только встретит что-нибудь еще, что должно быть напечатано. Разумеется, строка, выглядящая пустой на экране, на самом деле может представлять собой что-нибудь вроде

щщ

т. е. состоять из 3-х пробелов; такие строки легко образуются, если использовать текстовый редактор для удаления всех видимых частей строки. К счастью, TeX интерпретирует такую строку так же, как “пустую”. Более того, вы можете поставить в конце абзаца несколько пустых строк — TeX воспримет их как одну.

TeX позволяет довольно просто получать все четыре вида кавычек, обычно встречающихся в текстах. Для единичной кавычки достаточно нажать ‘, чтобы получить на выходе ‘, и нажать ’, чтобы получить на выходе ’. Для получения двойной кавычки нужно просто дважды набрать единичную кавычку соответствующего вида. Ввод

‘ ‘А буква ‘А’ получается вот так!’ ’

даст на выходе

“А буква ‘А’ получается вот так!”

Напоминаем, что если у вас отсутствуют эти клавиши, то вы можете использовать `\lq` для ‘ и `\rq` для ’; последовательности `\lq\lq` и `\rq\rq` дают соответствующие двойные кавычки.

Помимо единичных и двойных кавычек в текстах довольно-таки часто встречаются тире и дефисы — как в этом предложении. Дефис — это не то же самое, что знак переноса или минуса. В математических книгах, полиграфически грамотно подготовленных, вы встретите переносы, минусы и два рода тире:

короткое тире (en-dash) — и длинное тире (em-dash) —

(традиционно длинное тире (em-dash) было взято равным по ширине прописной букве М, а короткое — половине ее ширины). Короткие тире обычно используются при указании диапазона, например, “pages 13–34”, а также в контекстах, вроде “Fig. A–12”. Длинное тире используется в качестве знака пунктуации в предложениях — это те самые тире, которыми автор настоящей книги явно злоупотребляет. Эти различные символы получают следующим образом:

для переноса и дефиса набираем дефис (-);

для короткого тире набираем два дефиса (--);

для длинного тире набираем три дефиса (---).

Тире, подобно дефисам, обычно не имеют пробелов ни с одной из сторон, так что не вставляйте по невнимательности в исходный файл лишние пробелы.¹ Как уже упоминалось, дефис автоматически превращается в знак минуса, когда он используется между знаками \$, указывающими на то, что это математическая формула; в этом случае вы можете не беспокоиться по поводу пробелов, поскольку TeX имеет собственные установки относительно математических пробелов, которым он и следует.

• Упражнение 1.2:

Объясните, как был представлен следующий текст в TeX-файле

О говорит: “Я все еще набираю два пробела после точки — никак не могу отвыкнуть, но я всегда забочусь о том, чтобы в списках литературы при каком-нибудь перечислении, вроде страниц ‘480–491’, вместо дефиса стояло короткое тире.”

• Упражнение 1.3:

Что случится, если вы ошибочно наберете -- - вместо ---? А как вы думаете, что будет, если вы наберете ----?

Кроме кавычек и дефисов и всех производных от них вам может понадобиться специальный символ, что-то вроде ‘¶’, который иногда используется для обозначения абзаца. Например, вы можете захотеть перенумеровать абзацы вашего текста-образца как ¶a, ¶b и т.д.:

¶a. This is the first paragraph of a 1,000 word document that has been set by ...

¶b. Now let’s get down to the nitty-gritty: how did TeX know that it was ...

Вы добьетесь этого, набрав

```
\P a. This is the first paragraph of
. . .
\P b. Now let’s get down to the nitty-gritty: how
. . .
```

\P — это одно из многих “управляющих слов” TeX’a. Управляющее слово представляет собой комбинацию “бэкслэша” \, за которым следует любое число букв. Бэкслэш \ указывает TeX’у, что следующее за ним выражение — не текст, который надо ввести, а имя символа или другая специальная команда.

Заметим, что во входном тексте \P a имеет пробел после \P. Это существенно, потому что если пробела не поставить, то TeX решит, что мы собирались использовать управляющее слово \Pa. На выходе никакого пробела в ‘¶a’ не появится; TeX всегда воспринимает пробелы после управляющего слова просто как указание на то, где заканчивается имя управляющего слова, а не как пробел,

¹Это относится к англоязычным текстам, как, впрочем, и вся эта глава. При наборе текстов на русском языке следует, напротив, помнить о пробелах с двух сторон тире. — Прим. перев.

который следует изобразить. На самом деле имя управляющего слова не всегда заканчивается пробелом; оно может заканчиваться любым символом, не являющимся буквой. Например, если бы мы хотели перенумеровать наши абзацы ¶1, ¶2, и т. д., мы могли бы набрать \P1, \P2, и т. д. (Поскольку пробелы после управляющих слов никогда не проявляются на выходе, можно было бы также набирать \P 1, \P 2 с тем же результатом ‘¶1’, ‘¶2’. Так что вы всегда можете вставлять пробелы в конце управляющего слова, если не хотите задумываться о том, какого рода символ будет идти дальше.)

Хотя мы раньше об этом не упоминали, но для конца абзаца у ТрХ’a также есть управляющее слово. Если вы набираете

```
...
too long. ( But it can't correct errors.) \par Now let's
get down ...
```

то слова “Now let’s ...” начнут новый абзац, точно так же, как если бы перед Now let’s ... была пустая строка. Пустая строка с точки зрения ТрХ’a имеет тот же смысл, что и управляющая последовательность \par — когда ТрХ встречает пустую строку, он представляет себе, что было набрано \par.

Пустая строка — это не только дополнительное удобство для оператора — значительно проще дважды нажать клавишу возврата каретки (carriage-return), чем набирать \par. Пустые строки делают входной файл более удобочитаемым. Имейте в виду, что после первой распечатки ваш документ неизбежно подвергнется переработке, и файл придется прочесывать снова. Тогда-то вы и оцените удобство разделенного на порции файла. **ПЛАНИРОВАТЬ НУЖНЮ ЗАПЦД!**

• Упражнение 1.4:

Что произойдет, если вместо \par вы наберете \Par?

• Упражнение 1.5:

Что случится, если в вашем файле встретится

```
... end of first paragraph.\par
$$$
The second paragraph begins here ...
```

Управляющее слово \par используется не очень часто из-за принятого соглашения о пустых строках, равно как и \P — из-за того, что знак ¶ редко бывает нужен. Но есть один особый символ, который вам понадобится, если автор так бестактен, что упоминает о деньгах (настоящих долларах и центах). Предположим, что вы хотите завершить ваш небольшой документ, слегка напугавшись обещанными 1,000 словами, таким абзацем:

\$1.95 (дешево!) — такова цена этого документа.

Набрав

\$2.00 (дешево!) --- такова цена этого документа.

вы допустите не только большой коммерческий просчет, но и значительно более серьезную промашку как пользователь ТрХ’a, поскольку знак \$ указывает ТрХ’у

на начало математического выражения. В результате появится таинственная формула 2.00(дешево!) — такова цена этого документа., и затем ТЭХ пожалуется, что не может найти второго знака \$, извещающего о конце этого выражения! Поскольку \$ играет особую роль в идентификации математических формул, нужны другие способы представления знака \$ при его наборе. Можно было бы ввести управляющую последовательность вроде `\dollar`, но у ТЭХ'a имеется более короткое (и более наглядное) имя — последовательность `\$`. При ее помощи завершающий абзац можно набрать в виде

`\$1.95 (дешево!) --- такова цена этого документа.`

Обратите внимание, что `\$` не является управляющим словом — управляющее слово может содержать после `\` только буквы — но это то, что называется “управляющим символом”. Управляющий символ состоит из `\`, за которым следует *единичная небуква* (так что невозможно перепутать, где заканчивается управляющий символ). Вот примеры других управляющих символов: `\1`, `\:`, `\+` и даже `\.`

Управляющие слова и управляющие символы объединяет общее название “управляющие последовательности”; и мы, как правило, будем пользоваться именно этим термином, поскольку редко бывает нужно различать эти два вида. Есть, однако, одно важное отличие управляющей последовательности от управляющего символа. Мы уже видели, что ТЭХ никогда не обращает внимания на пробелы после управляющих слов, поскольку пробелы обычно используются только для указания конца управляющего слова. Но пробел после управляющего символа он *обязательно* воспроизведет (такая организация, кажется, отвечает чаяниям большинства операторов). Так что если вы наберете

`\$ 1.00`

то получите в результате ‘\$ 1.00’ с пробелом после \$.

Здесь следует остановиться на одном интересном моменте. Что произойдет, если вам *потребуется* пробел после управляющего слова — как, например, набрать ‘¶ 1’? Два пробела после управляющего символа `\P`, вроде `\P_ 1`, вам не помогут, поскольку ТЭХ всегда воспринимает всю последовательность пробелов как один пробел. А что действительно нужно — так это *новая управляющая последовательность для обозначения пробела*! Разумно было бы предложить управляющую последовательность `\space`, но и тут ТЭХ дает более короткое и (коль скоро вы думаете об этом) более очевидное решение. Если вы хотите дать ТЭХ'у явное указание, что вам нужен пробел, достаточно использовать комбинацию `_` (т.е. знак `\`, за которым следует пробел). Таким образом, набрав `\P_ 1`, вы получите ‘¶ 1’.

• Упражнение 1.6:

Как набрать следующее предложение?

Многие сносно печатают ‘\$\$’, гораздо труднее напечатать U.S.
 \$\$ так, чтобы они выглядели сносно.

• **Упражнение 1.7:**

AMS-TeX , не страдающий ложной скромностью, обеспечивает вас возможностью получать при помощи управляющих последовательностей $\backslash\text{TeX}$ и $\backslash\text{AmSTeX}$ слова ‘TeX’ и ‘AMS-TeX’. Как набрать следующее предложение?

AMS-TeX — это просто специализированная версия пакета TeX, но как только вы ее освоите, то приблизитесь к TeXнической революции.

• **Упражнение 1.8:**

Что получится в результате обработки $\backslash\$ \backslash \sqcup 1.00$ и $\backslash\$ \sqcup 1.00$?

Обратите внимание, что \sqcup — управляющий символ, поскольку он состоит из знака \backslash , за которым следует небуква. Решив три предыдущих упражнения, вы могли бы уже почувствовать себя докой в этом деле. Однако попытавшись разобраться в последствиях, которые повлечет за собой набор

$\backslash \sqcup \sqcup$

вы будете несколько обескуражены, поскольку на самом деле имеются два противоречащих друг другу правила. Пробел после управляющего символа не может быть проигнорирован, так что после $\backslash \sqcup$ должен получиться еще один пробел, что дает в итоге два пробела. Но существует также и общее правило, гласящее о том, что TeX обращается с последовательностью пробелов как с единственным пробелом, так что наша входная последовательность должна была бы быть эквивалентна $\backslash \sqcup$, дающей только один пробел. В этом особом случае TeX подчиняется последнему правилу: $\backslash \sqcup \sqcup$ — это то же самое, что и $\backslash \sqcup$.

Теперь мы узнали довольно много о том, как печатать при помощи TeX’a:

- (1) Несколько пробелов рассматриваются как единственный пробел, равно как и нажатие (carriage-return).
- (2) Пустые строки означают конец абзаца.
- (3) Знаки ‘ и ’ дают одинарные кавычки, ‘ ‘ и ’ ’ — двойные кавычки, -- дает короткое тире, --- дает длинное тире.
- (4) Специальные символы и команды порождают управляющими словами (например, $\backslash P$, $\backslash par$) и управляющими символами (например, $\backslash \$$, $\backslash \sqcup$), известными под общим названием *управляющие последовательности*. Пробелы после управляющих слов игнорируются, а после управляющих символов — нет, за исключением того, что $\backslash \sqcup \sqcup$ равносильно $\backslash \sqcup$.

Всем выдающимся людям хорошо известно, что иногда трудность заключается в том, чтобы скрыть от печати те или иные сведения. TeX поможет осуществить это при помощи специального приема: когда TeX встречает литеру %, он игнорирует ее вместе со всем, что следует за этой литерой в той же строке. Это позволяет вам вставлять “комментарии” в ваш файл — материал, который исчезнет при печати (даже пробелы, если вы окажетесь невнимательными). Например, если вы наберете

**материал, который исчезнет %Проверить правописание слова
% ‘изчезнет’ — стыдно выпускать учебник с опечатками
при печати (даже пробелы, если вы невнимательны)**

то получите

материал, который исчезнет при печати (даже пробелы, если вы невнимательны)

Если же вы по невнимательности наберете

**материал, который исчезнет% Проверить правописание слова
% ‘исчезнет’ – стыдно выпускать учебник с опечатками
при печати (даже пробелы, если вы невнимательны)**

то получите нечто значительно худшее, чем просто орфографическая ошибка:

материал, который исчезнетпри печати (даже пробелы, если вы невнимательны)

Это произошло потому, что символ % побуждает ТрХ проигнорировать все, что следует за этим символом, включая `(carriage-return)`.

Несмотря на то, что такое свойство символа % требует от вас повышенной внимательности, вы можете также извлечь из него и некоторую пользу: можно, например, разбив какое-нибудь длинное слово на две строки во входном файле, получить в выходном файле это же слово без разбивки. Вход

**самое длинное английское слово pneumonoultramicroscopic%
silicovolcanosconiosis**

даст самое длинное английское слово `pneumonoultramicroscopicssilicovolcanosconiosis` на выходе (уже перенесенное, если необходимо). А если вы поместите длинное тире где-то в конце входной строки англоязычного текста — что часто случается — можете использовать знак %, дабы избежать нежелательного пробела:

**if you get to an em-dash near the end of an input line---%
which frequently happens---you ...**

❖ Можно превратить в “комментарий” большой кусок текста, поместив в начале каждой строки знак %. Но АМS-ТрХ обладает для этой цели более удобным механизмом, который подробно объясняется в части 3.

❖ Когда управляющий символ `_` встречается в конце строки, это кажется несколько странным, поскольку пробела в конце строки вы фактически не видите:

**Мне не нравится эта идея ставить все эти знаки `_`
в начале абзацев.**

Мне больше нравится ставить знак % в конце, чтобы сделать знак `_` явным:

**Мне не нравится эта идея ставить все эти знаки `_` %
в начале абзацев.**

Эти завершающие литеры % способны предотвратить недоразумения, когда файлы переносятся с машины на машину, поскольку во время такой операции символы пробела в конце строки иногда исчезают. ТрХ устроен так, что это ему не повредит, однако со стоящим в конце строки символом % нет никакой необходимости даже задумываться о такой возможности.

Глава 2. Тиснем любой набор

Наши эксперименты с управляющим словом `\P` и управляющими символами `\$` и `_` были полезны при обучении тонкостям использования управляющих последовательностей, но то, о чем мы *действительно* хотели узнать — это как получать при печати курсив, хотя вы *несомненно* заметили, что в этом учебнике мы отдаем предпочтение наклонному начертанию — самому свежему веянию (для **дополнительного** выделения мы выбираем полужирное начертание). Вы можете переключиться на эти шрифты, используя управляющие последовательности `\it`, `\sl` и `\bf`; управляющая последовательность `\gm` (от слова *roman*) дает обычное прямое начертание. Таким образом, ввод

**Дружескую `\it` услугу `\gm` отличайте от
явной `\bf` услужливости `\gm` подхалима.**

дает на выходе

Дружескую услугу отличайте от явной услужливости подхалима.

Простой заменой `\it` на `\sl` в приведенном выше примере получаем слово “услугу” в наклонном начертании вместо курсивного. Могу вас обрадовать, что если вдруг *AMS-TeX*’у случайно понадобится получать текст в шрифтах иного размера, например в сносках, то автоматически будет выбран нужный размер кегля для курсивного, наклонного и полужирного начертания. Несмотря на эти удобства, вы можете негодовать по поводу включения всех этих команд `\gm`. Досадно набирать их все, но лишь одно пропущенное упоминание о начертании может привести к катастрофе: вместо одного курсивного слова у вас распечатается целая пачка страниц курсивом! На самом деле есть способ, позволяющий переключать начертание шрифта без дополнительных команд `\gm`, и для большинства целей он имеет неоспоримые преимущества. Используйте следующий ввод:

**Дружескую `{\it услуга}` отличайте от
явной `{\bf услужливости}` подхалима.**

Фигурные скобки `{}` и `}` указывают *TeX*’у, что `{\it услуга}` и `{\bf услужливости}` суть “группы”, выделенные из текста с обычным прямым начертанием. Когда *TeX* видит левую фигурную скобку `{`, которая открывает некоторую группу, он мысленно отмечает тип шрифтового начертания, который был использован, и возвращается к тому же самому шрифту, как только увидит `}`, закрывающую группу — правая фигурная скобка отменяет команду `\it`.

Разумеется, если вы забудете закрыть скобку `}`, которая отменяет команду `\it`, то получите кипу страниц, отпечатанных курсивом. Забытая закрывающая скобка `}` при некоторых обстоятельствах может привести даже к еще более страшным последствиям. Этого легко избежать, если на вашем терминале имеются дополнительные клавиши, которые можно запрограммировать таким

образом, чтобы получать на выходе любую последовательность, либо воспользоваться текстовым редактором, устроенным так, что нажатие единственной клавиши (обычно одной из управляющих) дает тот же эффект. Вы можете сделать так, чтобы один удар по клавише давал на выходе последовательность типа `{\it}` и затем возвращал курсор в позицию перед скобкой `}`; затем вы набираете внутри нужный текст, который хотите выделить курсивом, и перескакиваете через закрывающую скобку `}`, когда этот текст кончается. Таким образом, вы чуть-чуть сэкономите на наборе и не забудете поставить закрывающую скобку `}`.

Поскольку использование скобок позволяет отменять те или иные действия \TeX 'а, выделение групп бывает полезно и во многих иных ситуациях. Например, если мы хотели получить на выходе ‘ \llcorner a’ с пробелом после \llcorner , то раньше набирали `\R\llcorner a`. Но можно также набрать `{\R} a`. Теперь за управляющим словом `\R` следует `}`, а не пробел, и пробел после `}` не будет проигнорирован. Чтобы использовать этот прием, мы помещаем `\R` целиком в группу, но это не несет никакой дополнительной смысловой нагрузки — в этой группе ничего особенно интересного не происходит.

На самом деле есть другой способ — использовать фигурные скобки для получения ‘ \llcorner a’, который может показаться даже более хитроумным. Мы можем набрать

$$\backslash R\{ } a$$

используя пустую группу: комбинация `{ }` представляет собой группу, в которой нет литер, так что она ничего не произведет на выходе, но будучи вставленной между `\R` и пробелом, она пресечет миссию \TeX 'а искать и уничтожать дополнительные пробелы. Пустая группа `{ }` может показаться дешевым трюком, но он довольно изящно вписывается в математические формулы и иногда просто незаменим.

• Упражнение 2.1:

Снова выполните упр. 1.7 двумя различными способами, но уже без использования управляющего символа `\llcorner`.

В качестве последнего поединка с группированием здесь помещены два довольно техничных упражнения, предназначенных для того, чтобы заставить вас задуматься над этими вещами, а не просто проиллюстрировать реально встречающиеся.

• Упражнение 2.2:

Что даст на выходе последовательность `{\R\llcorner}\llcorner\llcorner\llcorner}{\llcorner}a`?

• Упражнение 2.3:

Что следует набрать, если вы хотите на выходе получить два дефиса подряд? (Не следует набирать `--`, но есть множество решений).

В нашей дискуссии относительно курсивного и наклонного шрифта мы упустили один момент, важность которого сразу оценим, если обратим внимание,

что *переход* от курсива к прямому шрифту сопровождается разными пробелами. Заметим, что пробел после “*переход*” в предыдущем предложении кажется очень маленьким, так как *д* склонилось над пробелом. В предыдущем примере это было незаметно, потому что курсивное *у* не так склоняется над строкой, как курсивное *д*. Размер выступа буквы вправо называется “поправкой на курсив”, и `TeX` позволяет вставлять дополнительное пространство в таких случаях при помощи управляющего символа `\/`, который должен быть набран после последней курсивной буквы. Таким образом, правильный *переход* от курсивного к прямому дал бы следующий набор:

... правильный `{\it переход\/}` от ...

Общее правило состоит в том, что нужно применять `\/` после группы слов наклонного начертания, если только за ней не стоит запятая или точка.

• Упражнение 2.4:

Объясните, как набрать следующее:

Восхитительный `TeX` объясняет, как вставлять в исходный файл текст, который исчезнет [*sic*] при печати.

(Если у вас нет [и], можете воспользоваться вместо них `\lbrack` и `\rbrack`.)

• Упражнение 2.5:

Как лучше всего набрать в `TeX`'е следующее предложение?

Закурсивленное *одно* слово или даже *префикс* или *суффикс* слова смотрятся неплохо, но не *перебарщивайте*.

• Упражнение 2.6:

Как набрать следующее:

Большинство грамотеев скажут, что знаки препинания следует ставить в том же начертании, что и *основной текст*; но многие пишущие предпочитают *переключиться*; это особенно верно в случае точки с запятой.

• Упражнение 2.7:

Какую неприятную ошибку вы могли бы совершить при наборе такого текста?

Заметим, что пробел после “*переход*” кажется слишком маленьким.

• Упражнение 2.8:

Как набрать такую библиографическую ссылку:

W. Ambrose, *Higher order Grassmann bundles* 3 (1964), 199–238.

Ответ к упр. 2.8 может вас напугать, пока вам не придет в голову мысль, что внутри группы можно делать то же самое, что и вне ее. На самом деле группу даже можно поместить внутрь другой группы. Когда мы начнем набирать математические формулы, этот прием будет использоваться даже слишком часто, а пока что просто запомните его.

• Упражнение 2.9:

Объясните, как набрать прямое слово внутри закурсивленного предложения.

Несмотря на все разнообразие возможностей, предоставляемых набором *курсивом*, *наклонным* и *полужирным* шрифтами, всегда найдется энтузиаст, который не может обойтись без подчеркнутых слов. Т_ЭX не снабжает вас удобным способом подчеркивания — нельзя набрать что-нибудь вроде `{\ul ...}`, чтобы подчеркнуть (*underline*) целую группу слов. Если художник книги замыслит ее так, чтобы подчеркивание в ней использовалось интенсивно, E придется создать специальный шрифт ‘\ul’, в котором подчеркивание заложено непосредственно в каждую литеру; разумеется, такие буквы, как g, j, p, q и y (а также русские p, y и ф — *Перев.*) создадут E дополнительные трудности. Т_ЭX, однако, предоставляет возможность подчеркнуть все слово. Если вы наберете `\underbar{трудности}`, то получите трудности, демонстрирующие один из способов преодоления этих трудностей.

Перечеркнув нестандартные ситуации, можем утверждать, что подчеркивание `\underbar` не столь уж важная команда. Но она важна с точки зрения использования фигурных скобок в `\underbar{трудности}`, потому что действие `\underbar` отличается от действия управляющих последовательностей, которые мы до сих пор изучали. Управляющие последовательности `\P`, `\$`, `_` и `\V` дают специальные символы или пробел некоторого размера, а управляющие последовательности `\par`, `\it`, `\sl` и `\bf` указывают Т_ЭX’у, что нужно изменить его действия (закончить абзац или сменить шрифт). Команда `\underbar` говорит Т_ЭX’у о том, что некоторую часть входного текста нужно обрабатывать особым образом. На Т_ЭX’ническом жаргоне `\underbar` представляет собой “управляющую последовательность с аргументом”, причем под аргументом подразумевается часть входного текста, которая получает специальную обработку. Если бы мы набрали, `\underbar не снабжает`, то получили бы “не снабжает”, потому что Т_ЭX решил бы, что только одна буква *n* является аргументом `\underbar`: если аргумент управляющей последовательности состоит более чем из одной литеры, его следует заключать в фигурные скобки, чтобы дать понять Т_ЭX’у, что вся группа является аргументом.

Большинство управляющих последовательностей, которые мы будем изучать в дальнейшем, тоже указывают Т_ЭX’у, что нужно сделать с вводом нечто специфическое. На самом деле у Т_ЭX’а столько управляющих последовательностей с аргументами, что новички Т_ЭX’нического набора иногда забывают, что `\it`, `\sl` и `\bf` к таковым не относятся. Распространенная ошибка выглядит примерно так:

... закурсивьте только `\it {одно}` слово в предложении.

При этом ожидается, что будет закурсивлено только *одно* слово. Но происходит совсем другое! Скобки вокруг `{одно}` не действуют вовсе: управляющая последовательность `\it` просто велит Т_ЭX’у начать использовать курсивный набор, и слово “одно”, *вместе со всеми словами, следующими за ним*, курсивится. Когда вы набираете все правильно, т. е.

... закурсивьте только `{\it одно}` слово в предложении.

действие команды использования курсива ограничивается только лишь группой `{\it одно\}`, в которой он нужен. Чтобы оградить себя от такого рода ошибок, следует про себя проговоривать команду `\it` как “переключить на курсив”. (Такого рода проблем у вас, разумеется не возникнет, если вы выделили особую клавишу, нажатие которой приводит к `{\it }`.)

• **Упражнение 2.10:**

Наберите следующее:

Если все предложение набрано наклонным шрифтом, можно подчеркнуть для выделения одно слово в прямом начертании.

Подошло время собрать воедино все, что мы изучили, и посмотреть, что же \TeX на самом деле делает с нашим файлом. Предполагается, что вы решили все задачи, встретившиеся в этой и предыдущей главах, а ответы сверили с приложением В. Последнее упражнение этой главы несколько отличается от предшествующих, потому что ставится задача создания компьютерного файла, который в следующей главе будет представлен \TeX 'у для обработки.

Для большинства компьютеров имя файла представляется двумя частями, разделенными точкой, наподобие `'file.ext'`, где часть после точки называется расширением (от слова “extension”). Если на вашем компьютере дело обстоит точно так же, то файл для последнего упражнения должен называться `'paper.tex'`. Название `'paper'` первой части не существенно, но вторая часть — расширение, обязательно должна быть `'tex'`. (Если на вашем компьютере дело с именами файлов поставлено по-другому, проконсультируйтесь с вашим системным программистом относительно дополнительных деталей.)

• **Упражнение 2.11:**

Создайте компьютерный файл `'paper.tex'`, после обработки которого \TeX 'ом получим на принтере следующее:

Everyone will be happy when \TeX finally arrives, and Dr. Treemunch can start typing his own so-called scholarly manuscripts. His last *opus*—which The Amer. Jour. Recr. Drugs recently published—created quite a sensation, particularly the material that he has expounded on pages 22–23. Including this material cost an extra \$1,000, but it did make ¶¶ 1 and 2 quite popular.

The head of the department, our university's own I. M. Stable, attributes Treemunch's recent aberrant behavior to the much-publicized “research” for his paper; notwithstanding, others say that Treemunch's name isn't on the computers' databases, so his name often gets hyphenated quite strangely, which may account somewhat for his feeling of being ill-used.

Глава 3. Ваш первый Т_ЭХ-практикум

То, что было сделано до сих пор, — это всего лишь поверхностная Т_ЭХ-инспекция, тем не менее пора передохнуть, и пусть теперь потрудится компьютер. Предыдущая глава завершалась файлом `paper.tex`. Пусть теперь Т_ЭХ обработает этот файл, и посмотрим, что получится. Мы постараемся как можно подробнее рассмотреть различные этапы этого процесса, но будет значительно полезнее для вас, если вы проделаете все это на своем терминале. Прежде чем к этому приступать, убедитесь, что вы правильно составили входной файл `paper.tex`, сверившись с ответом к упр. 2.11.

Следует отметить, что точные детали Т_ЭХ-обработки зависят от того, какой операционной системой вы пользуетесь. [Поскольку Т_ЭХ'у может понадобиться изрядное количество машинных ресурсов, ваша машина может потребовать даже, чтобы Т_ЭХ-задания выполнялись в пакетном режиме, без диалога с пользователем¹.] Подобные вещи смогут объяснить эксперты, обслуживающие вашу ЭВМ.

Прежде всего нужно кое-что добавить в файл `paper.tex` — он еще не вполне соответствует тому виду, с которым АМ_С-Т_ЭХ умеет оперировать. В самом начале файла следует добавить две строки:

```
\input amstex
\documentstyle{amspt}
```

Первая предлагает Т_ЭХ'у прочесть некоторый файл, в котором описаны особенности АМ_С-Т_ЭХ'а. Вторая строка задает некий стиль вывода (формат страницы, тип шрифта и другие типографские тонкости). Здесь мы запросили дежурный стиль, называемый 'amspt' (AMS preprint style). Если бы мы набрали

```
\documentstyle{bul}
```

это означало бы, что Т_ЭХ'у следует печатать текст в стиле 'bul' — т. е. в стиле *Bulletin of the American Mathematical Society*.

(Заметим, что в первой строке мы не написали `\input{amstex}`. Команда `\input` — одна из фундаментальных управляющих последовательностей Т_ЭХ'а, и Т_ЭХ знает, что ее аргументом является все вплоть до первого пробела или (carriage-return) (возврата каретки). По Т_ЭХническим причинам `\input` работает особым образом — фигурные скобки вокруг `amstex` не только излишни, они являются ошибкой! К счастью, управляющая последовательность `\input` есть единственное исключение такого рода, так что в дальнейшем вы можете больше об этом не заботиться.)

Обычно статья начинается с названия и фамилии автора. Эти данные следует поместить сразу после строки `\documentstyle`. Пока отложим это и проинструктируем Т_ЭХ о том, что сразу далее следует тело документа, добавив строку

```
\document
```

¹Здесь речь идет о работе на больших, многопользовательских ЭВМ. — Прим. ред.

Теперь собственно текст начинается с 4-й строки. Наконец, в самом конце файла добавим строку

```
\enddocument
```

Как вы догадываетесь, это говорит Т_EX'у о том, что документ завершен и работу следует закончить — все последующие строки будут проигнорированы.

Теперь файл `paper.tex` завершен, и можно выйти из редактора. Вы снова в диалоге с операционной системой. Она выдает вам символ запроса — для определенности пусть это будет символ `@` — и теперь можно продолжать.

Прежде всего следует попросить операционную систему выполнить Т_EX. Для этого наберите `TeX`, или `tex`, или `run tex`, или что-то вроде этого. После соответствующего заклинания, в которое входит и `(carriage-return)` в конце строки, Т_EX выдает сообщение

```
This is TeX, Version 3.0 (preloaded format=plain 90.6.1)
**
```

Символы `**` представляют собой запрос, поступающий от Т_EX'а, относительно имени файла, который вы собираетесь обрабатывать. Здесь вам следует набрать слово `paper`. (Прежде чем нажать `(carriage-return)` после `paper`, можно исправить ошибки набора при помощи клавиши `DELETE` или другой клавиши, которой вы обычно пользуетесь). Заметим, что мы молчаливо предполагаем, что набор `paper` влечет за собой обработку файла `paper.tex`; Т_EX считает, что файл имеет расширение `tex`, если не указано иного расширения.

Во многих системах такой двухступенчатый процесс можно сократить: набор

```
tex paper
```

после запроса `@` даст тот же результат. Какой бы способ вы ни использовали, стоит вам нажать `(carriage-return)` после `paper`, как Т_EX начнет обрабатывать `paper.tex`, и на вашем экране порой может появиться нечто вроде следующего:

```
(paper.tex (amstex.tex)
```

```
AmS-TeX - Version 2.0
```

```
COPYRIGHT 1985,1990 - AMERICAN MATHEMATICAL SOCIETY
Use of this macro package is not restricted provided
each use is acknowledged upon publication.
```

```
) (amspt.sty
```

```
AMSPT.STY - Version 2.14)
```

```
(amssym.tex)
```

```
Overfull \hbox (9.63954pt too wide) ...
\tenrm notwith-stand-ing, oth-ers say ... databases,|
```

В первой строке говорится, что Т_ЭX начал читать `paper.tex` и что встретившиеся ему команды потребовали сначала прочесть файл `amsrpt.tex`, где хранятся характерные особенности¹ АМ_S-Т_ЭX'а, и затем прочесть файл `amsrpt.sty`, где подробно описывается стиль `amsrpt`. При выводе на экран этой строки Т_ЭX может показаться чрезвычайно медлительным, так как чтение файлов предполагает обращение к операционной системе. Вы можете также заметить, что Т_ЭX читает непонятным вам образом и `amssym.tex`. Этот файл читается автоматически из `amsrpt`; с подробностями вы сможете ознакомиться в дальнейшем.

Как только Т_ЭX завершит чтение этих файлов, он, по-видимому, выдаст следующие, куда более загадочные строки, но уже очень быстро:

```
Overfull \hbox (9.63954pt too wide) ...
\tenrm notwithstanding, oth-ers say ... databases,|
```

В этих строках, видимо, сообщается, что здесь что-то не так, но прежде чем их расшифровывать, следует дождаться результата.

Затем появляется [1], говорящая о том, что Т_ЭX достаточно поработал над вашим файлом, чтобы выдать одну страницу, которая имеет номер "1"; другие номера не появляются, поскольку вся статья помещается на одной странице.

В этом месте Т_ЭX создает и помещает в ваш директорию новый файл с именем `paper.dvi`. Это — *device-independent* (не зависящий от устройства) файл, который может быть использован с соответствующей сопровождающей программой для вывода на печатающее устройство вашего компьютера, чтобы получить твердую копию. На самом деле на экране, вероятно, появится что-то вроде

Output written on paper.dvi

сообщающее вам об этом, возможно, с другими строками из 'transcript file' под именем `paper.lst` или `paper.log`, которыми мы займемся в дальнейшем.

В настоящий момент нас беспокоит только, как получить распечатку на принтере. В вашем компьютере должна быть другая программа, которая берет файл `paper.dvi` и посылает ее на принтер; следует проконсультироваться с системным программистом о том, как именно получать распечатку. Вам, по-видимому, захочется внимательно изучить эту распечатку — здесь вас ожидают некоторые неожиданности — так что вам следует приняться за это как можно скорее.

Если файл `paper.tex` не содержит ошибок, ваша распечатка должна отличаться от оригинала на с. 30. Вместо нее получится что-то вроде

Everyone will be happy when Т_ЭX finally arrives, and Dr. Treemunch can start typing his own so-called scholarly manuscripts. His last *opus*—which The Amer. Jour. Recr. Drugs recently published—created quite a sensation, particularly the material that he has expounded on pages 22–23. Including this material cost an extra \$1,000, but it did make ¶¶ 1 and 2 quite popular.

¹На некоторых компьютерах может существовать отдельная программа `amstex`, в которой особые свойства АМ_S-Т_ЭX'а уже добавлены к Т_ЭX'у. Если вы пользуетесь такой программой, то нет никакой необходимости в строке `\input amstex`, и если она там окажется, то будет в сущности проигнорирована.

The head of the department, our university's own I. M. Stable, attributes Treemunch's recent aberrant behavior to the much-publicized "research" for his paper; notwithstanding, others say that Treemunch's name isn't on the computers' databases, so his name often gets hyphenated quite strangely, which may account somewhat for his feeling of being ill-used.

Вы сразу поймете, почему появлялось сообщение `Overfull \hbox` (переполнение горизонтального прямоугольника). Слово "databases" (вместе с запятой после него) несколько вылезает за правое поле — так что `'Overfull \hbox'` представляет собой всю строку

notwithstanding, others say that Treemunch's name isn't on the computers' databases,

В сообщении `'Overfull \hbox'` сначала говорится о том, что ширина этого прямоугольника превышает норму на 9.63954pt; здесь под pt понимается "пункт" принтера, примерно равный 1/72 дюйма. 9.63954pt — это вполне заметная величина, и слово, которое даст такое переполнение с правого края, будет выделиться. Чтобы быть уверенным, что мы действительно это заметим, Т_ЭХ помещает большую черную пометку непосредственно после выступающего слова, так что наша первая попытка набора безусловно не дала того результата, который хотелось бы взять домой и показать своим друзьям. Информация об этом переполнении также пересылается на ваш терминал: следующие строки

```
\tenrm notwith-stand-ing, oth-ers say ... databases,|
```

описывают `Overfull \hbox` более подробно — сначала появляется `\tenrm`, название используемого шрифта (ten point roman), затем появляется приличная порция самого текста и, наконец, — последнее слово строки `databases`, за которым следует `|`, означающий черный прямоугольник, проставляемый Т_ЭХ'ом в распечатке.

Причиной этого печального события служит то обстоятельство, что Т_ЭХ не умеет переносить слово "databases". У Т_ЭХ'а есть весьма эффективная процедура нахождения точек возможных переносов, но абсолютно все он находить не умеет, и именно это и произошло со словом "databases". Кроме того, Т_ЭХ не желает перенести все это слово на следующую строку, поскольку тогда на этой строке будет слишком просторно:

notwithstanding, others say that Treemunch's name isn't on the computers'

что Т_ЭХ считает недопустимым.¹ Т_ЭХ скорее выдаст нечто, что вы сочтете недопустимым, вынуждая таким образом вас решать эту проблему самостоятельно.

¹На самом деле не сам Т_ЭХ а стиль `amsrpt` определяет, насколько далеко Т_ЭХ может растянуть строку; текст на с. 30 был получен при помощи модифицированного стиля `amsrpt`, который допускал большее вытягивание строки.

Итак, столкнувшись с необходимостью решать эту проблему, мы можем выбрать такой *sl* вариант — заставить Т_EX вопреки его желанию разорвать строку. Для этого необходимо набрать `\linebreak`. Таким образом, набрав

```
name isn't on the computers'\linebreak databases,
```

мы вынуждаем Т_EX разорвать строку перед словом “databases” и растянуть эту строку до правого поля. Если вы оставите пробел перед `\linebreak`, АМ_S-Т_EX просто решит, что вы не хотите этого пробела и все равно растянет строку до правого поля.

• Упражнение 3.1:

Добавьте `\linebreak` к `paper.tex` и обработайте файл Т_EX’ом, чтобы увидеть, что получится. Насколько плохо с точки зрения Т_EX’a разрывать строку таким образом? (Вам не следует беспокоиться об удалении файла `paper.dvi`; Т_EX удалит его, когда понадобится записать новый *dvi*-файл для измененного `paper.tex`.)

Безусловно, должно существовать лучшее решение нашей проблемы, чем просто заставлять Т_EX разрывать строку таким образом; у нас должен быть способ подсказать Т_EX’у как переносить “databases”. В самом деле, мы можем сделать это, записав вместо `databases`

```
data\~bases
```

Особый символ `\~` представляет собой “возможный перенос”, который говорит Т_EX’у, что перенос может быть сделан в этом месте, но это не обязательно. (Так что если мы впоследствии изменим некоторые слова, переноса в нежелательном месте не будет.)

Пропуск переноса в “databases” был обидным промахом со стороны Т_EX’a, Но: “не сделаешь — плохо, сделаешь — еще хуже”, так что Dr. Treemunch пришел бы, несомненно, в неопишемую ярость, увидев, как перенесена его фамилия во втором абзаце. Мы можем выйти из положения, набрав

```
Tree\~munch
```

Здесь `\~` не только сообщает Т_EX’у о возможной точке переноса, но и предотвращает иные варианты, поскольку Т_EX никогда не включает какие-либо еще переносы в слова, уже содержащие дефисы или возможные переносы,

• Упражнение 3.2:

Измените файл `paper.tex`, удалив `\linebreak` и вставив возможные переносы в `databases`, `Treemunch` и `Treemunch's`, и снова прогоните через Т_EX, чтобы посмотреть, что из этого выйдет.

Хотя возможный перенос `\~` спасет положение в тот раз, когда появится anomальное слово, было бы весьма неудобно снова и снова вставлять в то же самое слово возможные переносы. Если жалобы относительно д-ра Treemunch занимают несколько страниц, следует добавить правильный перенос его фамилии в базу Т_EX’a для всей статьи, набрав просто в начале файла:

```
\hyphenation{Tree-munch}
```

• **Упражнение 3.3:**

Удалите все возможные переносы \- из `paper.tex` (те, что вы включили в слова `data\~bases`, `Tree\~munch` и `Tree\~munch's`) и добавьте в начале файла

```
\hyphenation{Tree-munch}
\hyphenation{data-base}
```

(скажем, сразу после `\document`). Специально для этого упражнения мы явно не задаем Т_EX'у перенос для форм “Treemunch’s” и “databases”. Обработав `paper.tex` Т_EX'ом снова и увидев, что из этого вышло, обратитесь для большей ясности к ответам.

Наконец, поступим совсем правильно и дадим правильные переносы в словах “Treemunch”, “database” и “databases”. Нет необходимости для каждого из них задавать отдельно `\hyphenation`. Все это можно объединить в одной строке

```
\hyphenation{Tree-munch data-base data-bases}
```

• **Упражнение 3.4:**

Замените старые `\hyphenation`-строки этой одной новой строкой и снова пропустите `paper.tex`.

Теперь, когда вы знаете, как получать Т_EXовский вывод, вы можете делать упражнения по-новому. Вместо того чтобы выписывать каракули на клочке бумаги и после обращаться к приложению В, вы можете эти же каракули вбить в компьютер и посмотреть, что Т_EX с ними сделает. Разумеется, вам захочется сохранить эти ответы для одной-двух глав, вместо того, чтобы прогонять через Т_EX каждое упражнение, иначе вам пришлось бы потратить слишком много машинного времени. Но рано или поздно (а возможно — уже) вы допустите несколько ошибок, которые озадачат Т_EX даже больше, чем вас. Так что следующий шаг как раз и будет посвящен разбору сообщений об ошибках, поступающих от Т_EX'а.

Глава 4. ТрХнический контроль ошибок

Файл `paper.tex` был специально создан таким образом, чтобы получались неправильные переносы и возникали сообщения о переполнении `Overfull box`. Кроме того, общеизвестно, что первый блин обычно выходит комом. В реальности, однако, проблема переносов возникает не столь уж часто, так что сообщение `Overfull box` появляется довольно редко.

Прежде чем вы превратитесь в настоящего адепта ТрХ'а, вам придется, по-видимому, не раз столкнуться с сообщениями об ошибках (error messages); они появляются, когда вы делаете что-то такое, что приводит ТрХ в замешательство: он останавливается и требует разъяснений. Вам нет необходимости слишком напрягаться, когда вы столкнетесь с сообщением об ошибке, поскольку ТрХ можно всегда уговорить сделать то или иное предположение относительно ваших намерений. Вы можете рассматривать сообщение об ошибке как возможность постичь тайные помыслы ТрХ'а. Некоторым людям нравятся такие психоаналитические опыты над компьютером, но если вас это не привлекает, то вы всегда можете заставить ТрХ заняться разгадыванием ваших мыслей.

Оставим пока наш файл `paper.tex` в покое и начнем создавать другой файл, скажем, `goof.tex`, в который специально включим несколько ошибок:

```
\input amstex
\documentstyle{amspt}
\document
Here is a word in {\It italics\/} and here is one
in {\Bf boldface} and one in {\It italics\/} again.
\enddocument
```

[Перевод: Здесь одно слово выделено курсивом, одно – полужирным шрифтом и одно — снова курсивом.]

Как и в предыдущей главе, мы неоднократно будем вносить изменения в этот файл. Если вы будете делать эти изменения в вашем файле `goof.tex`, следуя за нами, то сможете прямо на экране наблюдать описанный здесь результат.

Прежде всего постараемся прогнать первоначальную версию `goof.tex` через ТрХ. Как только мы это сделаем, ТрХ завопит:

```
! Undefined control sequence.
1.4 Here is a word in {\It
                               italics\/} and here is one
?
```

В верхней строке, начинающейся с `!`, содержится сообщение об ошибке, в котором говорится, что ТрХ встретился с управляющей последовательностью, которая ему незнакома. Дополнительная информация содержится в следующих двух строках. В первой мы можем видеть неопределенную управляющую последовательность `\It` вместе с самой строкой (частью строки), предшествующей ей. Дальнейший выходной текст печатается строкой ниже, так что легко

явно увидеть, как далеко зашел Т_ЭХ во входном файле, прежде чем обнаружил ошибку.

Наконец, Т_ЭХ напечатал знак ? на следующей строке, означающий, что он спрашивает нас, что же делать. В этот момент у нас есть несколько возможностей. Самое малодушное — это нажать X или x с последующим (carriage-return), означающее выход (exit) из Т_ЭХ'а. Затем мы могли бы вернуться к нашему файлу, исправить ошибку, постараться снова прогнать файл через Т_ЭХ, обнаружить следующую ошибку, вернуться в файл, исправить эту ошибку и т. д., и т. п. Вместо того чтобы сразу капитулировать, можно просто нажать (carriage-return). Тогда Т_ЭХ сам постарается исправить ошибку наилучшим образом; в данном случае он просто проигнорирует управляющую последовательность \It и продолжит работу. Разумеется, это означает, что слово “italics” не будет выделено курсивом, но пока не следует особенно об этом тревожиться, поскольку мы, вероятно, найдем еще немало ошибок и не следует стараться с первого раза что-нибудь напечатать.

Несомненно, мы вскоре получим следующее сообщение об ошибке

```
! Undefined control sequence.
1.5 in {\Bf
          boldface} and one in {\It italics\} again.
?
```

смысл которого, должно быть, ясен. На этот раз попробуем более напористый прием: постараемся вставить правильную управляющую последовательность \bf. Для этого следует нажать i или I (от ‘Insert’), а затем (carriage-return), чтобы Т_ЭХ понял, что мы хотим что-нибудь вставить. (Давайте попробуем!) Т_ЭХ откликнется на это так:

```
insert>
```

Теперь надо набрать \bf и нажать (carriage-return). Как видите, Т_ЭХ выполнит это и тут же выдаст последнее сообщение об ошибке:

```
! Undefined control sequence.
1.5 in {\Bf boldface} and one in {\It
          italics\} again.
?
```

На этот раз просто нажмем x и выйдем, с тем чтобы изучить еще несколько типов ошибок.

Если вы вернетесь в редактор, то увидите, что в ваш файл \bf вставлено не было: Т_ЭХ просто вставил \bf в свою собственную копию вашего файла. В файле *goof.tex* все старые ошибки так и остались, но пока их не исправляйте, а добавьте еще одну: уберите скобку { перед последним \It, чтобы получилось

```
in {\Bf boldface} and one in \It italics\} again.
```


Теперь снова запустим Т_ЭХ. Естественно, сначала мы столкнемся со старой ошибкой:

```
! Undefined control sequence.
1.4 Here is a word in {\It
                                italics\} and here is one
?
```

Здесь мы вставили исправление `\it`, но сделали это только для одного шага. Наберем

```
i\it    или    \it
```

и затем нажмем (carriage-return). Точно так же исправим две другие ошибки.

Наконец Т_ЭХ доберется до четвертой ошибки:

```
! Too many }'s.
1.5 in {\Bf boldface} and one in \It italics\}
                                again.
?
```

Это сообщение об ошибке — слишком много скобок } — не требует разъяснений, но иногда может понадобиться дополнительная помощь. Тогда достаточно нажать `h` или `H` (от ‘Help’) и затем (carriage-return), и в такой ситуации Т_ЭХ ответит:

```
You've closed more groups than you opened.
Such booboos are generally harmless, so keep going.
```

[Перевод: Вы закрыли больше групп, чем открыли. Такие проделки обычно безвредны, так что продолжайте.]

Итак, последуем совету Т_ЭХ’а: не надо здесь ничего вставлять — время для вставки нужной скобки { давно уже прошло — так что двинемся дальше, нажав снова (carriage-return). В этом случае Т_ЭХ предпримет единственно возможную акцию: он просто удалит скобку } в надежде на лучшее. Теперь Т_ЭХ несомненно доберется до конца файла и создаст `dvi`-файл.

Нам не следует распечатывать этот `dvi`-файл, поскольку наша последняя ошибка была исправлена не так, как надо: удаление скобки } привело к тому, что слово “again” тоже оказалось напечатано курсивом. Теперь пора вернуться и исправить все ошибки разом. Вы их успели забыть? Не беспокойтесь! Существует “файл транскрипции” `goof.lst`, которым и надо воспользоваться; вы обнаружите, что все сообщения об ошибках, которые возникали на экране, там записаны, так что можете на досуге их изучить.

• Упражнение 4.1:

Обратитесь к файлу `goof.lst`, чтобы узнать, какая дополнительная информация там содержится. Исправьте все ошибки в `goof.tex` и снова прогоните его через Т_ЭХ чтобы убедиться, что больше ошибок нет.

• Упражнение 4.2:

Удалите скобку { перед последним `\it`, нажмите `h` или `H`, как только получите сообщение об ошибке, и затем снова нажмите ее, и посмотрите, что получится.

• **Упражнение 4.3:**

Верните обратно скобку {, но на этот раз удалите пробел после первого \it, чтобы получилось

```
{\ititalics\}/...
```

Какое сообщение об ошибке вы получите и как будете на него реагировать?

• **Упражнение 4.4:**

Исправьте предыдущую ошибку, но теперь уберите последнюю *правую* скобку }, чтобы получилось что-то вроде

```
... and one in {\it italics/ again.
\enddocument
```

со скобкой {, не имеющей парной }. Прогоните этот файл через Т_ЕХ и посмотрите, что произойдет. Если вы получите такое сообщение, имея дело с большим файлом, то как вам следует поступить?

Хотя в этой главе мы изучили несколько типов ошибок, они все очень просто диагностировались. Этого и следовало ожидать, поскольку на данном этапе мы можем набирать только очень простые вещи и, следовательно, делать только простые ошибки. Когда мы обратимся к набору более интересного материала, тогда нам придется заняться, возможно иногда непреднамеренно, более интересными ошибками, которые могли быть сделаны именно тогда, когда мы сконцентрировали все наше внимание на том, чтобы их не совершить.

◊ Вот несколько иных подробностей, касающихся сообщений об ошибках, выдаваемых Т_ЕХ'ом, о которых вам должно быть интересно узнать.

• **Упражнение 4.5:**

Сбалансируем скобки, но на этот раз заменим первое \it на \It еще раз. Как только Т_ЕХ оглушит вас сообщением об ошибке, ответьте ему тем же, нажав ! и затем (carriage-return), и посмотрите, что из этого выйдет.

• **Упражнение 4.6:**

Удалите в файле `goof.tex` самую последнюю строку `\enddocument` и посмотрите, что произойдет, когда вы прогоните этот файл через Т_ЕХ. Разберитесь в том, как довести дело до славного конца, сверившись с приложением В.

• **Упражнение 4.7:**

Посмотрите, что произойдет, если вы напутаете при наборе имени статьи. Попробуйте, например,

```
@tex goofy
```

(предполагается, что файла с именем `goofy.tex` не существует).

• **Упражнение 4.8:**

Попробуйте обратиться к А_МС-Т_ЕХ'у с вашим собственным (еще не существующим) стилем `mystyle`, изменив

```
\documentstyle{amsppt}
```

на

```
\documentstyle{mystyle}
```

и посмотрите, что из этого выйдет.

Глава 5. Разделяющие пробелы и связывающие тильды

Несмотря на то, что в файле `paper.tex` проблема переносов уже разрешена, он еще не вполне готов. Почему? Потому что опытный глаз тут же заметит, что пробелы в этом тексте не соответствуют требованиям типографского набора высокого качества, где пробелы между предложениями должны быть больше, чем пробелы между словами в предложениях.¹ Это улучшает текст внешне и облегчает чтение. `TeX` также следует этим правилам, но он не всегда знает, где заканчивается предложение! Беда в том, что `TeX` обычно предполагает, что точка служит признаком конца предложения.

Разумеется, как вы могли убедиться из предыдущего абзаца, `TeX` знает, что знаки ? и ! обозначают конец предложения, равно как ему известно это и относительно точек, за которыми следуют либо скобки, либо кавычки. Более того, он достаточно смышлен, чтобы разобраться в том, что точки после прописных букв означают; по-видимому, инициалы и следовательно, не служат концом предложения. Так что пробелы в имени "I. M. Stable" в распечатке `paper.tex` вполне удовлетворительные. Но `TeX`'у ничего не известно относительно сокращений, так что он может не догадаться, что встретившаяся ему в исходном файле аббревиатура типа `Dr. □` не означает конца предложения. Поэтому-то и получились такие большие пробелы в "Dr. Treemunch" и "The Amer. Jour. Recr. Drugs".

`AMS-TeX` имеет управляющий символ '\.' для представления точки, обозначающей сокращение, а не конец предложения. Так что для получения правильных пробелов следует набрать `Dr\ . Treemunch` и `The Amer\ . Jour\ . Recr\ . Drugs`. Заметим, что пробелы после '\.' не игнорируются, поскольку они следуют за управляющим символом.

Кроме неудачных пробелов после аббревиатур, нас также не может удовлетворить в `paper.tex` пара переходов со строки на строку, поскольку там нарушается непрерывность чтения. Далеко не идеально иметь слово "pages" в конце одной строки, а "22-23" в начале другой, разрыв между "¶¶" и "1 and 2" даже еще менее желателен. `TeX` не в состоянии решать такие психологические вопросы, так что вы должны здесь ему помочь. И снова имеется простой выход из затруднения. Если мы наберем тильду ~ между двумя словами, то `TeX` вставит пробел, но не допустит переноса на другую строку. Таким образом, вы можете набрать `pages~22--23` и `\P\P~1 and~2`. Во втором примере вторая тильда была использована для того, чтобы предотвратить появление "2" в начале строки.

Если вы оставили пробел с любой стороны от ~, `AMS-TeX` это просто проигнорирует. Это облегчает задачу вставки тильды в уже набранном тексте: если вы набрали `pages□` и потом вдруг поняли, что нужна связка, можете просто нажать ~, не возвращаясь назад, чтобы стереть □. (Если у вас нет клавиши ~, можете вместо нее набрать управляющую последовательность `\tie`; разумеется, в такой ситуации пробел после `\tie` игнорируется `TeX`'ом во всех случаях.)

¹Такого требования нет в типографских правилах набора русскоязычных текстов.
— Прим. перев.

К сожалению, по-видимому, нет хорошего свода правил, определяющих, когда именно нужно использовать тильды. Это дело здравого смысла и опыта, где именно Т_ЭХ-нический работник проявит при наборе свои склонности. По прошествии некоторого времени вам уже не придется вставлять тильды со второй попытки, поскольку привычка становится второй натурой. Некоторые предложения по использованию тильды вы найдете в третьей части, и тильды будут появляться в ответах к упражнениям там, где они нужны.

После аббревиатур связки часто предпочтительнее `_`. Например, лучше всего набирать `Dr. _Tree-munch`, поскольку пробел будет таким же, как между словами и, кроме того, будет предотвращен перенос на другую строку. Название журнала лучше всего набирать так:

`The _ Amer _ Jour _ Recr _ Drugs _`

• **Упражнение 5.1:**

Вставьте в подходящих местах `~` и `_` в файле `paper.tex` и снова прогоните через Т_ЭХ. Это будет уже шестая попытка. Расслабьтесь и насладитесь творением рук своих.

• **Упражнение 5.2:**

Когда автор впервые попытался выполнить упр. 5.1, он получил странную комбинацию: ‘`¶¶¶¶`’. Какую ошибку он допустил?

• **Упражнение 5.3:**

Как бы вы набрали следующее?

Weird fruit (mangos, papayas, etc.) are avoided by farmers, fastidious eaters, et al.

⚡ В нашей последней версии `paper.tex` все еще остается попытка нежелательного переноса на ‘en-dash’ между “22” и “23”. Т_ЭХ обычно рассматривает дефисы и тире как подходящие места для переноса на другую строку, но `AMS-TЭХ` обеспечивает некий простой способ обозначить те тире и дефисы, где перенос недопустим. Если вы набрали `⓪` — это означает дефис с запрещением переноса. То же относится и к `⓪--` и `⓪---`: перенос на другую строку здесь запрещен. (Вместе с тем, в большинство стилей заложено, что переносы после тире и дефисов в библиографии запрещены, так что там обычно можно обойтись без `⓪` — and `⓪--`.)

Если вы попытаетесь в файле `paper.tex` заменить `--` на `⓪--`, то обнаружите, что Т_ЭХ’у пришлось несколько перетасовать этот абзац, чтобы учесть все ваши спецификации.

Глава 6. С каждым зна́ком знако́м

Теперь вы способны напечатать в Т_ЕX'e 95% из обычно встречающихся текстов. Но достижение бо́льших успехов в Т_ЕXническом наборе требует бо́льших познаний: ведь вы владеете далеко не всеми возможностями Т_ЕX'a, поскольку все еще не умеете правильно набирать . . . , ну, скажем, то, что встретилось в этом абзаце.

Прежде всего, как уже было упомянуто в гл. 1, в текстах иногда встречаются некоторые специальные символы:

<i>Набор</i>	<i>Результат</i>
<code>\dag</code>	† (могильный крестик или обелиск)
<code>\ddag</code>	‡ (двойной могильный крестик)
<code>\S</code>	§ (знак параграфа)
<code>\P</code>	¶ (знак абзаца, или вежа)

Вы можете захотеть использовать и `\copyright` — знак © для защиты вашей уникальной продукции от копирования.

Еще важнее помнить, что символы

`\ { } $ & # % @ - ~ _ "`

используются в А_МS-Т_ЕX'e особым образом, так что нужны некоторые ухищрения, чтобы получить их в распечатке. Мы уже знаем, что знак \$ получается посредством `\$`. Точно так же получаются и остальные стандартные нематематические символы:

<i>Набор</i>	<i>Результат</i>
<code>\{</code>	{
<code>\}</code>	}
<code>\\$</code>	\$
<code>\&</code>	&
<code>\#</code>	#
<code>\%</code>	%
<code>\@</code>	@

Т_ЕX также позволяет набирать `_`, чтобы получить на выходе `_`, например для выражений типа *“first.letter”*; математики редко используют такие конструкции, но программисты — довольно часто. (Если у вас отсутствует клавиша `_`, можете использовать `\underline` вместо `_`.)

• Упражнение 6.1:

Правило, что Т_ЕX игнорирует все, что стоит в строке после знака %, справедливо не на все 100%. Что является исключением? Указание: см. упр. 6.1.

• Упражнение 6.2:

Как вы наберете в ТрХ'e следующее предложение:

My #1 solace is M & M's, though any candy is dandy; I agree
100% with O. Nash† that liquor is quicker, but a fifth of J & B
@ \$13.95 {price as of this writing} is beyond my means.

[Перевод: Для меня M & M идет под номером 1, хотя любая сладость — в радость; я на 100% согласен с O. Nash†, что ликер горячит больше, но бутылка J & B по цене @ \$13.95 {как в добрые старые времена} выше моих возможностей.]

В списке специальных символов знак прямой двойной кавычки " отсутствует по той простой причине, что на самом деле это вовсе не символ: в типографских шрифтах такого символа нет, и вы всегда будете набирать либо ‘, либо ’’. Еще один символ, который не появляется в обычном тексте, — это \ (бэкслэш). Но \ — весьма распространенный математический символ; фактически он используется в двух различных смыслах и в некоторых шрифтах имеет даже разные начертания. В математических формулах для этих символов имеются специальные управляющие последовательности.

Осталось упомянуть лишь о символах ~ и ^ . Они встречаются в обычном тексте только в виде акцентов ~ и ^ над буквами. На самом деле для иностранных языков, базирующихся на латинском алфавите, требуется целый вагон специальных символов и акцентов. Если вы ярый апологет англоязычной литературы и не можете даже подумать о том, чтобы изучать все эти правила расстановки акцентов, то будете просто давать неряшливую аппроксимацию для неанглийских слов, оставляя редактору журнала возможность позаботиться обо всех деталях; когда же ваши иностранные коллеги закипят от злости, просто скажите им: “¡Mañana писать могу — и то дело!”¹

Каждый акцент над буквой выражается управляющей последовательностью с аргументом. В приведенной ниже таблице в качестве аргумента постоянно используется буква o.

Набор	Результат
\‘o	ò (гравис)
\’o	ó (акут)
\^o	ô (циркумфлекс, или “крышка”)
\"o	ö (умлаут)
\~o	õ (тильда)
\u o	ő (краткость)
\v o	ö (гачек)
\H o	ő (венгерский умлаут)
\B o	ō (черта “над”)
\b o	o (черта “под”)
\D o	ò (точка)
\d o	o (точка “под”)
\c o	ç (цедилла)

¹Mañana — завтра, исп. — Прим. перев.

В AMS-TeX 'е также употребляется дужка над буквами, обозначаемая $\backslash t$, которая представляет собой управляющую последовательность с двумя аргументами.

$$\backslash t \text{ oo}$$

$$\text{ôo}$$

(Мы уже упоминали, что можно использовать $\backslash lq$, если на вашей клавиатуре отсутствует клавиша ' . Но нельзя вместо $\backslash \text{'}$ использовать $\backslash \backslash lq$, поскольку TeX будет интерпретировать это как управляющий символ \backslash , за которым следуют буквы lq ! Иные способы набора $\backslash \text{'}$, $\backslash \text{'}$, $\backslash \text{'}$ и $\backslash \text{'}$ описаны в приложении E).

Заметим, что после $\backslash u$, \dots , $\backslash c$ и $\backslash t$ нужны пробелы, поскольку это управляющие слова. Таким образом, слово “háček” следует набирать так:

$$h\backslash a\backslash v \text{ cek}$$

По общему мнению, слово с пробелом внутри кажется странным, так что набор в виде

$$h\backslash a\backslash v\{c\}ek$$

может показаться более наглядным. Но это несколько удлинняет набор, так что вы, возможно, предпочтете первый вариант.

Разумеется, пробелы после управляющих символов, наподобие $\backslash \text{'}$, не нужны. Вы можете даже посчитать, что пробел после $\backslash \text{'}$ как в

$$h\backslash \text{' } a\backslash v \text{ cek}$$

будет ошибкой согласно гл. 1, где говорится, что пробелы после управляющих символов не игнорируются. Однако это правило относится только к управляющим символам типа $\backslash \$$, $\backslash \square$ и $\backslash /$, которые не имеют аргументов. Когда управляющий символ имеет аргумент, действует более общее правило, приоритет которого выше:

TeX всегда игнорирует пробелы, когда ищет аргумент управляющей последовательности.

Следует запомнить это правило: оно вам пригодится в будущем, но для расстановки акцентов над буквами можете просто придерживаться правила типа $\backslash \text{'}$ a , которое во всяком случае упрощает набор.

• Упражнение 6.3:

Какие ошибки допущены в наборе следующего текста?

The word ‘‘ $h\backslash a\backslash v c \text{ ek}$ ’’ should be spelled with a $h\backslash \text{' } a\backslash v \text{ c ek}$.

• Упражнение 6.4:

Как бы вы набрали следующее: *belovèd protégé; rôle coördinator; soufflés, crêpes, pâtés, etc.*

Есть только одна деталь при расстановке акцентов, которую `TeX` не выполняет автоматически: чтобы получить акцент над буквами ‘i’ и ‘j’, нужно сначала взять эти буквы без точек: ‘ı’ и ‘j̄’, что достигается посредством `\i` и `\j`.

• **Упражнение 6.5:**

Как набрать слово `païve` и не проявить наивность?

• **Упражнение 6.6:**

Как бы вы набрали такие имена: Ernesto Cesàro, Pál Erdős, Sergeï Yurév, Eduard Čech, Tâbit ibn Qorra, Muḥammad ibn Mûsâ al-Khwârizmî?

Кроме расстановки акцентов над буквами, `TeX` также умеет распознавать несколько специальных букв:

<i>Набор</i>	<i>Результат</i>
<code>\oe, \OE</code>	œ, Œ
<code>\ae, \AE</code>	æ, Æ
<code>\aa, \AA</code>	å, Å
<code>\o, \O</code>	ø, Ø
<code>\l, \L</code>	ł, Ł
<code>\ss</code>	ß

• **Упражнение 6.7:**

Объясните, как набрать следующее: `Æsop’s Œuvres en français`.

• **Упражнение 6.8:**

Как бы вы набрали имена `Øystein Ore`, `Anders Jonas Ångström`, `Stanisław Świerczkowski`?

Наконец, знаки `j̄` и `¿` из испанских текстов не представляются управляющими последовательностями, а задаются лигатурами:

<i>Набор</i>	<i>Результат</i>
<code>!‘</code>	¡
<code>?‘</code>	¿

(¡Такой способ кажется весьма удачным, поскольку трудно представить себе ситуацию, при которой в тексте нужно было бы напечатать ‘¡’ или ‘¿!’)

• **Упражнение 6.9:**

Как набрать следующее:

`¿What did you say, Señor? I said, “¡Mañana is good enough for me!”`

Во всех примерах, приведенных до сих пор, использовался шрифт прямого светлого начертания, но все сказанное относится и к иным начертаниям. «В каждом начертании имеются свои собственные акценты и специальные литеры, так что `\bf"o` дает ö, а `\sl ^a` дает á.

• **Упражнение 6.10:**

Объясните, как набрать следующее предложение:

*Commentarii Academiæ Scientiarum Imperialis Petropolitanae is
now Akademiã Nauk RF, Doklady.*

Так ... , теперь, когда мы, наконец, благополучно миновали акценты, не следует пропускать обработку многоточия, указывающего на то, что что-то пропущено. Если мы просто наберем в строке три точки, то получим "...", где точки расположены слишком тесно. С другой стороны, если оставлять пробелы между точками, будет слишком просторно и Т_ЭХ может даже разорвать их между двумя строчками. Так что для этого случая у А_МС-Т_ЭХ'а есть специальная управляющая последовательность `\dots`. Ввод

`Гм\dots` как вы вставляете пробелы между точками?

дает текст "Гм ... как вы вставляете пробелы между точками?" Если вы оставите пробел перед `\dots`, то А_МС-Т_ЭХ просто проигнорирует его и вставит такое пространство, которое он считает правильным. Разумеется, пробелы после `\dots` к делу не относятся, поскольку они идут после управляющего слова (но необходим по крайней мере один пробел в нашем предложении между `\dots` и как). Не следует волноваться, если за точками следует знак препинания, Т_ЭХ обработает это как надо ... ! Да ... , волноваться надо самую малость.

• **Упражнение 6.11:**

Какая ошибка в наборе

Да `\dots` , волноваться надо самую малость.

• **Упражнение 6.12:**

Как правильно набрать первый абзац этой главы?

Поскольку в этой главе были затронуты довольно разнообразные вопросы, будет полезно краткое *résumé*:

- (1) Для печати литер {, }, \$, &, #, % и @ имеются очевидные управляющие символы, состоящие из \ и следующей за ним соответствующей литеры. Символы †, ‡, §, ¶ и © представляются управляющим словом, соответствующим названию данного символа. Для воспроизведения _ следует набирать _.
- (2) Акценты над буквами получаются применением различных управляющих последовательностей к той или иной букве (или к паре букв в случае \t). После управляющих символов пробелы необязательны (но допустимы). Для получения i и j используются \i и \j.
- (3) Имеются управляющие последовательности для специальных букв вроде œ, æ, å, ø, ð и ß, но испанские ÿ и ¡ получаются посредством лигатур.
- (4) Для '...' используется `\dots`.

Завершим эту главу перечислением нескольких забытых моментов, которые могут вам понадобиться отныне и в дальнейшем.

- (1) Как мы уже упоминали, TeX предполагает, что точки после прописных букв означают инициалы и, следовательно, не указывают на конец предложения. Но иногда точка или иной знак пунктуации после прописной буквы действительно означает конец предложения:

Возможно, вы не перестаете удивляться, кто это сочиняет такие глупые примеры про TeX. (Вы не найдете ответа в приложении В.) Не смею утаивать — это я, Спивак М. Д. Поддержанный NSF? Нет, кто же на себя возьмет такое.

В этом случае AMS-TeX предлагает ‘.’ для точки, означающей конец предложения. Аналогичные конструкции ‘,’; ‘,’; ‘,’ и ‘,’? используются для других знаков препинания:

... про \TeX. ... приложения~В.)
... я, Спивак М.~Д. ... NSF? Нет, ...

- (2) С другой стороны, правило относительно точек после прописных букв неприменимо, когда точка следует за прописной буквой с акцентом как в “E. Cartan”. Так что здесь для указания обычного межсловного пробела нам потребуется \. (или лучше ~): \’E.~Cartan.
- (3) Еще одна проблема с пробелами возникает, когда мы используем кавычки внутри кавычек:

“Они называют это ‘набором’” — ухмыльнулся он.

Не следует набирать

... ‘набором’{’’} --- ухмыльнулся он.

поскольку в таком случае останется слишком мало места после единичной кавычки — вы получили бы ””, что выглядит как три равностоящие друг от друга кавычки, тогда как ’’’ даст слишком большое расстояние. Особая структура ‘’” решит эту проблему. Следует набирать

... ‘набором’ ‘’ ” --- ухмыльнулся он.

Пробелы перед и после ‘’” игнорируются, так что нет смысла ужимать ваш ввод. Во всех иных законных комбинациях вы также получите правильные пробелы.

’’ ‘’ ”	”
‘ ‘ ‘’ ‘	“
‘ ‘’ ‘	“

Часть 2 ДЕЖУРНЫЕ БЛЮДА

которые по вкусу всем



Глава 7. Математический экстаз

Подлинную радость и восхищение при Трехническом наборе можно испытать, набирая ужасающе громоздкие формулы, с которыми математики сталкиваются повседневно. Посвященные знают, разумеется, что сложность этих формул обманчива. Большие формулы состоят из меньших формул, те в свою очередь — из еще меньших и К счастью, в отличие от Свифтовской иерархии кусающих друг друга блох, этот процесс не может продолжаться *ad infinitum*: все формулы можно свести за несколько шагов до относительно простых. Так что искусство Трехнического набора базируется на простых формулах.

Чтобы дать понять Трех'у, что следующий фрагмент абзаца есть часть математической формулы, его надо просто заключить в знаки $\$$; тогда Трех войдет в "математический режим" и особым образом обработает входной текст. Например, вы можете получить на выходе формулу $z < y = 3(x - 1.5)/(-2|x| + 1)$, набрав

... формулу $\$z < y = 3(x - 1. 5) / (- 2 | x$
 $|+1) \$$, набрав ...

Этот простой пример иллюстрирует несколько важных моментов:

- (1) Буквы в формулах автоматически набираются курсивом, тогда как цифры и знаки препинания (в том числе и скобки) остаются прямыми. Буква l (строчная буква L) превратится в l : ее даже легче отличить от цифры 1, чем обычную 'l'. Заметим, однако, что даже единичный символ является "формулой" — если вы наберете в исходном тексте $\$x\$$, то получите на выходе x .

- (2) Дефис превращается в знак минуса, а слэш — в наклонную черту дроби. Символ $|x|$ используется для “абсолютного значения x ”, но совсем не обязательно знать, что это значит. Важно знать только, что `TeX` выдаст `|`, даже если на клавиатуре вместо `|` находится `!`. Запомните также, что вместо `|` можно использовать `\vert`, если на вашей клавиатуре такой знак отсутствует.
- (3) Самое существенное — это то, что внутри знаков `$ TeX` полностью игнорирует все пробелы и `(carriage-return)` (возвраты каретки).

Имея дело с формулами, `TeX` следует своим собственным правилам расстановки пробелов, в которых, вероятно, учтено куда больше подробностей, чем вы соблюдали бы сами. Например, в формуле $z < y = 3(x - 1.5)/(-2|x| + 1)$ знаками `<` и `=` представлены “бинарные отношения”, которые выполняют функцию глаголов, знаком `+` и первым знаком `-` представлены “бинарные операторы”, выполняющие функцию связки, а последний знак `-` играет роль прилагательного. Стандартные полиграфические правила используют пробелы, которые отражают эти разные роли: т. е. вокруг знаков `<` и `=` ставятся несколько большие пробелы, чем вокруг знаков `+` и первого `-`, тогда как после второго знака `-` пробел вовсе не ставится. К счастью, вам не надо помнить обо всем этом, поскольку за вас это сделает `TeX`.

Несмотря на то, что `TeX` игнорирует все пробелы внутри знаков `$`, их не следует рассматривать как бесполезные, поскольку они помогают облегчить чтение входного текста. Например, отдельные части громоздких формул могут быть отделены друг от друга пробелами. Первоначально, следуя устоявшейся привычке, вы можете использовать пробелы даже в коротких формулах вроде

$$z < y = 3(x - 1.5)/(-2|x| + 1)$$

как бы приблизительно расставляя пробелы, которые должны появиться при распечатке. Однако вы скоро убедитесь, что эти пробелы не играют никакой роли, но замедляют процесс набора, и вы, вероятно, дорастете до чего-то вроде `z<y=3(x-1.5)/(-2|x|+1)`, сохранив пробелы для более сложных случаев. В этом учебнике мы будем использовать пробелы произвольным образом, только лишь чтобы подчеркнуть, что пробелы необязательны.

• Упражнение 7.1:

Что не так в следующем наборе:

$$\text{Если формула } y=x-1 \text{ справедлива ...}$$

• Упражнение 7.2:

Объясните, как набрать следующее предложение:

Удаление одного элемента из n -ки приводит к $(n - 1)$ -ке.

• Упражнение 7.3:

Как правильно набрать следующее:

Рассмотрим график $f - g + h$ в плоскости $x - y$.

Вы должны использовать преимущества того факта, что \TeX достаточно сообразителен, чтобы правильно выбирать пробелы в математических формулах, но из этого не следует, что \TeX может помочь вам вывести математические формулы. \TeX на самом деле не “понимает” формул (во всяком случае понимает их не более, чем наборщик). \TeX нический работник должен рассматривать это как благоприятное обстоятельство, поскольку это означает, что ему и вовсе не следует вникать в формулы. Например, во время набора нет необходимости понимать, почему именно так расставлены скобки в формуле $z < y = 3(x - 1.5)/(-2|x| + 1)$. Математики обычно используют скобки самыми разнообразными способами, но знать эти способы нет никакой необходимости. Если вы наберете $\$y=f(x)\$,$ то получите выражение $y = f(x)$, в котором скобки используются в особом математическом смысле. Точно так же, набрав $\$y=f[x]\$$ и $\$z=f(x,y)\$,$ вы получите формулы $y = f[x]$ и $z = f(x,y)$.

Математики в своих рукописях обычно очень заботятся о расстановке нужных скобок, но иногда им лень их пересчитать в формулах наподобие следующей:

$$1 + 2(3 + 4(5 + 6(7 + x)))$$

Даже если скобки несбалансированны, катастрофы не будет — \TeX на этот предмет не очень разборчив и с большим удовольствием напечатает вам

$$1 + 2(3 + 4(5 + 6(7 + x)))$$

хотя такая формула математически некорректна. И это весьма благоприятное обстоятельство, потому что \TeX не станет протестовать и против формулы $(x, x + 1)$, которая, хоть и кажется математически некорректной, на самом деле вполне допустима, поскольку здесь круглая и квадратная скобки имеют особый смысл.

Фигурные скобки, так же как и круглые и квадратные, часто используются в особом смысле, как в следующей формуле:

$$x + |x| + (x) + [x] + \{x\}$$

Чтобы получить такую формулу следует лишь помнить, что печать фигурных скобок должна специфицироваться управляющими последовательностями $\{$ и $\}$, поскольку обычные фигурные скобки выполняют особую функцию группировки элементов. Таким образом, следует набрать

$$\$x + |x| + (x) + [x] + \{x\}\$$$

Кроме $\backslash\text{vert}$, как замены $|$, в \TeX 'е имеются также $\backslash\text{lbrack}$ и $\backslash\text{rbrack}$, как заменители для $[$ и $]$; последние работают в математическом режиме, равно как и в текстовом. (В \TeX 'е имеется даже $\backslash\text{lbrace}$ и $\backslash\text{rbrace}$ вместо $\{$ и $\}$.)

Квадратные и фигурные скобки также могут выступать в роли круглых скобок, чтобы более явно выделить пары, как, например,

$$1 + 2\{3 + 4[5 + 6(7 + x)]\}$$

Такая конструкция выглядит слегка устаревшей, поскольку $[]$ и $\{ \}$ нередко наделяются особым смыслом, но в ряде разделов математики они встречаются достаточно часто.

• **Упражнение 7.4:**

Как была набрана приведенная выше формула?

• **Упражнение 7.5:**

Что получится при печати такого исходного текста?

$$\$1+2\{3+4[5+6(7+x)]\}\$$$

Указание: Вглядитесь внимательней.

• **Упражнение 7.6:**

А что скажете об этом?

$$\$x \{+\} y\$$$

(Ответ несколько неожиданный — убедитесь, попробовав это вывести, и справиться с приложением В).

◊ При наборе формулы $\$10x + 100y + 1,000z\$$ вас подстерегает неприятный сюрприз. Вы получите $10x + 100y + 1,000z$, поскольку \TeX обычно оставляет небольшой пробел после запятой в математическом режиме.

• **Упражнение 7.7:**

Как по вашему мнению можно преодолеть это?

До сих пор мы имели дело только с формулами, включающими буквы, цифры, знаки пунктуации, связанные символами $=$, $+$, $-$, $<$, $>$ и $|$. Все они имеются на большинстве клавиатур.*

Но математики обычно используют десятки других символов, вроде ∞ , \in , \leq и т. д. Для каждого из этих символов имеется управляющая последовательность с соответствующим именем. Символ ∞ , которым обычно обозначается “бесконечность” (более или менее), именуется `\infty`. Более короткое название `\infty` введено для других случаев, о которых вы узнаете в одной из следующих глав. Символ \in встречается в формулах типа $x \in A$. Чтобы получить формулу $x \in A$, вам следует просто набрать `\$x \in A\$`. Заметим, что пробел после `\in` нужен, несмотря на то, что \TeX “игнорирует” пробелы в формулах.

На некоторых клавиатурах могут иметься особые клавиши типа ∞ , но вам не следует ими пользоваться, разве что ваш \TeX специально приспособлен к такой клавиатуре.

◊ \TeX имеет управляющую последовательность `\ell` для символа ℓ , который математики используют иногда в особом смысле. Но прежде чем использовать `\ell`, следует убедиться, что мы хотим получить именно это, поскольку многие авторы часто в своих рукописях пишут ℓ вместо самого обыкновенного l . (Символ, похожий на ℓ , встречается на многих машинках, и старательная машинистка часто заменяет им обычное l , для того чтобы достичь большего отличия от машинописной l , чем добавляет еще больше путаницы!)

Мы должны также упомянуть клавишу ``! В обычном тексте она дает звездочку `*`, которая, как правило, используется для сносок. Поскольку \LaTeX - \TeX вместе с тем стилем, который вы используете, задает вид знака сноски и автоматически вставляет его, вам будет трудно получить этот знак в тексте. Но `*` иногда бывает нужна в математических формулах. Например, запись `\$(f*g)(x)\$` дает математическую формулу $(f * g)(x)$.

Названия всех математических символов перечислены в приложении F. Большинство из них можно изучать по мере надобности, однако несколько общепотребительных символов следует узнать прямо сейчас:

<code>\leq</code> или <code>\le</code>	\leq (“меньше или равно”)
<code>\geq</code> или <code>\ge</code>	\geq (“больше или равно”)
<code>\neq</code> или <code>\ne</code>	\neq (“не равно”)
<code>\notin</code>	\notin (“не принадлежит”)

Кроме того, управляющая последовательность `\not` дает /, что можно использовать для “отрицания” какого-нибудь бинарного оператора (кроме $=$). Например, управляющая последовательность `\equiv` дает оператор \equiv в математическом режиме, а `\not\equiv` дает $\not\equiv$; последовательности `\ne` и `\neq` фактически суть аббревиатуры для `\not=`. Заметим, однако, что / в \notin имеет другой наклон, чем / в \neq . Поэтому нам следует всегда набирать `\notin` вместо `\not\in`, который дает менее симпатичный символ \notin .

Кроме спецзнаков математики иногда используют акценты, такие, как в выражении \hat{A} , и разнообразные начертания букв, как, например, в формуле $ax + by$. Различные шрифты и акценты в математическом режиме и в обычном тексте достигаются совершенно разными способами; детальное рассмотрение этого отложено до части 3 по причине того, что этот предмет довольно специфичен.

С другой стороны, математики также дополняют свой арсенал специально созданных символов обыкновенными греческими буквами, и делают это очень часто. Например, `\alpha`, `\beta`, `\gamma` и `\delta` дают 4 первые греческие буквы α , β , γ и δ , тогда как `\Gamma` дает прописную Γ , а `\Delta` соответственно прописную Δ . (Не существует последовательностей `\Alpha` и `\Beta`, поскольку соответствующие прописные буквы ничем не отличаются от аналогичных латинских “A” и “B”).

Пятая буква греческого алфавита ϵ получается посредством `\epsilon`. Заметим, что это совершенно не похоже на символ математической принадлежности \in . К сожалению, не все математики различают эти символы в своих рукописях, так что небольшая консультация перед началом работы не помешает. Убедитесь также, что автор различает греческую букву `\nu` (ν) от v и букву `\kappa` (κ) от x или k . Греческая буква `\phi` должна также отличаться от символа \emptyset , который получается в результате набора `\emptyset`.

Чтобы еще больше усугубить положение, для буквы ϵ имеется другое написание: `\varepsilon`, которое получается посредством `\varepsilon`, и еще четыре греческие буквы имеют альтернативы:¹ ввод `(\theta, \pi, \rho, \phi)` дает $(\theta, \pi, \rho, \phi)$; тогда как ввод `(\vartheta, \varpi, \varrho, \varphi)` приводит к $(\vartheta, \varpi, \varrho, \varphi)$. Все прописные греческие буквы также имеют альтернативное начертание: например, `\varGamma` дает Γ и `\varDelta` дает Δ .

¹Относительно судьбы альтернативного написания пятой буквы — κ (`\varkappa`) см. приложение G.

Если у вас нет преимуществ гуманитарного образования (вы не изучали классиков в подлиннике и не были членом студенческого братства²), то вы можете и не знать названий греческих букв. Это не беда — все они перечислены в приложении F со всеми другими специальными знаками. Кроме того, вы могли подумать, что довольно мучительно каждый раз набирать α для всех этих α , особенно если они часто встречаются в статье. Не лучше ли было бы давать какие-нибудь коротенькие имена вроде α для α и β для β ? И, коль скоро мы коснулись этой темы: почему в TeX'e такие забавные имена для других забавных символов? Называть именем \equiv знак \equiv удобно математикам, которые обычно произносят \equiv как 'эквивалентно', но это мало помогает техническим исполнителям — наборщикам!

По части терминологии TeX придерживается весьма консервативных взглядов: он обычно пользуется длинными описательными именами для большинства управляющих последовательностей, поскольку в этом случае у него есть некоторое основание для общего соглашения относительно этих имен — любая система аббревиатур обычно имеет как ярых сторонников, так и ярых противников. TeX может позволить себе вести себя столь непринужденно, потому что наборщики TeX'a всегда могут сами определить свои собственные управляющие последовательности. Когда экзотический символ всего один-два раза встречается в статье, вы можете просто найти его в приложении F и использовать предлагаемое там имя. Но для часто повторяющегося символа вам следует ввести свое собственное имя — вы можете сказать TeX'у, что хотите использовать λ вместо α , и т. д. Процесс определения ваших собственных управляющих последовательностей не будет освещен вплоть до части 3, поскольку всегда легче изучать некую установившуюся систему, как бы она вас порой ни раздражала, чем создавать свой собственный особый вариант. Но вы можете всегда, когда выполняете то или иное упражнение, иметь ввиду такую возможность, если вам приходится иметь дело с более длинными управляющими последовательностями, чем хотелось бы.

• Упражнение 7.8:

Вы набрали длинную статью, в которой автор в многочисленных формулах использовал букву x . И вот O сообщает вам, что E курсивное x на самом деле не что иное как κ ! Так что вам придется при помощи вашего текстового редактора заменить все x на κ . Как бы вы поступили?

• Упражнение 7.9:

Теперь предположим, что у нас обратная задача. Взбалмошный автор просит поменять все κ на k . Какую предосторожность следует предпринять?

• Упражнение 7.10:

Заглянув в приложение F для выяснения названий различных символов, объясните, как набрать следующие формулы:

²В оригинале "fraternity or sorority" — элитарные студенческие клубы, в названия которых входят буквы греческого алфавита. — Прим. ред.

- (1) $X \setminus (A \cup B) = (X \setminus A) \cap (X \setminus B)$
 (2) $x \notin A \not\subset B$
 (3) $(X \times Y) \times Z \simeq X \times (Y \times Z)$
 (4) $\omega \wedge (\eta \wedge \lambda) = (\omega \wedge \eta) \wedge \lambda$
 (5) $V \oplus \Lambda(V)$
 (6) $\|a(x + y)\| \leq |a| \cdot (\|x\| + \|y\|)$
 (7) $2 \cdot \aleph = \aleph$
 (8) $2 \cdot \omega \neq \omega$
 (9) $\nabla R(X, Y)$
 (10) $(100 \pm .001) \div 5$
 (11) $\forall x \gg A$
 (12) $f * g: A \rightarrow B$
 (13) $x \mapsto \alpha + x$
 (14) $f(x) \in o(x) \ \& \ g(x) \in O(x) \implies f \circ g(x) \in o(x)$

В этой главе мы ознакомились только с основами набора математических формул — так что мы даже не в состоянии получить такое простенькое и весьма интересное выражение как $8 = 2^3$. Но мы уже знаем достаточно для того, чтобы начать сажать новые и интересные ошибки! Чтобы увидеть, как TeX успешно справляется с нашими новыми возможностями, создадим следующий файл (здесь мы временно используем `\par` вместо пустой строки, чтобы чуть позднее прояснить некоторые вещи):

```
This paragraph contains a formula $x that we forgot to
end properly. It also contains a second formula y$ that
we forgot to start, and a final
formula $x+y that we also forgot to end.\par This
paragraph contains another formula \alpha+\beta$
that we forgot to start.
```

[Перевод:

```
В этом абзаце содержится формула $x, которую мы забыли
правильно закончить. В нем также содержится другая формула y$,
которую мы забыли начать, и, наконец,
формула $x+y, которую мы также забыли закончить.\par Этот
абзац содержит еще одну формулу \alpha+\beta$,
которую мы забыли начать.]
```

TeX не умеет читать наши мысли, когда мы делаем ошибки такого рода. Коль скоро ошибка сделана, первые две строки мы специфицировали как формульный поток

xthatweforgottoendproperly.Italsocontainsasecondformulay

Ко всему прочему, здесь нет ничего такого, что не могло бы войти в формулу!

С другой стороны, начинать новый абзац внутри математической формулы совершенно бессмысленно, поэтому такая возможность запрещена. Так что когда TeX видит

`formula $x+y that we also forgot to end.\par This`

он знает, что формула *должна* заканчиваться прежде `\par`, означающего конец абзаца. В самом деле, когда мы прогоним нашу статью через TeX, то получим следующее сообщение об ошибке:

`! Missing $ inserted.`

. . .

`1.8 formula $x+y that we also forgot to end.\par
This`

со знаком ? на следующей строке, как обычно. Мы пропустили четыре строки настоящего сообщения, которые выглядят несколько более устрашающе, чем то, что мы видели раньше, поскольку основная стратегия при работе с такого рода сообщениями состоит в концентрации внимания на верхней и двух нижних строках и в игнорировании почти всей информации между ними.

Как уже было сказано, в двух нижних строках нам сообщается, что TeX обнаружил ошибку, когда наткнулся в строке 8 на `\par`. В верхней же строке сообщается

`! Missing $ inserted.`

т. е., что TeX пытался поправить ситуацию, вставив \$ непосредственно перед `\par` (разумеется, на самом деле TeX не вставил \$ в ваш файл, а только лишь в свою собственную копию). Если вы просто нажмете (carriage-return) в ответ на ?, TeX исправит положение как сумеет (TeX не в состоянии определить, что \$ на самом деле должен быть помещен после +, и потом уже слишком поздно). Затем мы получим сообщение `Overfull box` вроде

`Overfull \hbox ...`

`[]\tenrm This para-graph con-tains a for-mula $\teni xthatwe`

появившееся потому, что формула

xthatweforgottoendproperly.Italsocontainsasecondformulay

не помещается в строку!

Далее возникнет более интересное сообщение об ошибке:

```
! Missing $ inserted.
```

```
. . .
```

```
1.9 paragraph contains another formula \alpha
                                         +\beta$
```

Как явствует из этого сообщения, Т_ЭХ вставляет знак \$, который нужен в начале формулы $\alpha + \beta$. Так что если вы нажмете (carriage-return), Т_ЭХ не только до конца обработает файл, но и выдаст в этом случае правильную формулу.

Т_ЭХ оказался в состоянии определить, что знак \$ был необходим перед `\alpha`, потому что не ищет лавров на ниве греческого языка — так что когда дело касается греческих букв он рассматривает их как специальные математические символы и считает, что они допустимы только в математических выражениях. При необходимости вы могли бы получить выражение $\tau\epsilon\chi$ в тексте, набрав `$_\tau\epsilon\psi_\chi$`, но если вам нужен настоящий греческий текст, следует обратиться к иной версии Т_ЭХ'а, созданной для клавиатуры с греческими буквами, и вы, вероятно, даже не будете читать этот учебник, который, вне всякого сомнения, относится только к английским текстам.

• Упражнение 7.11:

Что происходит, когда Т_ЭХ обрабатывает следующий файл:

```
This paragraph has a formula $x that we forgot to end.
```

```
This paragraph has the formula 3\alpha+\beta
```

```
This is accurate within \pm.0003 percent.
```

```
This paragraph has only text.
```

• Упражнение 7.12:

¿Señor, что происходит, когда у вас вот такой файл?

```
The symbols < and > and | give
$<$ and $>$ and $|$ in math mode.
```

Глава 8. Скверный разрыв? Сделайте “выключку”

TeX использует специальный способ расстановки переносов, позволяющий делить абзацы на строки равной длины. Но эта процедура осложняется в случае текста, перемежающегося формулами. TeX изо всех сил старается не рвать строку в середине формулы, но если такой разрыв неизбежен, то TeX разорвет формулы только после бинарных соотношений (неохотно) или после бинарных операторов (как последнее средство).

Так, если TeX должен разорвать строку где-то посередине формулы $f(x, y) = (x + y)(x - y)$, он постарается сделать это после знака $=$. Но если такой вариант не проходит, TeX решится разорвать, например, так: $f(x, y) = (x + y)(x - y)$, после знака $-$, или после знака $+$ (и то, и другое смотрится одинаково плохо). К счастью, разрыв после бинарных операций встречается довольно редко, но если вы все-таки его сделали, а автор этого не одобрил, можете заставить TeX разорвать формулу после знака $=$ при помощи `\mathbreak`. AMS-TeX имеет также `\nomathbreak` для предотвращения разрыва в формуле, так что можете попытаться набрать `\nomathbreak` после $+$ и $-$, но тогда у вас блестящие перспективы получить сообщение **Overfull box**. И, конечно, получить его еще легче, когда именно в конце строки текста встретила формула вроде $abc(def)ghi[jkl]mnp$, которую вообще нельзя рвать.

В большинстве случаев лучшим выходом из плохого разрыва или **Overfull box**, которые бывают в математических формулах, будет обращение к автору с просьбой слегка изменить текст. В самом деле, большинство математиков с радостью согласятся добавить или удалить несколько строк из своей не столь уж бессмертной прозы ради сохранения драгоценных формул; с появлением TeX'a возникли возможности менять текст, которым прежние типографы не обладали.

◆ Обычные правила, используемые TeX'ом для разрыва формул в тексте, позволяют рвать их только после бинарных отношений, а ни в коем случае не до них¹. TeX может попытаться разорвать формулу после знака равенства: $f(x, y) = (x + y)(x - y)$, но и в этом случае получит **Overfull box**, поскольку первая часть формулы “ $f(x, y) =$ ” не помещается в строку. В таком случае вместо того, чтобы переписывать, автор может позволить перенести знак $=$ на другую строку, хотя большинство полиграфистов посчитало бы это дурным тоном. Вы можете обязать сделать разрыв, набрав $f(x, y)\mathmathbreak =$. Имеется и другая возможность: `\allowmathbreak=`, которая просто позволяет разрывать до знака $=$, не делая это обязательным (в этом случае при изменении абзаца в будущем вам не придется беспокоиться о заданном обязательном разрыве, который теперь может оказаться неподходящим.)

Хотя TeX может рвать формулу $f(x, y) = (x + y)(x - y)$ после знака $=$ или даже после знаков $+$ или $-$, он не станет разрывать ее после запятой. Запятые в формулах обычно отделяют части единого математического выражения, поэтому мало кто из математиков способен примириться с таким разрывом: $f(x, y) = (x + y)(x - y)$. TeX также использует особый пробел после запятой: более

¹Напоминаем, что речь идет о наборе англоязычных текстов. — Прим. перев.

внимательно приглядываясь к формуле $f(x, y) = (x + y)(x - y)$, действительно можно заметить, что пробел после запятой в формуле меньше пробела после запятой в тексте.

По этой причине истинные знаки пунктуации должны всегда находиться и вне знаков $\$$. Например, если вы хотите получить

We have $a < b$, $a = b$ or $a > b$ in this case.

то следует набирать

We have $a < b$, $a = b$ or $a > b$ in this case.

Если бы вы набрали

We have $a < b$, $a = b$ or $a > b$ in this case.

то получили бы

We have $a < b$, $a = b$ or $a > b$ in this case.

В этом случае пробел смотрится значительно хуже, и разрыв строки после “ $a < b$,” становится невозможным.

• **Упражнение 8.1:**

Как бы вы набрали следующее?

There exist such division algebras only for $n = 1, 2, 4$ or 8 .

• **Упражнение 8.2:**

А как бы вы набрали

We have $f(x) = A, B$ or C for $x = 0, 1, \dots, n$.

• **Упражнение 8.3:**

Объясните, как как нужно набирать такой текст:

For all a and b we have $a < b$, $a = b$, or $a > b$. We say that $<$ is a *partial ordering*.

• **Упражнение 8.4:**

Как бы вы набрали следующее?

If $a, b, c > 0$, then $f(a, b, c) > 0$.

Упражнение 8.2 иллюстрировало использование `\dots` между формулами, однако математики часто вставляют точки прямо в формулы. Например, набор `\mathbf{f(1, \dots, n)}` дает формулу $f(1, \dots, n)$; заметим, что пробелы после запятых здесь меньше. А если вы наберете `\mathbf{1 + \dots + n}`, произойдет нечто удивительное: вы получите формулу $1 + \dots + n$, с точками, расположенными посередине строки. (Более подробно относительно поведения `\dots` в математическом режиме можно прочесть в части 3).

• Упражнение 8.5:

Как бы вы набрали следующее?

We have shown that $f(1, \dots, n) \leq f(0, \dots, 0) + f(1, \dots, 1) + \dots + f(n, \dots, n)$ for $n \geq 1$.

Порой даже предельно пунктуально придерживаясь правил пунктуации можно не получить удовлетворительных результатов из-за авторских ляпсусов. Насаждая хороший математический стиль, математики часто забывают употребить хоть слово, чтобы отделить друг от друга формулы, принадлежащие разным частям предложения. Вместо того чтобы сказать

If $x > 0$, then $y > 1$.

математику ничего не стоит выдать

(*) If $x > 0$, $y > 1$.

Это предложение было набрано следующим образом:

If $x>0$, $y>1$.

с запятой вне формул, но выражение (*) все еще слегка смущает, если читать его быстро, поскольку две формулы здесь имеют тенденцию сливаться в одну. Чтобы несколько улучшить положение, сохранив при этом авторскую индивидуальность, Трехнический наборщик использует $_$, чтобы получить дополнительный пробел между двумя формулами. Таким образом, вы могли бы набрать

If $x>0$, $_$ $y>1$. or If $x>0$, $_ _$ $y>1$.

и получить

If $x > 0$, $y > 1$.

(Это один из тех случаев, упоминаемых в ответе к упр. 5.3, когда вы бы действительно не хотели, чтобы $_$ проигнорировал пробел перед $_$, потому что вы используете комбинацию $_ _$ или $_ _$, для получения дополнительного пробела.)

Если в тексте появилась длинная формула, то она почти наверняка разорвется плохо. Поэтому длинные формулы часто делают “выключными” — помещают на отдельной строке вот так:

$$1 + 1 = 2$$

На самом деле даже короткие формулы часто делают выключными, чтобы подчеркнуть их важность. Искусство правильно выделить формулы представляет одну из существенных составляющих математического стиля, так что Трех никогда сам не примет решения о том, выключать или нет ту или иную формулу — вам следует подсказать Треху сделать это, вставив в исходный текст знаки $_ _$ вместо $_$, что означает перейти к “выключной формуле в математическом режиме”. Например, ввод

... If $_ _ f(x)=x+1$, then we will have $_ _ f([x+1] _ / [x+2])=\{ [x+1] / [x+2] \} + 1$
 $_ _ = (2x+3) / (x+2) . _ _$

Consequently, ...

даст на выходе

... If $f(x) = x + 1$, then we will have

$$f([x + 1]/[x + 2]) = \{[x + 1]/[x + 2]\} + 1 = (2x + 3)/(x + 2).$$

Consequently, ...

Заметим, что все, что заключено между знаками **\$\$** воспринимается как однострочная формула — разговор о многострочных формулах отложим до гл. 15. Заметим, также, что хотя запятая была вынесена за знак **\$**, точка была набрана между знаками **\$\$** — в противном случае точка появилась бы в начале следующей строки, непосредственно перед “Consequently”!

Как и в случае обычного математического режима, внутри знаков **\$\$** пробелы и (carriage-return) игнорируются, так что выключную формулу

$$\mathbf{\$f(x)=(x+y)(x-y)\$}$$

можно набрать и так:

$$\mathbf{\$f(x)=(x+y)(x-y)\$}$$

(Мне больше нравится набирать выключные формулы именно так, потому что их легче выявить, когда потом просматриваешь файл, но это дело вкуса, разумеется.)

• Упражнение 8.6:

Как бы вы набрали следующее?

После невообразимых усилий, предпринятых в предыдущей главе, мы, наконец, успешно доказали фундаментальный результат

$$1 + 1 = 2,$$

а теперь попробуем доказать, что

$$2 + 2 = 4.$$

В качестве первого шага в этом направлении докажем дистрибутивный закон умножения.

• Упражнение 8.7:

Как бы вы получили такую формулу?

$$1 + 2 + 3 + 4 + 5 + 6 + 7 + 8 + 9 + 10 + 11 + 12 \\ + 13 + 14 + 15 + 16 + 17 + 18 + 19 + 20 = 190$$

А как такую?

$$a + b = c \\ A + B = C$$

◊ В некоторых журналах выключные формулы не центрируют, а вместо этого располагают их так:

$$1 + 1 = 2$$

и

$$1 + 1 + 1 = 3$$

с фиксированным отступом от левого края. По-видимому, это делается лишь для того, чтобы избавить наборщика от рутинной работы по центрированию формул, однако \TeX также умеет размещать выключные формулы таким образом, когда это указано в стиле.

Глава 9. 2^й уровень сложности

Трехнический набор не был бы такой уж большой проблемой, если бы все сводилось только к пробелам и разрывам строк. Но в математических формулах уйма информации передается взаимным расположением их частей. Даже если вы не математик, то вероятно знаете, что a^2 и a^3 — принятая сокращенная форма записи для $a \times a$ (“ a в квадрате”) и $a \times a \times a$ (“ a в кубе”). И существует много иных ситуаций, когда математики используют верхние индексы ^{сверху} и нижние _{снизу}. Вы не можете указать то или иное расположение непосредственно на терминале, поскольку ввод производится построчно, так что вся эта информация должна быть передана другим способом.

Чтобы Трех понял, что вы хотите поместить литеру в верхний индекс, нужно просто набрать перед ней значок `^`:

Набор	Результат
<code>\$x^2\$</code>	x^2
<code>\$x^a\$</code>	x^a
<code>\$x^\alpha\$</code>	x^α
<code>\$2^x\$</code>	2^x

На многих клавиатурах имеются клавиши “стрелка вверх” `↑`. Их иногда используют просто для перемещений, но если вам повезет, то можете встретить и такую клавишу, которая на самом деле выдаст на экран `↑`; тогда эта клавиша будет, вероятно, заменять клавишу `^`. Бывают случаи (хотя лучше проконсультироваться у вашего местного эксперта), когда символ `↑` можно использовать вместо `^`, что даже симпатичнее, поскольку $x\uparrow^2$ смотрится лучше, чем x^2 . Разумеется, вы можете оказаться настолько невезучим, что у вас не окажется ни `^`, ни `↑`. В таком случае придется иметь дело с управляющей последовательностью `\sp` для получения верхних индексов:

<code>\$x\sp2\$</code>	x^2
<code>\$x\sp a\$</code>	x^a
<code>\$x\sp\alpha\$</code>	x^α
<code>\$2\sp x\$</code>	2^x

Заметьте, что следует позаботиться о пробеле после `\sp`, если следующая литера — буква.

То, что на большинстве клавиатур нет ничего, что бы также уверенно указывало “вниз”, как `^` или `↑` указывает “вверх”, не может не удручать. Самый похожий символ — “подчеркивание” `_`. Так что Трех использует `_` для получения нижнего индекса:

<code>\$x_2\$</code>	x_2
<code>\$x_y\$</code>	x_y

Если вам повезло, и у вас есть клавиатура с клавишей, дающей на экране \downarrow , можете употреблять ее вместо $_$. Если же у вас нет ни $_$, ни \downarrow , воспользуйтесь управляющей последовательностью $\backslash sb$:

$$\begin{array}{ll} \$x\sb2\$ & x_2 \\ \$x\sb y\$ & x_y \end{array}$$

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

$$\begin{array}{ll} \$x^2y^2\$ & x^2y^2 \\ \$x \^ 2y \^ 2\$ & x^2y^2 \\ \$x_2y_2\$ & x_2y_2 \\ \$\{}_2F_3\$ & {}_2F_3 \end{array}$$

Обратите внимание на использование пустой группы $\{\}$ в последнем примере для получения “предшествующего” нижнего индекса. Для этого нижний индекс применяется к пустой формуле. Вы могли бы также набрать просто $\{}_2F_3\$$, потому что \TeX считает, что если формула начинается с $\^$ или $_$, то в начале нее существует некая пустая группа. Но всегда лучше и для \TeX 'а, и для самого себя явно выражать свои намерения, добавляя пустую группу.

• **Упражнение 9.1:**

Попробуйте набрать две формулы $\$x + \{}_2F_3\$$ и $\$x + _2F_3\$$ и посмотрите, есть ли между ними какие-нибудь различия.

Если вы хотите поместить в верхний или нижний индекс целое выражение, его достаточно заключить в фигурные скобки:

$$\begin{array}{ll} \$z=x^{\{2y\}}\$ & z = x^{2y} \\ \$2^{\{32\}}\$ & 2^{32} \\ \$x_{\{10\}}\$ & x_{10} \\ \$x^{\{\{3y\}\}}\$ & x^{\{3y\}} \end{array}$$

В приведенных выше примерах верхний индекс 32 и нижний индекс $_{10}$ должны быть помещены в фигурные скобки, потому что они представляют собой два символа, хотя для читателя каждый из них — всего лишь одно число. С другой стороны, $\$x^{\alpha}\$$ на предыдущей странице не требует заключения α в фигурные скобки, потому что α — всего лишь один символ, несмотря на то, что в управляющей последовательности для его наименования использовано несколько литер.

• **Упражнение 9.2:**

Объясните, как набрать следующее предложение:

If the $n-1$ numbers $x_1, \dots, x_{\alpha-1}, x_{\alpha+1}, \dots, x_n$ are all $\neq x_\alpha \pm 1$, then $f(x_1, \dots, x_n) > 0$.

Поскольку верхние и нижние индексы бывают нужны только в математическом режиме, \TeX категорически запрещает использовать их вне математического режима, чтобы легче было обнаруживать пропущенные знаки $\$$.

• **Упражнение 9.3:**

Создайте файл из следующих трех абзацев:

This paragraph has the formula x^2 .

This paragraph has the formula x^{10}
and the formula x^{10} .

This paragraph has only text.

и посмотрите, какое сообщение об ошибке вы получите при прогоне его через Т_ЭX, нажимая при необходимости клавишу (carriage-return).

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

$(x+1)^3$	$(x+1)^3$
$(x^2)^3$	$(x^2)^3$
$[x^2]^3$	$[x^2]^3$
$\{x^2\}^3$	$\{x^2\}^3$

Наверное, математики и опытные операторы порой недоумевают — откуда Т_ЭX узнает, например, что 3 служит верхним индексом всего выражения $(x+1)$ или (x^2) ? Ответ весьма прост: Т_ЭX этого не знает — он просто пунктуально следует инструкции и ставит 3 в качестве верхнего индекса правой скобки! Если же вы заключите формулу в фигурные скобки, показатель степени будет относиться ко всему выражению:

$\{(x^2)\}^3$	$(x^2)^3$
$\{[x^2]\}^3$	$[x^2]^3$
$\{\{x^2\}\}^3$	$\{x^2\}^3$
$\{((x^2)^2)\}^4$	$((x^2)^2)^4$

С точки зрения логики это кажется более правильным, но обозначение $(x^2)^3$ и так вполне понятно, так что нет оснований усложнять себе жизнь. В самом деле, это обозначение применялось столетиями, и более “логичное” обозначение может только озадачить математиков — так что оставим $(x^2)^3$ для случаев, когда необходимо особое выделение.

• **Упражнение 9.4:**

Объясните, как набрать следующее:

В некоммутативной группе имеем:

$$(ab)^{-1} = b^{-1}a^{-1}$$

и

$$(ab)^{-2} = [(ab)^{-1}]^2 = [b^{-1}a^{-1}]^2 = b^{-1}a^{-1}b^{-1}a^{-1},$$

но

$$(a^m)^2 = a^{m+m} = a^{2m}.$$

Более интересная проблема возникает, когда нужно получить выражение вроде

$$a^{b^c}.$$

Для математика это совсем иное выражение, чем $(a^b)^c$. В формуле a^{b^c} буква a имеет верхний индекс, но это не b , а целая формула b^c :

$$a^{\boxed{b^c}}$$

TeX обращается с такой формулой точно так же, как математик: поскольку вся формула b^c служит верхним индексом, то следует заключить выражение b^c в фигурные скобки:

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

$$a^{b^c}$$

Освоив этот принцип, вы без труда сможете выдавать самые разнообразные формулы, при виде которых обычные наборщики стали бы рвать на себе волосы:

$$\text{\$}a^{\{b^{\{c+1\}}\}}\text{\$}$$

$$a^{b^{c+1}}$$

$$\text{\$}2^{\{(2^x)\}}\text{\$}$$

$$2^{(2^x)}$$

$$\text{\$}2^{\{2^{\{2^{\{2^x\}}}\}}\text{\$}$$

$$2^{2^{2^{2^x}}}$$

$$\text{\$}2^{\{(a+b)^2\}}\text{\$}$$

$$2^{(a+b)^2}$$

$$\text{\$}x_{\{y_2\}}\text{\$}$$

$$x_{y_2}$$

$$\text{\$}x_{\{y^2\}}\text{\$}$$

$$x_{y^2}$$

Обратите внимание, что в формуле a^{b^c} буква a набрана шрифтом нормально-го размера, которым набран основной текст. Этот размер называется “t-size”, верхний индекс b меньшего размера, “s-size”, а c еще меньшего размера, “ss-size”. Более мелких размеров TeX не производит, так как столь мелкие буквы было бы очень трудно прочесть.

• Упражнение 9.5:

Наберите следующую выключную формулу:

$$2^{x_1+\dots+x_{n+1}} = 2^{x_{n+1}} \cdot (2^{x_1+\dots+x_n})$$

• Упражнение 9.6:

Объясните, как набрать следующее:

$$\text{Suppose that } x^{x^x} = (x^x)^x. \text{ Prove that } x = 1 \text{ or } 2.$$

• Упражнение 9.7:

Опишите отличия между результатом набора $\text{\$}\{x^y\}^z\text{\$}$ и $\text{\$}x^{\{y^z\}}\text{\$}$.

• Упражнение 9.8:

В некоторых компьютерных системах используются конструкции типа a^b^c для обозначения a^{b^c} , но TeX такого ввода не допускает. Посмотрите, что произойдет, если вы попытаетесь предложить TeX’у файл с конструкцией $\text{\$}a^b^c\text{\$}$.

Хотя выражения типа A^a_b или A_b^a Т_ЕX не приемлет, допускаются формулы с верхним и нижним индексами, причем их можно указывать в любом порядке.

A^a_b	A^a_b
A_b^a	A^a_b
$x^{31415}_{92} + \pi$	$x^{31415}_{92} + \pi$
$\Gamma_{y^a_b}^{z^c_d}$	$\Gamma_{y^a_b}^{z^c_d}$

К формулам типа A^a_b часто прибегают наборщики типографий, потому что на линотипе трудно получить A^a_b . Для Т_ЕX'а это, очевидно, не проблема, однако многие математики в некоторых ситуациях все еще предпочитают располагать верхние и нижние индексы на разных расстояниях от буквы. Например, формула типа x_i^2 по всей видимости представляет собой сокращенную запись для $x_i \times x_i$, где роль i совершенно отличается от роли 2. В этих случаях многие математики предпочитают x_i^2 . Можно вынудить Т_ЕX отодвинуть индекс, воздвигнув на его пути искусственные баррикады из пустых групп $\{\}$:

$x_{i\{\}^2}$ или $\{x_{i\{\}^2}$	x_i^2
$R_{i\{\}\{jk\}_1}$	R_i^{jk}

Второй из этих примеров отнюдь не плод большого воображения автора; это, скорее, плод большого воображения “тензорного анализа” — области математики, шутиливо определяемой как исследование нижних и верхних индексов, где точное местоположение играет важную роль. Если автор использует такие обозначения, надо точно следовать Е указаниям, но сначала не вредно проверить. Вы можете также захотеть узнать, чему автор отдает предпочтение при наборе x_i^2 . Если нет полной уверенности, то запомните, что легче заменить $x_{i\{\}^2$ и $x_{i\{\}^3$ на $x_{i\{\}^2}$ и $x_{i\{\}^3}$ — хороший текстовый редактор запросто заменит в нужных местах \wedge на $\{\}$ — но не так легко заменить x^2_i и x^3_i на $x_{i\{\}^2}$ и $x_{i\{\}^3}$. По этой причине я всегда стараюсь набирать сначала нижний индекс, а потом верхний.

• Упражнение 9.9:

Объясните, как набрать следующее:

Пусть не существует такого λ , что $x_i = \lambda y_i$, $i = 1, 2$. Тогда уравнение $(\lambda y_1 - x_1)(\lambda y_2 - x_2) = 0$, т.е., уравнение

$$\lambda^2(y_1^2 + y_2^2) - 2\lambda(x_1 y_1 + x_2 y_2) + (x_1^2 + y_1^2) = 0$$

не имеет решения λ .

Как вы могли заметить из упр. 9.2, 9.4 и 9.5, дополнительные пробелы вокруг бинарных операторов типа $+$ и $-$ исчезают, когда они находятся в верхних или нижних индексах. Более того, бинарные операторы могут быть верхними или нижними индексами и сами по себе. Например, $*$ представляет собой бинарный оператор в конструкциях типа

$$(f * g)(x)$$

но можно также набрать и

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

Более того, можно даже получить что-то вроде

$$\begin{array}{ll} \text{\$f_+} & f_+ \\ \text{\$f_-} & f_- \end{array}$$

хотя последний набор выглядит лучше, если использовать дополнительные фигурные скобки.

$$\begin{array}{ll} \text{\$f_{+}} & f_+ \\ \text{\$f_{-}} & f_- \end{array}$$

Вам, по-видимому, не захочется заботиться о наборе необязательных фигурных скобок, но бывает иная ситуация, в которой настраиваемый редактор может кое-что значительно упростить. Вы можете сделать так, чтобы одним нажатием клавиши получать $\overset{\sim}{\{}$ или $\underset{\sim}{\{}$, и затем вернуть курсор на одну литеру, так что вы готовы набирать верхние и нижние индексы и затем перескочить через правую фигурную скобку. Это требует одного дополнительного нажатия, когда вам не нужны фигурные скобки, но экономит массу усилий, когда они вам нужны. У такого способа есть еще одно преимущество: начав левой фигурной скобкой длинный индекс, вы можете не волноваться о том, что вы забудете его закрыть фигурной скобкой (очень распространенная ошибка).

Кроме верхних и нижних индексов математики часто используют обозначение f' (“ f штрих”). Т_ЭX обладает управляющей последовательностью `\prime`, но если вы наберете `f\prime`, то получите fl , совсем не то, что хотели! Штрихи следует употреблять, как и другие верхние индексы:

$$\text{\$f^{\prime}} \quad f'$$

Вашей первой реакцией будет: Почему Т_ЭX так сложно манипулирует со штрихами? Почему не иметь в качестве `\prime` меньший знак штриха, который уже занимает верхнее положение? Ответ крайне прост: штрихи часто используются вместе с нижними индексами, как в

$$\text{\$f_2^{\prime}} \quad f_2'$$

Если бы штрих представлял собой обычный символ, то он оказался бы правее f_2 , вместо того, чтобы быть непосредственно над 2. Несколько успокоившись, вы еще думаете, что все же довольно обременительно каждый раз набирать `^{\prime}`, когда нужен $'$, а если немного забежать вперед, то нетрудно представить, сколько препятствий вас ждет на пути получения f''' — придется набирать `f^{\prime\prime\prime}`! К счастью, Т_ЭX позволяет обойти это неудобство. Когда Т_ЭX находится в математическом режиме, он транслирует $'$ в `^{\prime}`; более того, `''` транслируется в `^{\prime\prime}` и `'''` — в `^{\prime\prime\prime}`, и т.д.

$$\begin{array}{ll} \text{\$f'[g(x)]g'(x)} & f'[g(x)]g'(x) \\ \text{\$y_1'y_2''+y_3''' } & y_1' + y_2'' + y_3''' \end{array}$$

Это отличный пример того, как легко пользоваться штрихами в Т_ЭX'e.

• **Упражнение 9.10:**

Объясните, что неправильно в таком наборе:

В формуле '\$\{x+y\}\$' знак '\$+\$' есть бинарный оператор.

У $\mathcal{AMS-TeX}$ 'а нет специального способа для получения штрихов в нижнем индексе, поскольку они довольно редко бывают нужны. Чтобы получить конструкцию типа $F'_i(w, z)$, следует просто набрать '\$F_{\prime}(w, z)\$'. Иногда бывает нужен \prime в случаях вроде этого:

$$g^{\prime\prime} \qquad g'^2$$

Но здесь также можно воспользоваться трюком с пустой группой:

$$g^{\{\}^2} \qquad g'^2$$

• **Упражнение 9.11:**

Почему нельзя было просто набрать '\$g'^2\$'?

• **Упражнение 9.12:**

Объясните, как набрать следующее выключное выражение:

$$f^{(n+2)} = [f^{(n+1)}]' = [f^{(n)}]'' = [f^{(n)}]'''$$

Прежде чем мы завершим эту тему, неплохо бы узнать об одной западне, которая, возможно, подстерегает нас при наборе нижних и верхних индексов: иногда бывает, что нужно ставить фигурные скобки в случаях, когда они, казалось бы, совсем не обязательны. Предположим, например, вам нужен символ A_{\neq} . Вам кажется, что было бы правильным набрать

A_{\neq}

поскольку набор A_{α} вполне корректен и приводит к A_{α} . Но в случае A_{\neq} вы получите нечто странное: ' A_{\neq} '! Фокус в том, что \neq — вовсе не один обособленный символ, а просто аббревиатура для $\not=$. Так что \mathcal{TeX} , получив ввод A_{\neq} , оттранслирует его в $A_{\not=}$, после чего старательно поставит / в качестве нижнего индекса буквы A ! Если вы являетесь счастливым обладателем специализированного редактора, автоматически заключающего в фигурные скобки нижние и верхние индексы, то вам нечего беспокоиться о таких аномалиях. Но даже если это не так, тоже нет повода для беспокойства. Такие ситуации крайне редки и достаточно быть осведомленными о них, чтобы не теряться при их появлении.

• **Упражнение 9.13:**

Что происходит, когда вы стараетесь \mathcal{TeX} изировать файл, содержащий следующее:

Этот файл состоит только из формулы A_{\notin} .

Глава 10. Наших проблем громадье

В формулах с верхними и нижними индексами символы все же следуют “слева направо”. Но составные части многих формул дислоцируются более замысловато, зачастую громоздясь одна на другую. У \LaTeX ’а имеется целый ряд управляющих последовательностей, позволяющих справляться с этими проблемами.

Самая важная из таких управляющих последовательностей — `\frac`, которая порождает

$$(*) \quad \frac{n+1}{n+3}$$

и аналогичные дроби. `\frac` — это управляющая последовательность с двумя аргументами: числителем над чертой дроби и знаменателем под ней. Дробь (*) была получена посредством

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

Эта выключная формула представляет собой \TeX ’овскую формулу размера “d-size”, с числителем и знаменателем обычного размера t-size. Если мы наберем `$$\frac {n+1}{n+3}$$`, то получим $\frac{n+1}{n+3}$. Теперь вся дробь имеет размер t-size, из чего следует, что числитель и знаменатель имеют меньший размер, т. е. s-size.

Заметим, что (*) можно набрать и так:

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

В этом случае пробел после `\frac` не нужен, поскольку за ним стоит небуква `{`; а пробел перед `{n+3}` игнорируется не только потому, что мы находимся в математическом режиме, но и потому, что \TeX всегда игнорирует пробелы, когда ищет аргументы управляющей последовательности.

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

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

так что вам нет необходимости об этом беспокоиться. Заметим также, что дроби типа

$$\frac{1}{n+1} \quad \text{и} \quad \frac{N-1}{2}$$

встречаются довольно часто, так что вам не раз представится возможность опустить фигурные скобки.

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

Так как в выключных формулах числитель и знаменатель имеют размер t-size, то, если в дроби в свою очередь содержится дробь, вы получите такие результаты:

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

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

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

Но некоторые авторы предпочитают $\frac{x}{2}$, а другие настаивают на

$$\frac{x}{1 + \frac{x}{2}} \quad \text{и} \quad \frac{\frac{x}{2} + 1}{2}$$

с $\frac{x}{2}$ размера d-size. У \LaTeX ’а есть управляющая последовательность `\ds`, вынуждающая набирать формулу в размере d-size. Например, чтобы в тексте появилось $\frac{x}{2}$, нужно набрать

`\ds\frac{x}{2}`

Для получения же $\frac{x}{2} + \frac{x}{2}$ достаточно набрать

`\ds\frac{x}{2} + \frac{x}{2}`

— заметьте, что `\ds` вызывает переключение на d-size для *всей формулы*; в этом смысле `\ds` аналогичен `\rm`, `\it`, `\bf` и `\sl`, за исключением того, что его действия ограничены той формулой, в которой он появляется.

• Упражнение 10.1:

Как были набраны выключные формулы

$$\frac{x}{1 + \frac{x}{2}}$$

и

$$\frac{\frac{x}{2} + 1}{2}$$

• **Упражнение 10.2:**

Наберите следующее:

Hölder's Inequality: Let $0 \leq p, q \leq \infty$ with $\frac{1}{p} + \frac{1}{q} = 1$. If $f \in L^p(\mu)$ and $g \in L^q(\mu)$, then $fg \in L^1(\mu)$ and

$$\|fg\|_1 \leq \|f\|_p \|g\|_q.$$

Из ответов к упр. 10.1 и 10.2 должно стать понятно, почему `\dsize` работает так же, как `\rm`, `\it`, `\bf` и `\sl` в тексте, а не как управляющая последовательность с одним аргументом: когда вы хотите заставить дробь стать размера d-size, обычно достаточно просто вставить перед ней `\dsize` — дополнительные фигурные скобки могут понадобиться в редких случаях. Но 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, что избавляет вас от всяких забот о фигурных скобках. Управляющая последовательность `\dfrac` автоматически даст размер d-size; таким образом, набор `\dfrac` аб эквивалентен набору `{\dsize\frac ab}`. Вы найдете `\dfrac` особенно удобным, когда, набрав сначала `\frac`, потом вдруг решите, взглянув на распечатку, что на самом деле был нужен размер `\dsize` — необходимо будет всего лишь набрать дополнительно `d` перед `f`. Так что можете оставить `\dsize` для тех редких случаев, когда d-size потребуется для иных конструкций.

• **Упражнение 10.3:**

Снова выполните упражнение 10.1, используя `\dfrac`.

Как вы могли догадаться, $\mathcal{A}\mathcal{M}\mathcal{S}\text{-}\mathcal{T}\mathcal{E}\mathcal{X}$ также имеет управляющую последовательность `\tsize` для получения формул размера t-size.

• **Упражнение 10.4:**

Объясните, как набрать следующую выключную формулу:

$$fg = \frac{1}{2}[(f+g)^2 - f^2 - g^2]$$

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 размера t-size для `\frac`, совершенно при этом не беспокоясь о фигурных скобках.

• **Упражнение 10.5:**

Снова выполните 10.4, не пользуясь `\tsize`.

Когда дробь появляется в верхнем индексе, размер которого s-size, это значит, что числитель и знаменатель должны быть еще мельче, а именно, ss-size:

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

В верхних индексах почти всегда более предпочтительно представление “через косую”, но опять же, некоторые авторы могут оказаться весьма разборчивыми, и форма “через прямую” для некоторых ситуаций будет стандартной. (Обычные типографские наборщики имеют массу хлопот с дробями размера s-size, но

TeX не выражает недовольства на этот счет — только бедные читатели будут выражать недовольство по поводу слишком мелких символов.)

AMS-TeX оснащен также `\ssize` и `\sssize` для превращения размера формулы в *s-size* или *ss-size*, но вряд ли вам это когда-нибудь понадобится. Естественно, `\dsize`, `\tsize`, `\ssize` и `\sssize` действуют совершенно одинаково, вынуждая всю формулу изменить размер, т. е. по аналогии с изменением шрифта `\rm`, `\it`, `\sl` и `\bf`; так что вам не надо заботиться о “switch to d-size”, и т. п. Вам и в дальнейшем не следует также беспокоиться о подобных аномалиях — в AMS-TeX'e имеются управляющие последовательности, работающие аналогично.

Кроме дробей, математики часто пользуются особым обозначением

$$\binom{n}{k},$$

так называемым “биномиальным коэффициентом”. Вам не надо знать, что это означает, достаточно помнить, что для получения его используется `\binom`, так что верхняя выключная формула была получена посредством

`$$\binom nk$$`

Если отвлечься от того, что результат выглядит несколько иначе, работа `\binom` несколько не отличается от работы `\frac`, с теми же самыми соглашениями относительно размеров верхней и нижней части:

`$$\binom n{\frac k2}$$` $\binom{n}{\frac{k}{2}}$

`$$\frac{\binom nk}2$$` $\frac{\binom{n}{k}}{2}$

• **Упражнение 10.6:**

Объясните, как представить первый пример в виде

$$\binom{n}{k/2} \quad \text{или} \quad \binom{n}{\frac{1}{2}k}$$

а второй — в виде

$$\frac{\binom{n}{k}}{2}$$

Как вы уже догадываетесь, AMS-TeX имеет также `\dbinom` и `\tbinom` для `\binom` размера *d-size* или *t-size*.

• **Упражнение 10.7:**

Снова выполните последнюю часть упражнения 10.6, не прибегая к `\dsize`.

До сих пор дроби и биномиальные коэффициенты были наиболее общими конструкциями, в которых одна формула взгромождалась на другую. Но возможны и другие случаи. В некоторых ситуациях, например, автор может захотеть получить дробь с более тонкой чертой:

$$\frac{\frac{(u' \circ u^{-1})(v'' \circ u^{-1})}{(u' \circ u^{-1})} - \frac{(v' \circ u^{-1})(u'' \circ u^{-1})}{(u' \circ u^{-1})}}{(u' \circ u^{-1})^2}$$

Для таких дробей \AA M S T E X может посредством `\thickfrac` менять толщину дробной черты. Кроме того, есть еще и “символ Лежандра” $\binom{a}{b}$, который похож на биномиальный коэффициент, за тем исключением, что он имеет дробную черту. Не было также упомянуто и “число Эйлера” $\langle n \rangle$, также напоминающее биномиальный коэффициент, но в “угловых скобках” $\langle \rangle$ вместо круглых. У \AA M S T E X нет специальных управляющих последовательностей для этих и других фантазий математиков, но имеется механизм “порождения дробей”, позволяющий получить любой подобный символ, который вам нужен. Подробности вы найдете в части 3.

Управляющая последовательность `\frac` \AA M S T E X ’а создана на основе управляющей последовательности `\over` T E X ’а, которая также позволяет получать дроби, но пользователи \AA M S T E X ’а могут и не предполагать о ее существовании, так как `\frac` применять легче. Если вы, однако, допустите оплошность с `\frac`, то можете попасть в более сложную ситуацию, поскольку T E X сначала оттранслирует `\frac` на свой язык, прежде чем начнет разбираться, в чем дело.

• Упражнение 10.8:

Составьте файл из трех абзацев:

Этот абзац состоит из $\frac{1}{a+b}$ и $\frac{1}{1+a+b}$.

Этот абзац состоит из $\frac{1}{a+b}$.

Этот абзац состоит только из текста.

и прогоните через T E X . Для двух первых обобщений об ошибке нажимайте клавишу `\carriage-return`, но когда получите сообщение

`Runaway argument?`

с которым раньше не сталкивались, нажмите `h` или `H`, чтобы получить помощь, прежде чем снова нажать `\carriage-return`.

Глава 11. Большие дары TeX'a

Хотя $+$ это “бинарный оператор”, связывающий между собой два выражения в формулах, вроде $a + b$, его можно использовать и для связи нескольких выражений, как, например, в формуле $a + b + c + d$ — сумме a , b , c и d . Но когда математикам нужно указать, что суммируется много чисел a_1, a_2, \dots, a_n , они обычно используют особое выражение $\sum_{i=1}^n a_i$. Хотя нет необходимости точно понимать, что это значит, важно знать о некоторых особенностях такой формулы. Знак \sum представляет собой в принципе греческую прописную букву сигма, но обычно большего размера и несколько отличающегося по начертанию от Σ . Чтобы получить такой символ \sum , вам следует набрать `\sum` (в математическом режиме). Разумеется, порой трудно отличить Σ от \sum в рукописном или машинописном варианте, но догадаться о том, что это знак \sum можно по верхним и нижним подключкам типа $\sum_{i=1}^n$. Эти мелкие формулы называются “пределами” суммирования, и в выключных формулах с \sum для них отводится особое место. Если набрать `$$\sum_{i=1}^n a_i`, то в тексте появится выражение $\sum_{i=1}^n a_i$, но если вы набираете выключную формулу

`$$\sum_{i=1}^n a_i`

то получите

$$\sum_{i=1}^n a_i$$

Заметим, что в этом случае \sum имеет больший размер, а “пределы”, хотя они были набраны как верхний и нижний индексы, таинственным образом сместились над и под знак \sum .

\sum называется “большим оператором”, и некоторые другие бинарные операторы имеют “больших” собратьев. Например, бинарные операторы `\cup` (\cup) и `\cap` (\cap) имеют соответствующие большие операторы `\bigcup` и `\bigcap`. В тексте эти управляющие последовательности дают \cup и \cap , а в выключных формулах они дадут \bigcup и \bigcap , и “пределы”, которые были набраны как верхние и нижние индексы, будут проставлены над и под этим значками. О других больших операторах можно прочитать в приложении F.

• Упражнение 11.1:

Как вы наберете следующие формулы:

$$(1) \quad (f \cdot g)^{(n+1)}(a) = \sum_{k=0}^{n+1} \binom{n+1}{k} f^{(k)}(a) g^{(n+1-k)}(a)$$

$$(2) \quad 0 = (1 + -1)^n = \sum_{j=0}^n (-1)^j \binom{n}{j}$$

$$(3) \quad \sum_{i=1}^n x_i^2 \cdot \sum_{i=1}^n y_i^2 = \sum_{i=1}^n x_i^2 y_i^2 + \sum_{i \neq j} x_i^2 y_j^2$$

$$(4) \quad \sum_{i=1}^p \sum_{j=1}^q \sum_{k=1}^r a_{ij} b_{jk} c_{ki}$$

$$(5) \quad \Sigma^2: [X, S_0(\infty)] \rightarrow [\Sigma^2 X, S_0(\infty)]$$

$$(6) \quad \bigcup_{n=1}^m (A_n \cup B_n)$$

$$(7) \quad X \setminus \bigcup_{i \in I} A_i = \bigcap_{i \in I} X \setminus A_i$$

Иногда у больших операторов бывают многострочные пределы, как, например

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

Для однострочных пределов используется `_` или `\sb`, а для получения этой формулы следует набрать

```
$$\sum \sb 0\le i\le m\ 0<j<n \endSb P(i,j)$$
```

Между `\sb` и `\endSb` каждый `\\` свидетельствует о переходе на новую строку. Точно так же имеются `\sp ... \endSp` для получения многострочных верхних пределов.

• Упражнение 11.2:

Представьте формулу (4) в упр. 11.1 как

$$\sum_{\substack{1 \leq i \leq p \\ 1 \leq j \leq q \\ 1 \leq k \leq r}} a_{ij} b_{jk} c_{ki}$$

• Упражнение 11.3:

Посмотрите, что происходит, когда в вашем файле есть

```
$$\sum \sb 0\le i\le m\ 0<j<n $$
```

```
\enddocument
```

с пропущенным в выключной формуле `\endSb`.

Есть еще один “большой оператор”, действие которого отличается от действия `\sum`. Если вы наберете `\int` в математическом режиме, получите символ \int , который математики называют “знаком интеграла”; в выключной формуле этот знак будет большего размера, точно так же, как и в случае знака `\sum`. Но “пределы” не переместятся над и под интеграл.

$$\begin{array}{ll} \text{\$}\int_a^b\text{\$} & \int_a^b \\ \text{\$\$\int_a^b\$\$} & \int_a^b \end{array}$$

Кроме `\int` в $\mathcal{A}\mathcal{M}\mathcal{S}$ -Т_EX'e также имеется `\oint`, который выдает \oint и \oint . Знаки `\int` часто появляются группами, например `\iiint` и `\iiiiiint`. Но не вздумайте набирать что-нибудь вроде `\int\int`, так как это даст $\int\int$ со слишком большим пробелом между символами. Вместо этого существуют специальные символы

$$\begin{array}{ll} \text{\$\$\iint\$\$} & \iint \\ \text{\$\$\iiint\$\$} & \iiint \\ \text{\$\$\iiiiiint\$\$} & \iiiiiint \\ \text{\$\$\idotsint\$\$} & \int \dots \int \end{array}$$

Вопрос о расстановке “пределов” в `\sum` и `\int` Т_EX на самом деле не решает: это зависит от того, какой стиль вы используете. В некоторых журналах предпочитают ставить пределы в `\sum` в виде верхних и нижних индексов даже в выключных формулах, а в некоторых — пределы интегрирования в выключных формулах ставятся сверху и снизу. 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

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

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 Т_EX проставит все верхние и нижние индексы как “пределы”.

• Упражнение 11.4:

Как набрано соотношение (*)?

• Упражнение 11.5:

Объясните, как следует набирать выключную формулу

$$\iiint_M d\omega = \iint_{\partial M} \omega$$

Хотя Т_EX снабжает нас очень удобным средством выбирать стили для больших операторов в выключных и текстовых формулах, авторы иногда хотят, 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 приплюснутой. Конечно, вы могли бы добиться этого, просто набрав

`\$ \displaystyle \sum_{n=1}^{\infty} \frac{(-1)^n}{n} \$`

Такие конструкции часто используются в книгах, но редакторы журналов недолюбливают их из-за того, что расточается много бумаги, не говоря уже о том неопрятном виде, который принимает страница. Приемлемым компромиссом будет

`\$ \sum \limits_{n=1}^{\infty} \frac{(-1)^n}{n} \$`

который дает $\sum_{n=1}^{\infty} \frac{(-1)^n}{n}$; такие конструкции несколько повышают удобочитаемость, не принося в жертву слишком многого.

◆ `\limits` имеет свою противоположность `\nolimits`, который приводит к тому, что у больших операторов пределы остаются справа, даже если обычно они переставляются. Однако `\limits` и `\nolimits` должны использоваться только в особых случаях. Если вам не нравятся соглашения, задаваемые стилем `amsppt` для пределов в `\sum` и `\int`, вам не следует пользоваться `\limits` и `\nolimits`, чтобы подправлять каждую формулу. Правильнее будет ввести такие изменения на постоянной основе, при которых вы создадите нечто вроде своего стиля. В части 3 объясняется, как это сделать.

Глава 12. Создание своего пространства

Хотя обычно в формулах Т_ЭX задает правильные расстояния между символами, 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”. Этот маленький интервал настолько полезен для подгонки математических формул, что у Т_ЭX’a для него имеется чрезвычайно простая управляющая последовательность $\backslash,$. Таким образом, правильный способ получения приведенной выше формулы такой:

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

Вы также можете захотеть заложить тонкую шпацию в любых формулах математического анализа перед dx или dy или d нечто, таким образом:

$$dx \, dy = r \, dr \, d\theta$$

Но перед выражениями вроде $\frac{dy}{dx}$ или перед dx в dy/dx опция $\,$, не используется.

• Упражнение 12.1:

Как набрать следующие формулы?

$$(1) \quad y \, dx - x \, dy$$

$$(2) \quad x \, dy/dx$$

$$(3) \quad \int_1^x \frac{dt}{t}$$

$$(4) \quad \int dy = \int \frac{dy}{dx} dx$$

• Упражнение 12.2:

Как получить формулу

$$\frac{dz}{dx} = \frac{\frac{dz}{dy}}{\frac{dy}{dx}}$$

в которой дробная черта между $\frac{dz}{dy}$ и $\frac{dy}{dx}$ слегка длиннее?

⚡ А_МS-Т_ЭX позволяет использовать $\,$, также и вне математического режима, но к этому вы будете прибегать редко, разве что для создания специальных эффектов. Один из подобных случаев — это выражение ‘55 mi/hr’ (или 55 км/ч), которое выглядит лучше со шпацией $\,$, после ‘55’ вместо обычного межсловного пробела.

• **Упражнение 12.3:**

Как бы вы набрали следующие формулы?

- (1) 1 ml equals 1.000028 cc
 (2) $g = 9.8 \text{ m/sec}^2$
 (3) $0^\circ \text{ C equals } 32^\circ \text{ F}$

Совершенно иного рода проблема с пробелами возникает в случае, когда имеется выключная формула с дополнительным условием типа

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

Кроме тонкой шпации, полиграфисты имеют дело с большими пробелами, называемыми “квадрат” (“quad space”), и обычно считается, что между основной формулой и дополнительным условием должен быть пробел в 2 квадрата. Т_ЭX имеет управляющую последовательность `\quad` для установки пробела в 1 квадрат и даже особую аббревиатуру `\qqquad`, заменяющую `\quad\quad`, так что приведенную выше формулу можно набрать так:

$$\begin{aligned} & \$\$ \\ & F_n = F_{n-1} + F_{n-2}, \quad \qqquad n > 1. \\ & \$\$ \end{aligned}$$

• **Упражнение 12.4:**

Объясните, как получить следующую выключную формулу?

$$\int_0^\infty \frac{t - ib}{t^2 + b^2} e^{iat} dt = e^{ab} E_1(ab), \quad a, b > 0.$$

Т_ЭX способен делать пробелы любого размера, но в этой главе мы даже не побеспокоились о введении различных мер, с которыми знаком Т_ЭX (в их число входят дюймы, сантиметры, пункты, цитеро и др.), так как управляющие последовательности `\,`, `\quad` и `\qqquad` обычно дают именно те пробелы, которые нам и требуются. Есть несколько иных ситуаций, в которых формулы можно улучшить посредством вставки или удаления тонких шпаций на основе здравого смысла, и А_МS-Т_ЭX имеет множество остроумных приспособлений, позволяющих справиться со всеми иными специфическими проблемами расстановки пробелов. Но такие тонкости мы отложим до части 3.

Глава 13. Длина меняется сама собой

Когда вы набираете `\frac`, черта дроби автоматически простирается до нужной длины, достаточной для того, чтобы охватить и числитель, и знаменатель. У TeX'a много других управляющих последовательностей, которые автоматически выбирают нужный размер символа в зависимости от контекста. Например, вы можете провести черту под (`\underline`) или над (`\overline`) формулой:

$$\begin{array}{ll}
 \underline{4} & \frac{4}{4+x} \\
 \underline{\underline{4+x}} & \frac{4+x}{x^{n+m}} \\
 x^{\underline{n+m}} & \frac{4}{x^3 + x^3} \\
 \overline{\overline{x^3+x^3}} &
 \end{array}$$

Можете снабдить ее различного рода стрелками сверху:

$$\begin{array}{ll}
 \overrightarrow{x+y} & \overrightarrow{x+y} \\
 \overleftarrow{x-y} & \overleftarrow{x-y} \\
 A^{\overleftarrow{\overrightarrow{x+y}}} & A^{\overrightarrow{x+y}}
 \end{array}$$

Можете нарисовать стрелки под формулами посредством `\underrightarrow`, `\underleftarrow` и `\underleftrightharrow`. Самые распространенные стрелки, `\overrightarrow` и `\underrightarrow`, имеют также краткие имена `\overarrow` и `\underarrow`. Разумеется, если бы вы часто использовали такие стрелки, вам бы захотелось дать им еще более короткие имена, как это объясняется в гл. 18.

Стрелки над и под буквами не следует путать с другими стрелками, которые действуют как бинарные отношения. Мы уже проходили такие вещи как `\rightarrow` и `\Longrightarrow`; другие стрелки перечислены в приложении F.

• Упражнение 13.1:

Объясните, как набрать каждое из следующих предложений:

(1)

Сумма векторов \overrightarrow{OA} и \overrightarrow{OB} определяется как

$$\overrightarrow{OA} + \overrightarrow{OB} = \overrightarrow{OP},$$

где $OAPB$ есть параллелограмм.

• **Упражнение 13.6:**

Наберите следующее:

Для $0 < x \leq 1$ имеем

$$f(x) = \frac{1}{\left[\frac{1}{x}\right]}$$

Можно также варьировать и размерами фигурных скобок — не забывайте только, что надо набирать `\{` и `\}`:

$$\left\{\left\{\frac{1-x^2}{1-x^2}\right\}\right\}$$

• **Упражнение 13.7:**

Наберите следующую выключную формулу:

$$\|f\|_p = \left\{ \int_0^1 |f|^p \right\}^{\frac{1}{p}}$$

Круглые, квадратные и фигурные скобки — всего лишь часть “ограничителей”, которые могут быть использованы после `\left` и `\right`. Среди этих ограничителей такие:

вертикальная черта:		or <code>\vert</code>
двойная вертикальная черта:		\ or <code>\Vert</code>
левая “кочерга вниз”:	[\lfloor
правая “кочерга вниз”:]	\rfloor
левая “кочерга вверх”:	[\lceil
правая “кочерга вверх”:]	\rceil

(Более редкие ограничители можно найти в части 3.) Если они используются сами по себе, то управляющие последовательности справа воспроизведут символы обычного размера, показанные слева. Но вместе с `\left` и `\right` они порождают символы разного размера.

• **Упражнение 13.8:**

Как набрать выключную формулу:

$$\left\| \frac{x}{a} \right\| = \frac{\|x\|}{|a|}$$

После `\left` и `\right` можно использовать какие угодно ограничители — это не обязательно должны быть парные скобки. Например, можно набирать

$$\left\langle \left\langle \frac{a}{b}, \frac{c}{d} \right\rangle \right\rangle \quad x \in \left(\frac{a}{b}, \frac{c}{d} \right]$$

или даже (вот дикость!)

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

Когда вы имеете дело с такой непонятной конструкцией, как во втором из этих примеров, `\left` и `\right` не только выбирают соответствующий размер ограничителей; они также помогают установить правильный пробел в формуле. Как правило, \TeX истолковывает `]` как правый ограничитель, так что если бы вы набрали `$\$x \in]a, b[$` , то получили бы $x \in]a, b[$, потому что \TeX не оставляет дополнительные пробелы между бинарным отношением и правым ограничителем. Но `\left]` указывает \TeX 'у на то, что `]` выбран в качестве левого ограничителя, так что если набрать `$\$x \in \left]a, b\right[$` , то получится $x \in]a, b[$; в этом частном случае скобки будут, разумеется, обычного размера, но зато пробелы будут правильными.

• **Упражнение 13.9:**

Математикам, настаивающим на столь диких обозначениях, объясните, как правильно набрать следующую формулу: $]-\infty, T[\times]-\infty, T[$.

Имеются еще два довольно специальных ограничителя:

левая угловая скобка <code><</code>	<code>\langle</code>
правая угловая скобка <code>></code>	<code>\rangle</code>

Для них нет особого повторяющегося расширения, поэтому для них установлен максимальный размер. В шрифте, использованном для этой книги, ваш потолок

⟨ ⟩

Если ваша формула невероятных размеров, \TeX просто установит для нее наибольшую из существующих скобок. Кстати, \TeX позволяет набирать как `\left<` и `\right>`, так и `\left\rangle` и `\right\langle`, несмотря на то, что `<` и `>` не то же самое, что `<` и `>`.

• **Упражнение 13.10:**

Наберите следующее:

Связка ∇ совместна с метрикой тогда и только тогда, когда

$$\frac{d}{dt} \langle V, W \rangle = \left\langle \frac{DV}{dt}, W \right\rangle + \left\langle V, \frac{DW}{dt} \right\rangle.$$

В конструкции `\left... \right` \TeX вычисляет необходимый размер ограничителей, прежде всего взглянув на формулу внутри `\left` и `\right`. Так что команды `\left` и `\right` аналогично фигурным скобкам всегда должны появляться парами.

• **Упражнение 13.11:**

Наберите следующее:

Мы вывели формулу корней квадратного трехчлена, “дополняя до полного квадрата”:

$$ax^2 + bx + c = a \left(x^2 + \frac{b}{a}x + \frac{c}{a} \right) = a \left[\left(x + \frac{b}{2a} \right)^2 + \left(\frac{c}{a} - \frac{b^2}{4a} \right) \right].$$

TeX как бы расставляет фигурные скобки вокруг пар `\left... \right`, так что действие измененного размера `\dsize`, `\tsize`, ... распространяется только внутри пары.

• **Упражнение 13.12:**

Наберите формулу

$$\left(\frac{1}{\sqrt{A}} + \sum_{i=1}^n a_i \right) \left(\frac{1}{\sqrt{B}} + \sum_{i=1}^n b_i \right)$$

сделав как можно меньше ударов по клавишам.

Требование, чтобы `\left` и `\right` всегда выступали в паре, может вызвать легкую панику при наборе формулы вроде

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

с *нечетным* ограничителем. Специально для таких ситуаций TeX позволяет поставить точку после `\left` и после `\right`, что дает “пустые” ограничители `'\left.'` и `'\right.'`. Таким образом, TeX достигает сбалансированности, но в распечатке не выдаст никакого символа. Итак, нужно набрать

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

• **Упражнение 13.13:**

Что произойдет, если вы наберете

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

с запятой, случайно нажатой вместо точки?

• **Упражнение 13.14:**

Что произойдет, если вы полностью проигнорируете `\left.` и просто наберете следующее:

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

• **Упражнение 13.15:**

Как получить такую выключную формулу:

$$\left[\frac{dy}{dx} \right]_{x=a}^2$$

Может возникнуть необходимость в пустом ограничителе и в паре с еще одним ограничителем — последним, о котором мы здесь упомянем. Это ограничитель `/`, максимальный размер которого ограничен, как и у ограничителей `(` и `)`. Его можно получить либо из `\left/`, либо из `\right/`.

$$\left. \frac{c+1}{d} \right/ x^2$$

Для ограничителей `\left/` или `\right/` важно перекрыть самую высокую часть формулы, так как размер ограничителей зависит от того, что именно находится между `\left` и `\right`.

• Упражнение 13.16:

Как следует набирать выключную формулу

$$x^2 \left/ \frac{c+1}{d} \right.$$

Несмотря на удобства, предоставляемые конструкцией `\left... \right`, бывают случаи, когда вам нужно самим указывать точные размеры ограничителей. Например, у TeX'a имеются `\bigl` и `\bigr` для получения ограничителей, размером чуть-чуть больше обычного.

$$\begin{array}{ll} \bigl(x-s(x)\bigr)\bigl(y-s(y)\bigr) & (x-s(x))(y-s(y)) \\ \bigl[x-s[x]\bigr]\bigl[y-s[y]\bigr] & [x-s[x]][y-s[y]] \\ \bigl||x|+|y|\bigr| & ||x|+|y|| \\ \bigr\lfloor\sqrt{A}\bigr\rfloor & \lfloor\sqrt{A}\rfloor \end{array}$$

Здесь `\left` и `\right` не дали бы такого эффекта, потому что они не создают объекты большего размера, чем необходимо, так что ввод

$$\left[x-s[x]\right]\left[y-s[y]\right]$$

привел бы просто к $[x-s[x]][y-s[y]]$.

(Есть еще одно важное отличие `\bigl` и `\bigr` от `\left` и `\right`: хотя конструкции `\bigl` and `\bigr` [приводят к тому, что TeX считает] левым ограничителем, а `[` — правым, `\bigl` и `\bigr` не группируют объекты; вы можете даже поставить в формулу `\bigl`, а парный `\bigr` не вставлять.)

Хотя ограничители `\big` могут делать удивительные вещи в смысле удобочитаемости некоторых формул, с точки зрения издательских редакторов математических книг и журналов они могут показаться слишком большими. Там веками складывалась традиция использовать один и тот же размер круглых скобок для вложенных выражений одного размера. В формуле вроде $(x+f(x))^2$ нет необходимости ставить скобки большего размера для внешней пары, равно как и нет необходимости поднимать еще выше верхний индекс². Так что пользуйтесь ограничителями `\big` аккуратно. Математики предпочитают старый стиль, так что всякого рода “улучшения” вызывают у них чувство дискомфорта.

Бывает также, что `\left... \right` дает слишком большие ограничители, например, когда вы пользуетесь `\left` и `\right` для того, чтобы заключить в скобки `\sum` с пределами:

$$\left(\sum_{k=1}^n A_k \right)^2$$

Смотрится несколько лучше, когда скобки немного не закрывают пределы

$$\left(\sum_{k=1}^n A_k \right)^2$$

так что здесь нужен особый размер. Кроме ограничителей `\bigl` и `\bigr`, которые только слегка больше обычных, Т_ЕX располагает ограничителями `\biggl` и `\biggr`. Их размер именно такой, какой нужен Т_ЕX'у для двустрочной формулы

$$\left\{ \frac{b}{d} \right\}$$

и они как раз подойдут для `\sum`:

$$\biggl(\sum_{k=1}^n A_k \biggr)^2$$

• Упражнение 13.17:

Наберите следующее:

$$(1) \quad \left(\sum_{i=1}^n x_i y_i \right)^2 = \sum_{i=1}^n (x_i y_i)^2 + \sum_{i \neq j} x_i y_i x_j y_j$$

$$(2) \quad \pi(n) = \sum_{m=2}^n \left[\left(\sum_{k=1}^{m-1} \lfloor (m/k) / \lceil m/k \rceil \right)^{-1} \right]$$

$$(3) \quad m^* \left(A \cap \left[\bigcup_{i=1}^n E_i \right] \right) = \sum_{i=1}^n m^* (A \cap E_i)$$

• Упражнение 13.18:

Как вы наберете следующее?

$$(1) \quad \left(\sqrt{\frac{A}{B}} + \sum_{i=1}^N a_i \right)^2$$

$$(2) \quad f \left(\sum_{i=1}^n p_i x_i \right) \leq \sum_{i=1}^n p_i f(x_i)$$

Глава 14. Прямым текстом

В математических формулах \TeX автоматически дает курсивные буквы, но иногда бывают нужны обычные прямые буквы как в формуле

$$y = f(x + \text{constant})$$

\LaTeX позволяет временно отключить математический режим при помощи `\text`: приведенную выше формулу можно получить посредством

```
$$y=f(x+\text{constant})$$
```

В этой формуле вся конструкция `\text{constant}` трактуется как обычный символ вроде x или y , и соответственно определяются пробелы.

Заметим, что `\text` представляет собой управляющую последовательность с неким аргументом, только аргументом этим служит обычный текст. То, что следует за аргументом, опять будет представлено в математическом режиме. Таким образом, если вы наберете

```
$$f(x)=x^{17}+\text{lower order terms}+e^x$$
```

то получите

$$f(x) = x^{17} + \text{lower order terms} + e^x$$

• Упражнение 14.1:

Наберите следующее:

$$g(x) = f(x + \text{constant}) + f(x - \text{constant})$$

Внутри `\text`, как и в обычном тексте, вы можете менять шрифты. Например, если вы наберете

```
$$f(x)=x^{17}+\text{terms of {\it different\} order.}$$
```

то получите

$$f(x) = x^{17} + \text{terms of } \textit{different} \text{ order.}$$

Здесь `{\it different\}` встретилось внутри `\text`, так что это не будет отличаться от результата, который получался при наборе `{\it different\}` в тексте. С другой стороны, если мы наберем просто `$different$`, то просто получим формулу *different*, которая выглядит чрезвычайно странно, частично из-за курсивных букв, использование которых в математическом режиме несколько расширено, но в основном из-за пробелов, больше подходящих для формул, нежели для текста, во многих случаях *сильно отличающихся*.

• **Упражнение 14.2:**

Что здесь набрано не так?

```
$$y=f(x+{\text constant})$$
```

(По некоторой причине это весьма распространенная ошибка новичков в AMS-TeX ’е.)

Когда вы перемежаете формулы с текстовыми вставками, важно помнить, что в математическом режиме пробелы всегда исчезают. Так что если нужно набрать выключную формулу

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

лучше не набирать ее как

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

что дает

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

а набрать

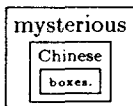
```
$$\Gamma(n)=(n-1)!\qquad\text{когда } n \text{ \text{целое}}$$
```

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

К счастью, имеется гораздо более естественный способ получать правильные пробелы в подобных ситуациях. Внутри `\text` можно не только переключать шрифты, но и вернуться в математический режим, так что вы можете иметь математику, внутри которой находится текст с математикой. Таким образом, допускается такой набор: набрать

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

Поначалу это может слегка обескуражить, но вскоре вы привыкнете к идее вложения одного режима в другой по схеме волшебных китайских шаров:



• **Упражнение 14.3:**

Наберите следующее:

$$F_n = F_{n-1} + F_{n-2} \quad \text{для любого } n > 1.$$

• **Упражнение 14.4:**

Наберите формулу (1) из упр. 12.3 в виде $1\text{ ml} = 1.000028\text{ cc}$ и формулу (3) в виде $0^\circ\text{ C} = 32^\circ\text{ F}$.

• **Упражнение 14.5:**

Как вы думаете, что будет, если вы повторите ошибку из упр. 14.2 и наберете

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

Между математической формулой $\Gamma(n) = (n-1)!$ из обычного текста и формулой, которая находится в `\texttt{\textcode{Gamma}(n)=(n-1)!}` в выключной формуле, имеется одно важное отличие. В последнем случае математическая формула автоматически будет сделана в размере `d-size`. Таким образом, если вы набрали

$f(a) > f(b) \quad \text{\texttt{\textcode{f(a)>f(b)}}}$ provided that $\frac{a}{b+1} > \sqrt{3}$.

то получите

$$f(a) > f(b) \quad \text{provided that } \frac{a}{b+1} > \sqrt{3}.$$

Так как мы находимся в режиме “выключных формул”, в котором уже предусмотрены дополнительные пробелы, то не имеет особого смысла экономить место, переводя формулу $\frac{a}{b+1} > \sqrt{3}$ в размер `t-size`. Таким образом, `\mathcal{AMS-TeX}` автоматически дает `d-size` (хотя вы всегда можете получить `t-size`, используя `\tsize...` внутри `\texttt{\textcode{}}`).

• **Упражнение 14.6:**

Наберите следующее:

Имеем

$$\frac{(n+1)^{p+1}}{p+1} = \sum_{k=1}^n k^p + \text{члены, содержащие } \sum_{k=1}^n k^r \text{ для } r < p.$$

По индукции выводим, что

$$\frac{\sum_{k=1}^n k^p}{n^{p+1}} = \frac{1}{p+1} + \text{члены с отрицательными степенями } n.$$

Очень часто математики позволяют тексту свободно блуждать внутри их выключных формул, так что иногда вам придется приложить немало усилий для выбора правильных пробелов. На помощь должны прийти авторы, указывающие, где какой (большой или меньший) пробел подходит, но Техническим наборщикам надо еще обладать воображением, чтобы перевести их пожелания в стандартные для \TeX 'а расстановки пробелов.

• **Упражнение 14.7:**

Подумайте, как могли быть набраны выключные формулы

$$X_n = X_k \quad \text{if and only if} \quad Y_n = Y_k \quad \text{and} \quad Z_n = Z_k$$

и

$$Y_n = X_n + 1 \quad \text{and} \quad Z_n = X_n - 1 \quad \text{for all } n \geq 0.$$

Между `\text` и обычным текстом имеется еще одно немаловажное отличие: `\text{...}` создает только одну строку текста, неразбиваемый элемент, который нельзя оформить в виде абзаца. Как правило, именно это нам и нужно в выключной формуле, но иногда дополнительное условие бывает таким длинным, что из него приходится делать небольшой абзац. Это обстоятельство обсуждается в части 3.

Если `\text` используется внутри невыключной формулы, любые математические формулы `$. . $.` внутри `\text` окажутся размера *t-size*. Но идея так использовать `\text` не блестяща, поскольку у вас образуется неразрывная строка текста, в которой с большой вероятностью получится `Overfull box`. Обычно лучше всего — входить и выходить из математического режима.

• **Упражнение 14.8:**

Какой способ набора следующего утверждения самый лучший?

Каждый хотел бы знать, будет ли множество $\{p : p \text{ и } p + 2 \text{ простые}\}$ бесконечным или нет!

(Используйте в этой формуле обычное двоеточие `:`, но вставьте тонкие шпации между фигурными скобками и остальной частью формулы — это “теоретико-множественные обозначения”, обсуждаемые в части 3.)

Хотя для формул в тексте `\text` не очень полезное средство, оно чрезвычайно удобно в индексах и в индексах второго порядка, потому что прямые буквы меняют свой размер точно так же, как и курсивные буквы математических формул. (Полужирные буквы тоже меняют размер, а наклонные — нет, потому что в шрифтах нет наклонных букв соответствующего размера.)

• **Упражнение 14.9:**

Как можно набрать фразу “the n^{th} Fibonacci number F_n ”?

• **Упражнение 14.10:**

Наберите такую выключную формулу:

$$\sum_{l \text{ odd}} \binom{n}{l} = 2^{n-1}$$

• **Упражнение 14.11:**

Хотя обозначения f' , f'' и f''' общеприняты, многие полиграфисты предпочитают $f^{(iv)}$, $f^{(v)}$, $f^{(vi)}$, ... вместо f'''' , f''''' , f'''''' , ... , Потому что очень утомительно считать все эти '. Как бы вы набирали такие формулы?

• **Упражнение 14.12:**

Что произойдет, если в вашем файле вместо $\$2^{\{\text{text}\}\{\text{nd}\}}\$$ окажется $\$2^{\text{text}\{\text{nd}\}}\$$?

Мы полагаем, что $\backslash\text{text}$ всегда представляет все прямым текстом. Это не совсем верно. Когда вы используете $\backslash\text{text}$ в выключных формулах или для однострочных формул, шрифт, который используется, — это “текущий шрифт”. Например, если в формулировке теоремы встретилась выключная формула, а вы при этом работали в стиле `amsrpt`, при котором `\proclaim` предлагает наклонное начертание, то $\backslash\text{text}$ также установит наклонное начертание. Если вы переключитесь на иной стиль, в котором для `\proclaim` отведен иной шрифт, шрифт для $\backslash\text{text}$ автоматически изменится. Именно это и бывает нужно, когда $\backslash\text{text}$ используется в условии. Если почему-либо требуются именно прямые буквы, можно набрать $\backslash\text{text}\{\text{rm}...\}$, когда $\backslash\text{text}$ появляется в конструкциях, которые могут изменить шрифт. С другой стороны, когда $\backslash\text{text}$ используется в индексах, автоматически выбирается прямой шрифт, поскольку в этом случае вы, вероятно, потому обращаетесь к $\backslash\text{text}$, чтобы явным образом получить прямые буквы в математическом режиме.

Теперь вы можете решить, что любой прямой шрифт в формулах, какой бы вам ни встретился, можно получать исключительно при помощи $\backslash\text{text}$. Но прямым буквам в математических формулах отводится особая роль, совершенно иная чем в обычном тексте. Такие операции, как “суммирование” и “интегрирование” обозначаются специальными символами \sum и \int , но для многих других математических операций, таких как “sine” (синус), “cosine” (косинус) и “logarithm” (логарифм) приняты аббревиатуры — “sin”, “cos” и “log”. В таких случаях обычно используется прямой набор, но было бы обременительно каждый раз, как только нам понадобится “sin”, набирать $\backslash\text{text}\{\text{sin}\}$. Еще более обременительно было бы подбирать нужные пробелы в формулах $\sin 2x$ и $\sin(2x)$ [в первом случае после ‘sin’ стоит тонкая шпация, а во втором ее нет]. Поэтому у $\mathcal{A}\mathcal{M}\mathcal{S}\text{-}\mathcal{T}\mathcal{E}\mathcal{X}$ ’а есть управляющие последовательности $\backslash\text{sin}$, $\backslash\text{cos}$ и $\backslash\text{log}$, которые все это делают за вас:

$$\begin{array}{ll} \$\sin 2\theta = 2\sin\theta\cos\theta\$ & \sin 2\theta = 2\sin\theta\cos\theta \\ \$\sin(2x) = 2\sin x\cos x\$ & \sin(2x) = 2\sin x\cos x \\ \$x = e^{\backslash,\backslash\log x}\$ & x = e^{\log x} \end{array}$$

(Последний случай демонстрирует ситуацию, в которой тонкая шпация может быть полезна.)

В приведенных выше примерах операции трактуются скорее как \int , нежели как \sum : любые верхние и нижние индексы располагаются сбоку от них, в том числе и в выключных формулах:

$$\begin{array}{ll} \$\sin^2 x + \cos^2 x = 1\$ & \sin^2 x + \cos^2 x = 1 \\ \$\log_2 x = (\log_2 e)(\log x)\$ & \log_2 x = (\log_2 e)(\log x) \end{array}$$

Обратите внимание, что \sin^2 , \cos^2 и \log_2 автоматически истолковываются как новые математические операции, так что в выражение $\sin^2 x$ тонкая шпация заложена заранее.

Есть и другие общепотребительные математические операции, которые истолковываются как \sum , т. е. верхние и нижние индексы для них в выключных формулах располагаются над и под знаком операции:

$$\begin{aligned} \max_{1 < n < m} \log_2 P_n & \qquad \max_{1 < n < m} \log_2 P \\ \lim_{x \rightarrow 0} \frac{\sin x}{x} & = 1 \\ \frac{\max_{1 < n < m} \log_2 P_n}{\lim_{x \rightarrow 0} (\sin x/x)} & \qquad \frac{\max_{1 < n < m} \log_2 P_n}{\lim_{x \rightarrow 0} (\sin x/x)} \end{aligned}$$

• **Упражнение 14.13:**

Что нужно изменить в последнем примере, чтобы получилось

$$\frac{\max_{1 < n < m} \log_2 P_n}{\lim_{x \rightarrow 0} \frac{\sin x}{x}}$$

• **Упражнение 14.14:**

Объясните, как набрать такие формулы:

$$(1) \quad \lim_{x \rightarrow 0} \frac{\sin^2 ax}{\sin^2 bx} = \left(\frac{a}{b}\right)^2$$

$$(2) \quad \lim_{x \rightarrow \infty} x \sin \frac{1}{x} = \lim_{x \rightarrow \infty} \frac{\sin \frac{1}{x}}{\frac{1}{x}} = \lim_{x \rightarrow 0^+} \frac{\sin x}{x}$$

$$(3) \quad \frac{1}{2} + \cos x + \cos 2x + \cdots + \cos nx = \frac{\sin(n + \frac{1}{2})x}{2 \sin \frac{x}{2}}$$

$$(4) \quad (\log \circ f)' = f'/f$$

$$(5) \quad \lim_{x \rightarrow 0^+} x(\log x)^n = 0$$

$$(6) \quad \lim_{h \rightarrow 0^+} \int_{-1}^1 \frac{h}{h^2 + x^2} = \lim_{h \rightarrow 0^+} \arctan \frac{x}{h} \Big|_{-1}^1 = \pi$$

$$(7) \quad \arctan \frac{1}{2} + \arctan \frac{1}{3} = \arctan \left(\frac{\frac{1}{2} + \frac{1}{3}}{1 - \frac{1}{6}} \right) = \frac{\pi}{4}$$

$$(8) \quad l - m = \lim_{\substack{n \rightarrow \infty \\ n \text{ even}}} \frac{2 - a_n^2}{1 + a_n} = \frac{2 - m^2}{1 + m}$$

• **Упражнение 14.15:**

Как бы вы справились с этой жуткой формулой (которую действительно можно встретить в ответах в задачниках по математическому анализу)?

$$\delta = \min \left(\sin^2 \left(\frac{[\min(1, \varepsilon/10)]^2}{9} \right) + \min(1, \varepsilon/10), [\min(1, \varepsilon/6)]^2 \right)$$

• **Упражнение 14.16:**

Кстати, о жутких формулах Как бы вы набрали такие:

$$\int_{k\pi+\pi/2-\delta}^{k\pi+\pi/2+\delta} \left| \frac{\sin x}{x} \right| dx \geq \frac{\delta}{k\pi + \pi/2}$$

и

$$\lim_{x \rightarrow \infty} \frac{\int_x^{x+\frac{\log x}{2x}} e^{t^2} dt}{e^{x^2}} = \lim_{x \rightarrow \infty} \frac{e^{(x+\frac{\log x}{2x})^2} - e^{x^2}}{2xe^{x^2}} = \frac{1}{2}$$

Можно использовать `\limits` и `\nolimits` после `\max` или `\lim`, с тем же эффектом, какой они дают для `\sum`. (Но `\limits` и `\nolimits` не дают такого результата после `\sin`, `\cos` и `\log`; у этих операторов никогда не бывает верхних и нижних индексов в качестве пределов.)

• **Упражнение 14.17:**

Снова выполните упр. 14.13, используя `\limits`.

Далее приводится список известных \LaTeX -у операторов; те, для которых “пределы” устанавливаются как верхние и нижние индексы, помечены буквой (L):

<code>\arccos</code>	<code>\cot</code>	<code>\exp</code>	(L) <code>\lim</code>	<code>\sec</code>
<code>\arcsin</code>	<code>\coth</code>	(L) <code>\gcd</code>	<code>\ln</code>	<code>\sin</code>
<code>\arctan</code>	<code>\csc</code>	<code>\hom</code>	<code>\log</code>	<code>\sinh</code>
<code>\arg</code>	<code>\deg</code>	(L) <code>\inf</code>	(L) <code>\max</code>	(L) <code>\sup</code>
<code>\cos</code>	(L) <code>\det</code>	<code>\ker</code>	(L) <code>\min</code>	<code>\tan</code>
<code>\cosh</code>	<code>\dim</code>	<code>\lg</code>	(L) <code>\Pr</code>	<code>\tanh</code>

Все эти управляющие последовательности соответствуют названиям этих операций. \LaTeX имеет также `\liminf`, `\limsup`, `\injlim` и `\projlim`, которые дают операции ‘`\lim inf`’, ‘`\lim sup`’, ‘`\inj lim`’ и ‘`\proj lim`’. Некоторые математики используют иные обозначения:

<code>\varliminf</code>	\varliminf
<code>\varlimsup</code>	\varlimsup
<code>\varinjlim</code>	\varinjlim
<code>\varprojlim</code>	\varprojlim

Этот список довольно впечатляющий, но вряд ли исчерпывающий, потому что математики продолжают изобретать новые операторы для своих собственных целей. Если в рукописи появляются постоянно повторяющиеся аббревиатуры вроде ‘`\trac`’, ‘`\Tor`’, ‘`\ISO`’ или ‘`\Res`’, есть все основания полагать, что автор использует их в качестве новых операторов. Чтобы \TeX представил `\Tor` в прямом начертании и трактовал его как новый оператор, нужно набрать лишь

`\operatorname{\Tor}`

А `\operatornamewithlimits` используется, если нужен оператор ‘Res’ с “пределами”:

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

Разумеется, вы не захотите постоянно набирать такие чудовищные сооружения. Если ‘To’ или ‘Res’ встречаются в статье несколько раз, имеет смысл определить свою собственную управляющую последовательность `\To` или `\Res`, как это объясняется в части 3.

Если `\operatorname` состоит из двух частей, как ‘liminf’ и ‘limsup’, то такое выражение выглядит лучше, если между этими частями только тонкая шпация `\,`.

• **Упражнение 14.18:**

Некоторые авторы вместо \sinh^{-1} используют ‘argsinh’. Как бы вы это набрали?

Хотя в `\operatorname` и `\operatornamewithlimits` должны вставляться особые пробелы, все значки вроде ‘или’ или * или - или / истолковываются как в тексте, при этом ‘и’ дают “и”, а -- и --- дают – и —. За точками, запятыми и двоеточиями будет оставлено небольшое пространство.

Кроме особого пробела, который дают последовательности типа `\sin`, `\cos`, ... , и всяких новых штук, поставляемых `\operatorname`, есть еще одно важное отличие этих операторов от `\text`: они всегда идут в прямом начертании, даже если основной текст был набран как-то иначе.

• **Упражнение 14.19:**

Иногда названия операторов должны быть представлены в другом начертании. Например, многие математики вместо $SO(n)$ и $SU(n)$ предпочитают $SO(n)$ и $SU(n)$, а другие — $SO(n)$ и $SU(n)$. Как это получить?

Глава 15. Держите равнение

Мы теперь научились составлять большинство комбинаций математических символов, которые могут понадобиться в формулах, а для получения выключных формул сможем воспользоваться знаками `$$`. Но в математических работах часто содержатся различные комбинации формул в выключном виде, которые требуют специальной обработки.

Прежде всего, математики любят снабжать такие формулы номерами, или привешивать им еще какие-нибудь метки. В некоторых журналах эти метки ('tags') принято располагать слева от формулы

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

тогда как в других — справа:

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

Чтобы получить такую нумерованную формулу, достаточно набрать

```
$$x=y\tag3-1$$
```

`AMS-TeX` автоматически выберет подходящее место для метки, а если формула окажется слишком длинной, поместит ее на отдельной строке:

```
(3-1)      очень длинная формула ...                ... оканчивающаяся здесь
```

Заметьте, что нет необходимости набирать `... \tag{3-1}$$` с фигурными скобками вокруг метки: `AMS-TeX` знает, что метка — это все, что располагается между `\tag` и знаками `$$`. Обратите внимание также, что скобки вокруг метки ставятся автоматически; в некоторых стилях могут потребоваться иные выделения, например `[3-1]` или `3-1` и т. д. И последнее: метка обрабатывается как обычный текст, а не как формулы в математическом режиме, так что `-` и `--` дают дефис и короткое тире, а не знаки минуса.

• Упражнение 15.1:

Математик З. Акорючка любит расставлять после выключных формул знаки препинания. Но после `E` попытки набрать

$$(3-2) \quad x = y,$$

получилось

$$((3-2),) \quad x = y$$

Какую ошибку `O` допускает?

• Упражнение 15.2:

Как нужно поступить, если понадобилась метка типа `A'`?

• **Упражнение 15.3:**

А как быть с метками вроде (*), (**) и (***)?

◊ В стиле `amsppt` метки помещаются слева, но в части 3 будет рассказано, как получить метку справа, если это понадобится.

Часто бывает нужно выровнять выключные формулы по какому-нибудь символу. Например, рассмотрим следующие выключные формулы:

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

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

Здесь делается выравнивание по знакам `=`, а затем обе формулы располагаются по центру как одно целое. Это достигается при помощи команды `AMS-TeX`'а `\align`:

```

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

```

Отдельные формулы между `\align` и `\endalign` разделяются посредством `\\` (за последней формулой `\\` ставить не надо). В каждой формуле также содержится знак `&`, который ставится между той частью формулы, что должна быть слева, и той, что будет справа. Метка `\tag` в любой из этих формул может быть, а может и отсутствовать.

При использовании `\align... \endalign` `TeX` выстраивает отдельные формулы, а затем превращает каждую из них в строку на всю ширину страницы (хотя в такой "строке" может быть полно пустых мест). Таким образом, если вы попробуете набрать что-то вроде

```

$$
A\align... \endalign B
$$

```

`TeX` выдаст сообщение об ошибке. Нельзя ничего помещать между первыми `$$` и `\align`, а также между `\endalign` и последними знаками `$$`.

В приведенных выше примерах мы постарались создать красиво выровненные объекты для облегчения чтения, но, разумеется, это вовсе не обязательно (и не очень практично, если в формулах достаточно много символов). Но чтобы помочь `\align` расставить правильно пробелы, следует помнить, что нужно правильно набирать знаки `&`, непосредственно перед символами, по которым должно делаться выравнивание. Обратите также внимание на размещение точки и запятой.

• **Упражнение 15.4:**

Как бы вы набрали следующее:

$$(1) \quad Q^l = Q_1 \left\{ \sum_k (-1)^k (PQ_1 - I)^k \right\}$$

$$(1_r) \quad Q^r = \left\{ \sum_k (-1)^k (Q_1 P - I)^k \right\} Q_1$$

• **Упражнение 15.5:**

Что произойдет, если вы пропустите знак & во второй формуле?

• **Упражнение 15.6:**

Наберите следующее:

$$\alpha_4 = \sqrt{\frac{1}{2}}$$

$$\alpha_8 = \sqrt{\frac{1}{2} + \frac{1}{2}\sqrt{\frac{1}{2}}}$$

$$\alpha_{16} = \sqrt{\frac{1}{2} + \frac{1}{2}\sqrt{\frac{1}{2} + \frac{1}{2}\sqrt{\frac{1}{2}}}}$$

etc.

Выравниваемые формулы обычно воспринимаются как одно целое, так что \LaTeX , как правило, не позволяет переносить на другую страницу часть формулы между `\align... \endalign`. Однако, как вы узнаете из части 3, существует масса способов подсказать \LaTeX у, что можно разорвать выключную формулу.

`\align` предоставляет возможность получать выровненные формулы, растянутые на всю ширину страницы, но математики любят иногда сгрудить несколько выровненных формул в один выключной формульный массив:

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

Для обработки такого выключного материала \LaTeX предоставляет конструкцию `\aligned... \endaligned`. В отличие от команды `\align`, которая указывает \TeX у выровнять последовательность строк по всей ширине страницы, конструкция `\aligned... \endaligned` создает один выровненный массив, который рассчитывается на ширину, достаточную для всех входящих в него формул, и с которым можно обращаться, как с любым другим символом. Например, приведенная выше выключная формула была набрана так:

```

$$
\left\{
\aligned \alpha&=f(z)\ \ \ \beta&=f(z^2)\ \ \
\gamma&=f(z^3)\endaligned
\right\}\quad\left\{
\aligned x&=\alpha^2-\beta\ \ \ y&=2\gamma\endaligned \right\}.
$$

```

• **Упражнение 15.7:**

Наберите следующую выключную формулу:

$$k_1, k_2 = H \pm \sqrt{H^2 - K} \quad \text{where} \quad \begin{cases} K = \frac{eg - f^2}{EG - F^2} \\ H = \frac{Eg - 2Ff + Ge}{2(EG - F^2)}. \end{cases}$$

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 вы можете поставить `\tag` после `\endaligned`. Например, ввод

```


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


```

дает

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

Обратите внимание, что метка теперь находится в центре всего выключного массива, выровненного посредством `\aligned`.

• **Упражнение 15.8:**

Как следует набрать такую нумерованную двустрочную выключную формулу?

$$(23) \quad \begin{aligned} K &= \frac{eg - f^2}{EG - F^2} \\ H &= \frac{Eg - 2Ff + Ge}{2(EG - F^2)} \end{aligned}$$

• **Упражнение 15.9:**

Посмотрите, что произойдет, если ваш файл будет состоять из единственной строки

```


$$\aligned A&=B \ \ C&=D$$


```

с пропущенным `\endaligned`.

Время от времени автору хочется объединить вместе несколько формул и выключить их все по центру, не выравнивая:

$$\begin{aligned} a &= b + c \\ d &= e \\ f + g &= h \end{aligned}$$

Если вы наберете их как отдельные выключные формулы,

$$a=b+c \quad d=e \quad f+g=h$$

то между ними останется слишком много пространства. Так что лучше пользуйтесь `\gather`:

$$\begin{aligned} & \\ \gather a=b+c \ \ d=e \ \\ & f+g=h \end{gather}$$

Как обычно, `\` указывает переход на другую строку, но внутри `\gather` знаков `&` вовсе быть не должно, поскольку выравнивать здесь нечего. Как и в случае `\align`, каждой формуле можно дать свою метку `\tag`.

• **Упражнение 15.10:**

Наберите следующее:

$$(3-2) \quad g = \det(g_{ij})$$

$$(3-3) \quad g^{kl} = (k, l) \text{ entry of the inverse matrix of } (g_{ij})$$

$\mathcal{A}\mathcal{M}\mathcal{S}\text{-}\mathcal{T}\mathcal{E}\mathcal{X}$ 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.

• **Упражнение 15.11:**

Наберите следующее:

Имеем $(a + bi)^2 = \alpha + \beta i$ тогда и только тогда, когда

$$(*) \quad \begin{aligned} a^2 - b^2 &= \alpha \\ 2ab &= \beta, \end{aligned}$$

Решением последнего служат

$$\left. \begin{aligned} a &= \sqrt{2\alpha + 2\sqrt{\alpha^2 + \beta^2}} \\ b &= \frac{\beta}{2\sqrt{2\alpha + 2\sqrt{\alpha^2 + \beta^2}}} \end{aligned} \right\} \text{ или } \left\{ \begin{aligned} a &= -\sqrt{2\alpha + 2\sqrt{\alpha^2 + \beta^2}} \\ b &= \frac{-\beta}{2\sqrt{2\alpha + 2\sqrt{\alpha^2 + \beta^2}}} \end{aligned} \right.$$

Глава 16. Столько всяких удобств

Иногда выключную формулу невозможно напечатать в одну строку, потому что она в нее не помещается, несмотря на все усилия \TeX 'а сжать входящие в нее члены:

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

Эта формула, в сущности, составлена из нескольких связанных между собой более мелких формул, и в таких случаях принято разбивать выражение на несколько строк, выравнивая по символам бинарных операций. Таким образом, даже более длинные формулы можно представить в виде одной выключной:

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

• Упражнение 16.1:

Как, вы думаете, была набрана эта выключная формула?

В упр. 16.1 показано, как некоторая изобретательность приводит к далеко идущим последствиям, но дело усложняется, как только разбитая на части формула снабжается меткой. Если метки помещаются слева, то формула обычно набирается так:

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

Но если метки справа, то ее следует набирать по-другому:

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

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

```


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


```

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

Заметьте, что метку `\tag` надо помещать после **всей** конструкции `\split... \endsplit`; если вы 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 в тексте формулы разрывают *после* бинарных операций.

• Упражнение 16.2:

Предположим, вам нужен такой набор в $\mathcal{T}\mathcal{E}\mathcal{X}$ 'е:

$$(1) \quad (a+b)(a+b) = a^2 + 2ab + b^2,$$

$$(2) \quad (a+b)(a-b) = (a+b)a - (a+b)b \\ = a^2 + ab - ab - b^2 \\ = a^2 - b^2.$$

если метки располагаются слева, и набор

$$(a+b)(a+b) = a^2 + 2ab + b^2, \tag{1}$$

$$(a+b)(a-b) = (a+b)a - (a+b)b \\ = a^2 + ab - ab - b^2 \\ = a^2 - b^2. \tag{2}$$

если метки справа. Какой ввод был использован, как вы думаете?

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

$$(f \circ g)'''(x) = [f'''(g(x)) \cdot g'(x)^3 + 2f''(g(x)) \cdot g'(x)g''(x)] \\ + [f''(g(x)) \cdot g'(x)g''(x) + f'(g(x)) \cdot g'''(x)]$$

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

```


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


```

так что `=\bigl[` выровнено с невидимым `\quad`. Снова отметьте себе, что такой разрыв обычно делают *перед* знаком бинарных операций, но формулы в тексте, как правило, разрывают *после* бинарных операций.¹

• Упражнение 16.3:

Как было набрано следующее:

$$\Delta = [a + b + c]^n (a_{11} + b_{11} + c_{11} + a_{12} + b_{12} + c_{12} + a_{22} + b_{22} + c_{22}).$$

Можно было бы создать обширный свод правил для разрыва и расположения выключных формул, но эти правила варьируются от издателя к издателю, и все равно они не будут адекватны. Т_ЕX даже не делает никаких попыток разорвать длинную выключную формулу, потому что это целое искусство — авторы математических рукописей должны сами определить все разрывы, поскольку это зависит от неуловимых нюансов математического представления. (Разумеется, часто автору бывает трудно предугадать, что понадобится разрыв, так что мнение опытного Т_ЕXнического наборщика может также быть очень ценным.)

Следующая выключная формула иллюстрирует еще одно важное соглашение, о котором вам следует знать:

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

Здесь первая строка сдвинута почти вплотную влево, а вторая — почти вплотную вправо. Это было набрано при помощи еще одной команды $\mathcal{A}\mathcal{M}\mathcal{S}$ -Т_ЕX'a — `\multline`:

```


$$\begin{aligned} & \int_a^b \biggl\{ \int_a^b [f(x)^2 g(y)^2 + f(y)^2 g(x)^2 \\ & \quad - 2f(x)g(x)f(y)g(y) dx \biggr\} dy \\ & = \int_a^b \biggl\{ g(y)^2 \int_a^b f^2 + f(y)^2 \int_a^b g^2 \\ & \quad - 2f(y)g(y) \int_a^b fg \biggr\} dy \end{aligned}$$


```

¹В отечественной литературе на русском языке знаки бинарных операций при разрыве формул как выключных, так и в тексте, принято повторять. — Прим. перев.

Точное расстояние от полей определяется форматом. В стиле `amsppt` закладывается один квадрат; в некоторых форматах пробел может вовсе отсутствовать.

После `\multline... \endmultline` можно задать метку `\tag`, точно так же, как и после `\split... \endsplit`. Например,

```


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


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


```

дает либо

$$(17) \quad \int_a^b \left\{ \int_a^b [f(x)^2 g(y)^2 + f(y)^2 g(x)^2 - 2f(x)g(x)f(y)g(y)] dx \right\} dy \\ = \int_a^b \left\{ g(y)^2 \int_a^b f^2 + f(y)^2 \int_a^b g^2 - 2f(y)g(y) \int_a^b fg \right\} dy$$

либо

$$\int_a^b \left\{ \int_a^b [f(x)^2 g(y)^2 + f(y)^2 g(x)^2 - 2f(x)g(x)f(y)g(y)] dx \right\} dy \\ = \int_a^b \left\{ g(y)^2 \int_a^b f^2 + f(y)^2 \int_a^b g^2 - 2f(y)g(y) \int_a^b fg \right\} dy \quad (17)$$

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 `$$` и `\multline`, ни между `\endmultline` и вторым `$$` ничего быть не должно.

• Упражнение 16.4:

Наберите выключную формулу

$$f^{(k)}(x) = e^{-1/x^2} \left[\sum_{i=1}^{3k} \frac{a_i}{x^i} \sin \frac{1}{x} + \sum_{i=1}^{3k} \frac{b_i}{x^i} \cos \frac{1}{x} \right]$$

for some numbers $a_1, \dots, a_{3k}, b_1, \dots, b_{3k}$.

Как явствует из названия, `\multline` можно использовать для создания многострочных выключных формул. Все строки, за исключением первой и последней, выключаются по центру, хотя есть способ сдвинуть любую из них влево или вправо. Подробнее об этом можно прочесть в части 3.

Разумеется, математики на самом деле любят создавать даже еще более вычурные выключные формулы, чем те, с которыми мы до сих пор встречались. Но это, к счастью, можно отложить до следующей, последней главы этой части.

Глава 17. Затруднительное положение

Математикам, безусловно, понравится такая наборная компьютерная система, которая позволяет получать всякого рода выключные формулы. Но возможность набирать “матрицы” их просто потрясет. Матрицы — это прямоугольные массивы, которые появляются в формулах вроде

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

Конструкция $\mathcal{A}\mathcal{M}\mathcal{S}\text{-}\mathcal{T}\mathcal{E}\mathcal{X}$ 'а `\matrix... \endmatrix` позволяет получать матрицы без особых усилий: для приведенной выше формулы мы просто наберем

```
$$  
A=\left( \matrix  
x-\lambda & 1 & 0\\  
0 & x-\lambda & 1\\  
0 & 0 & x-\lambda  
\endmatrix \right).  
$$
```

Обратите внимание, что `\matrix` слегка напоминает `\aligned` — в каждой строке между 1-м, 2-м, 3-м ... элементами мы помещаем `&`, а для разделения на строки используем `\\`. Но есть также и важные отличия: элементы каждого столбца в `\matrix` центрируются, а между столбцами закладывается расстояние `\quad`. Кроме того, размер элементов матрицы обычно устанавливается `t-size`, а между строками закладывается то же расстояние, как между строками текста, тогда как при `\align` или `\aligned` обычно устанавливается размер `d-size`, а строки раздвинуты чуть шире.

Математики редко используют матрицы, не заключенные в скобки или какие-нибудь иные ограничители. У $\mathcal{A}\mathcal{M}\mathcal{S}\text{-}\mathcal{T}\mathcal{E}\mathcal{X}$ 'а есть несколько других управляющих последовательностей, обеспечивающих такие ограничители. Приведенная выше выключная формула могла быть набрана так:

```
$$  
A=\pmatrix  
x-\lambda & 1 & 0\\  
0 & x-\lambda & 1\\  
0 & 0 & x-\lambda  
\endpmatrix.  
$$
```

Для получения квадратных скобок `\left[...\right]` вокруг матрицы имеется конструкция `\bmatrix... \endbmatrix`, для вертикальных черточек `\left|...\right|` — конструкция `\vmatrix... \endvmatrix`, а для двойных вертикальных

черточек `\left\{...\right\}` — средство `\Vmatrix... \endVmatrix`. Внимание: если вы начнете формулу с `\matrix`, а закончите ее `\endmatrix`, то, в сущности, зададите формулу с открывающей скобкой `\left(` (но без парного закрывающего правого ограничителя `\right(ограничитель)`), так что вам будет выдано сообщение об ошибке.

• **Упражнение 17.1:**

Наберите следующее:

$$(I.3) \quad \begin{pmatrix} a & b \\ c & d \end{pmatrix} \cdot \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} = \begin{pmatrix} a \cdot 0 + b \cdot 1 & a \cdot 1 + b \cdot 0 \\ a \cdot 0 + d \cdot 1 & c \cdot 1 + d \cdot 0 \end{pmatrix} \\ = \begin{pmatrix} b & a \\ d & c \end{pmatrix}$$

• **Упражнение 17.2:**

Наберите такую матрицу, составленную из матриц:

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

где между столбцами заложено по два квадрата, а между строками — по две пробельные строки.

• **Упражнение 17.3:**

Как можно было бы набрать такую матрицу? *Указание:* В $\mathcal{A}\mathcal{M}\mathcal{S}\text{-}\mathcal{T}\mathcal{E}\mathcal{X}$ е есть команда `\hdots` для получения горизонтальных отточий, `\vdots` — для вертикальных и `\ddots` — для диагональных.

$$\begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{pmatrix}$$

В некоторых ситуациях центрирование элементов `\matrix` в каждом столбце может быть нежелательным. Например, в такой матрице в круглых скобках

$$\begin{pmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{pmatrix}$$

некоторые могут предпочесть, чтобы элементы выровнились по правому краю столбцов, а в матрице

$$\begin{array}{rcc} x & .1 & 1 \\ x + y & .11 & 11 \\ x + y + z & .111 & 111 \end{array}$$

десятичные дроби выровнены по левому краю, тогда как числа третьего столбца — по правому. $\mathcal{A}\mathcal{M}\mathcal{S}\text{-}\mathcal{T}\mathcal{E}\mathcal{X}$ обладает особой конструкцией `\format... \ll` для

получения матриц какого-нибудь нового формата. Приведенная выше матрица была набрана следующим образом:

```


$$\begin{matrix} \text{\format\c\quad\l\quad\r} \\ \text{x\&.1\&1} \\ \dots \\ \end{matrix}$$


```

Строка `\format\c\quad\l\quad\r` задает новый формат; `\c` выделяет столбец, в котором элементы центрируются, `\quad\l` соответствует столбцу, элементы которого выровнены по левому краю с предшествующим им пустым столбцом, шириной в `\quad`, а `\quad\r` соответствует столбцу с выравниванием по правому краю тоже с предшествующим пустым столбцом ширины `\quad`. Можно использовать `\format` также с `\matrix` и т. д.

• **Упражнение 17.4:**

Наберите следующую формулу:

$$\begin{pmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{pmatrix} \begin{pmatrix} \cos \phi & \sin \phi \\ -\sin \phi & \cos \phi \end{pmatrix} = \begin{pmatrix} \cos \rho & \sin \rho \\ -\sin \rho & \cos \rho \end{pmatrix}, \quad \rho = \theta + \phi.$$

• **Упражнение 17.5:**

Наберите такой столбец чисел:

```

3.14159
2.71828
1.61808
.57701

```

• **Упражнение 17.6:**

Наберите следующее выражение со столбцами, выровненными по левому краю:

$$\det \begin{vmatrix} c_0 & c_1 & c_2 & \dots & c_n \\ c_1 & c_2 & c_3 & \dots & c_{n+1} \\ c_2 & c_3 & c_4 & \dots & c_{n+2} \\ \vdots & \vdots & \vdots & & \vdots \\ c_n & c_{n+1} & c_{n+2} & \dots & c_{2n} \end{vmatrix} > 0.$$

• **Упражнение 17.7:**

Наберите следующее:

$$\omega = \begin{pmatrix} 0 & \omega_{12} & 0 & \dots & 0 \\ -\omega_{12} & 0 & \omega_{23} & 0 & \dots & 0 \\ 0 & -\omega_{23} & 0 & \omega_{34} & 0 & \dots & 0 \\ \vdots & & & \ddots & & & \vdots \\ 0 & \dots & & 0 & & \omega_{n-1,n} \\ 0 & \dots & 0 & -\omega_{n-1,n} & & 0 \end{pmatrix}$$

У \LaTeX 'а нет особой управляющей последовательности для получения матрицы в фигурных скобках $\{ \}$, потому что это почти никогда не бывает нужно. Но математики очень часто определяют что-либо путем перечисления случаев, используя конструкцию вроде

$$f(x) = \begin{cases} x + 1, & \text{for } x > 0 \\ x - 1, & \text{for } x \leq 0. \end{cases}$$

Вы могли бы получить это, используя подходящий формат `\format` и `\left\{` и `\right.`, но у \LaTeX 'а специально на этот предмет имеется особая конструкция `\cases... \endcases`, которая сделает все это за вас. Наберите просто

```


$$f(x) = \begin{cases} x+1, & \text{for } x > 0 \\ x-1, & \text{for } x \leq 0. \end{cases}$$


```

позабывшись о знаках препинания и помня об использовании `\text` при необходимости. Обратите внимание, что первый элемент не должен быть набран как `$x+1$`: \LaTeX и так обрабатывает элементы `\matrix` и `\cases` в математическом режиме, так что неожиданный знак `$` просто приведет к полному хаосу.

• **Упражнение 17.8:**

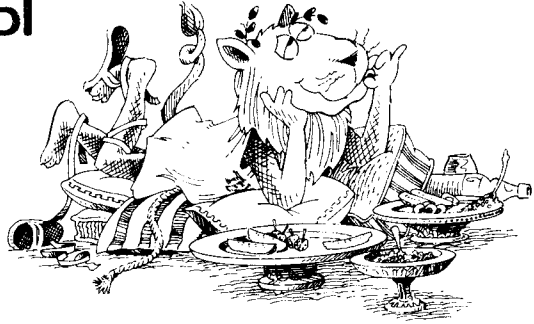
Наберите следующее:

$$f^{(k)}(x) = ax^{m-k} \sin \frac{1}{x} + \sum_{l=k+1}^{2k-1} \left(a_l x^{m-l} \sin \frac{1}{x} + b_l x^{m-l} \cos \frac{1}{x} \right) \pm \begin{cases} x^{m-2k} \sin \frac{1}{x}, & k \text{ even} \\ x^{m-2k} \cos \frac{1}{x}, & k \text{ odd.} \end{cases}$$

Часть 3

НАПИТКИ И ПРИПРАВЫ

для особых случаев



Глава 18. Практическое самоуправление

\LaTeX оснащен десятками управляющих последовательностей для обработки любых мыслимых и немыслимых математических формул. Но вы можете создавать еще и свои собственные новые управляющие последовательности для сокращения записи сложных выражений, слишком часто повторяющихся в статье. Например, если формула $\alpha^2 + \beta^2$ появляется снова и снова, вы можете пожелать вместо нее иметь дело с более коротким выражением вроде $\backslash ab$. Для этого нужно просто вставить в исходный файл

```
 $\backslash define\ab{\alpha^2+\beta^2}$ 
```

В результате этого \TeX станет подставлять $\backslash alpha^2+\backslash beta^2$ всякий раз, как только встретит $\backslash ab$, так что $\$ \backslash ab \$$ выдаст формулу $\alpha^2 + \beta^2$, а при наборе $\$ \$ \backslash ab . \$ \$$ получим выключную формулу

$$\alpha^2 + \beta^2.$$

• Упражнение 18.1:

Предположим, что $\backslash ab$ было определено именно так. Как бы вы набрали

$$\sqrt{\alpha^2 + \beta^2} + \sqrt{\gamma^2} \leq x + \frac{\alpha^2 + \beta^2}{\gamma}$$

с минимальными наборными затратами?

• Упражнение 18.2:

А как (это будет похитрее!) насчет формул $2^{\alpha^2+\beta^2}$ и $\Gamma_{\alpha^2+\beta^2}$?

• **Упражнение 18.3:**

Что произойдет, если в вашем файле есть строка

```
\define\ab{ \alpha^ 2 +\beta^ 2 }
```

Единственное, что можно определять при помощи `\define`, это управляющие последовательности, так что если мы по ошибке наберем

```
\define ab{\alpha^2+\beta^2}
```

то получим сообщение об ошибке. В действительности \LaTeX чрезвычайно придирчив к `\define`, и вы получите сообщение об ошибке, даже если наберете

```
\define{ab}{\alpha^2+\beta^2}
```

— фигурные скобки вокруг первой управляющей последовательности недопустимы! Если вы по одной из этих причин получите сообщение об ошибке и попытаетесь вынудить \LaTeX продолжить работу, нажав `(carriage-return)`, \TeX сочтет за лучшее проигнорировать `\define` и двинуться вперед. Но когда вы в дальнейшем воспользуетесь `\ab`, \TeX пожалуется, что эта управляющая последовательность не определена — неправильно выполненная `\define` приводит обычно к катастрофе, так что вам, по-видимому, следует немедленно выйти и все уладить.

В аббревиатуре `\ab` нет, разумеется, ничего специфического, и многие наборщики придумают немало совершенно других аббревиатур. Можно, например, просто воспользоваться `\1`, если это ваша первая аббревиатура и вы полагаете, что ее будет довольно легко запомнить. Но если вы выбрали такой управляющий символ, то следует соблюдать другую предосторожность. Набирая

```
\define\1{\alpha^2+\beta^2}
```

убедитесь, что вы не оставили пробела между `\1` и `{`; дополнительный пробел все испортит, а каким образом — расскажем позже.

Даже в обычном тексте использование `\define` для введения особых управляющих последовательностей может оказаться очень полезным. Например, если фамилии “de Rham” и “É. Cartan” в статье повторяются часто, можно определить

```
\define\deRham{de~Rham}
\define\Cartan{\'E.~Cartan}
```

что сэкономит массу ваших физических и умственных усилий.

• **Упражнение 18.4:**

Как использовать эти управляющие последовательности для набора следующего текста?

The de Rham cohomology ring $H^*(G)$ had, of course, already essentially been computed by É. Cartan.

• **Упражнение 18.5:**

Вы намереваетесь поведать людям о тех многих вещах, которые они **должны** делать, и поэтому вам бы хотелось иметь управляющую последовательность `\должны`, которая автоматически давала бы полужирное ‘**должны**’. Как бы вы ее определили с помощью `\define`? (Как и в упр. 18.2, здесь все не так просто.)

`\define` используется не только для замены некоторой сложной комбинации новой управляющей последовательностью; этим приемом можно воспользоваться и для стандартной управляющей последовательности, если ее название вам не нравится. Например, определение

```
\define\alpha
```

позволит вместо `\alpha` пользоваться коротким названием `\a`. Кроме всего прочего, это определение иллюстрирует еще одно место, где \LaTeX так привередлив по отношению к `\define`. Даже хотя `\a` представляет собой замену единственной управляющей последовательности `\alpha`, мы *должны* заключить `\alpha` в фигурные скобки; эти скобки существенны: без них `\define` не в состоянии понять, в чем же именно состоит определение, и если вы наберете `\define\a\alpha` — катастрофа неминуема.

• **Упражнение 18.6:**

Допустим, в вашем файле есть корректное определение `\define\alpha{\alpha}`. Как бы вы набрали такую формулу:

$$\alpha^\alpha + \alpha^{\alpha+\alpha} + \alpha^{\alpha^\alpha} + \sqrt{\alpha^2 + \alpha^3}$$

• **Упражнение 18.7:**

Что произошло бы, если бы в вашем файле было

```
\define\ab{\a^2+\beta^2}
```

• **Упражнение 18.8:**

Вы могли бы соблазниться определением

```
\define\alpha{\$alpha\$}
```

в надежде набирать в текстовом режиме просто `\a` и получать α . Почему это на самом деле плохая идея?

Хотя \TeX 'у было очень легко объяснить, что `\a` используется вместо `\alpha`, нас ожидает сюрприз при попытке определить

```
\define\b{\beta}
```

Встретив такое определение, \LaTeX откликнется сообщением об ошибке¹

```
! AmS-TeX error: \b is already defined.
```

Ах да, действительно, `\b` уже определялось: это управляющая последовательность для получения подчеркивания (`bar-under`) отдельной буквы. (Возможно,

¹Это сообщение поступает с небольшим объявлением, информирующим нас о том, что его произвел не сам \TeX а \LaTeX , но вы можете обращаться с ним точно так же, как и с другими сообщениями об ошибке.

вы никогда не читали о `\b`, потому что это было в гл. 6, которую вы, будучи снобом, пропустили. Не страшно, `\b` можно найти в указателе.)

Если, получив такое сообщение, вы нажмете `h` или `H`, то обнаружите, что `AMS-TeX` проигнорировал ваше `\define`, чтобы избежать конфликтной ситуации. Но это немного досадно, потому что β встречается часто, а подчеркивание отдельной буквы, может быть, никогда не понадобится. Вот почему `AMS-TeX` предоставляет вам и средство `\redefine`. Если вы поместите в файл

```
\redefine\b{\beta}
```

то все последующие `\b` будут заменены на `\beta`, несмотря на то, что `\b` изначально имело другой смысл. Если вы при этом хотите еще иногда делать одиночные подчеркивания, вам следует поступить так:

```
\predefine\barunder{\b}
\redefine\b{\beta}
```

Тогда управляющая последовательность, позволяющая подчеркивать одиночные буквы, получит новое имя `\barunder`, а новым именем для `\beta` станет `\b`. Так что теперь для получения β нужно набирать `$$b$`, а набрав

```
\barunder T`abit ibn Qorra
```

вы получите имя 'T`abit ibn Qorra'. Очень существенно, чтобы `\predefine` появилось в файле до `\redefine` (не отсюда ли и название `\predefine`?), но нет необходимости помещать их подряд. Обратите внимание, что `\predefine` значительно более специфичен, чем `\define` и `\redefine` — внутри фигурных скобок не должно быть ничего, кроме *единственной управляющей последовательности*.

• Упражнение 18.9:

Вероятно, вам также захочется ввести аббревиатуру для `\gamma`, третьей буквы греческого алфавита. Наборщики часто используют `\g`, но если вам по нраву репутация логично мыслящего оператора, под стать настоящему математику, лучше выбрать `\c`. Как вынудить `AMS-TeX` взять `\c` в качестве аббревиатуры для `\gamma`, причем так, чтобы не испортить `façade`?

Вы могли бы соблазниться идеей с самого начала воспользоваться `\redefine`, вместо того, чтобы дожидаться, признает ли `AMS-TeX` ваше первоначальное определение `\define`. Но этого не стоит делать ни в коем случае: `\define` позволяет `AMS-TeX` у предохранить вас от тех ошибок, о возможности которых вы даже и не подозревали. Например, предположим, что вы имеете дело с громоздким сложным математическим выражением и решаете обозначить его просто `\box`:

```
\define\box{...}
```

Когда `AMS-TeX` доберется до этой строки, он снова выдаст сообщение об ошибке, гласящем, что `\box` уже определялся:

```
! AmS-TeX error: \box is already defined.
```

Вы, по-видимому, `\box` не припоминаете, поскольку нигде ранее не встречали упоминания о нем, и поэтому заглядываете в указатель. В чем дело? И тут его нет! Оказывается, `\box` — это не та управляющая последовательность, которой вы могли когда-либо воспользоваться, а управляющая последовательность для внутренних нужд Т_ЭX'a, и довольно-таки важная. Если бы вы ввели `\redefine` для управляющей последовательности `\box`, то вызвали бы вселенский хаос. Поэтому **всегда** сначала примените `\define`, и если вы случайно наткнулись на управляющую последовательность, которая не представлена в указателе, ни при каких обстоятельствах **не** прибегайте к `\redefine`. (В указателе представлено несколько управляющих последовательностей, которые используются только для внутренних целей, но все они снабжены предостерегающей пометкой '[`\redefine` запрещено]').)

На самом деле `\box` — это исключение: большинство управляющих последовательностей, к которым нельзя применять `\redefine`, имеют такие странные имена, что они вряд ли придут вам в голову. Разумеется, существует не так уж много управляющих *символов*, а определено их еще меньше: это, например, `\`, и `\'` и `\&`. Кое-что, однако, еще осталось, в том числе `\0`, `\1`, ..., `\9`.

(Несмотря на только что сделанное предостережение, существует одна ситуация, когда безопасно и даже целесообразно использовать `\redefine` с первой попытки — это когда вы намереваетесь переопределить управляющую последовательность, которую вы сами изначально ввели посредством `\define`. Если было набрано

```
\define\ab{\alpha^2+\beta^2}
```

поскольку казалось, что статья изобилует комбинациями $\alpha^2 + \beta^2$, но вдруг они исчезли и всюду заперестрели $\alpha^3 + \beta^3$, то вы можете решить использовать

```
\redefine\ab{\alpha^3+\beta^3}
```

Необходимо именно `\redefine`, поскольку `\ab` уже было определено посредством `\define`.)

Хотя команда АМ_S-Т_ЭX'a `\define` старается обезопасить вас от ошибок, есть еще несколько предосторожностей, которые вам следует самостоятельно предпринять. Все будет в порядке, если управляющую последовательность определить посредством `\define` в терминах других управляющих последовательностей. Но **не** определяйте управляющую последовательность в терминах самой себя! Если в вашем входном файле содержится

```
\define\regress{бесконечный \regress}
```

то как только вы воспользуетесь `\regress` произойдет нечто ужасное: сначала Т_ЭX заменит `\regress` на *бесконечный* `\regress`, затем снова заменит `\regress`, получив уже *бесконечный бесконечный* `\regress`, затем опять заменит `\regress`, ... Вскоре Т_ЭX истощит запас своих возможностей хранить этот хлам, и вы получите тревожное сообщение об этом:

```
! TeX capacity exceeded, sorry. ...
```



Если вы, однако, используете

```
\define\selfreference{\selfreference}
```

то произойдет худшее: найдя в вашем файле `\selfreference`, `TeX` посмотрит, как это было определено, и затем заменит на определение, т. е. на `\selfreference`. Затем это `\selfreference` будет заменено на `\selfreference`, а это `\selfreference` — опять на `\selfreference`, ...! В подобном случае `TeX` никогда не исчерпает своих возможностей и заикнется — после нескольких миллиардов циклов вы, по-видимому, начнете недоумевать, почему на экране ничего не происходит. Здесь `TeX` можно остановить только при помощи системного прерывания.

• Упражнение 18.10:

Что произойдет, если задать

```
\define\vicious{\circle}
\define\circle{\vicious}
```

а затем использовать в файле `\vicious`?

Вам следует определиться относительно того, где помещать ваши `\define`. Прежде всего, новую управляющую последовательность нужно определять до того, как вы ею воспользуетесь. Так что один из способов состоит в том, чтобы собрать все `\define` в начале статьи, скажем, сразу после строки `\document`. Таким образом, вы будете знать, куда надо посмотреть, чтобы вспомнить, что же было определено посредством `\define`. Но иногда вы можете осознать, что вам надо что-то определить, только по мере продвижения набора, и вместо того чтобы перескакивать к началу, можно вставлять `\define` тогда, когда назреет необходимость. В таких случаях `\define` вводится непосредственно перед тем абзацем, где это потребуется. Можно даже вводить `\define` прямо там, где это нужно, посередине абзаца, но если вы так поступите, будьте особенно внимательны к пробелам. Если набрать что-то вроде

```
If we have \define\ab{\alpha^2+\beta^2} the
condition $\ab+1>0$,
then $\frac{\ab}{\gamma}>0$.
```

то получится

If we have the condition $\alpha^2 + \beta^2 + 1 > 0$, then $\frac{\alpha^2 + \beta^2}{\gamma} > 0$.

с двумя пробелами после 'have': один — от пробела после 'have' и один — от пробела после `\define\ab{...}`. Так что один из пробелов следовало бы удалить, делая текст распечатки несколько более сжатым. Даже если бы вы набрали

```
If we have \define\ab{\alpha^2+\beta^2}
the condition $\ab+1>0$, then $\frac{\ab}{\gamma}>0$.
```

то получилось бы два пробела, так как `(carriage-return)` в конце первой строки воспринимается как пробел. Я предпочитаю собирать все мои `\define` на отдельной строке с % в конце ее, чтобы отменить действие `(carriage-return)`:

```
If we have
\define\ab{\alpha^2+\beta^2}%
the condition $\ab+1>0$, then $\frac{\ab}{\gamma}>0$.
```

Кроме того что надо следить за правильной расстановкой пробелов, также надо не упустить из виду, что `\define` внутри группы будет забыто, когда группа завершится. Если вы наберете

```
{\bf If we have
\define\ab{\alpha^2+\beta^2}%
the condition  $\ab+1>0$ , then  $\frac{\ab}{\gamma}>0$ .}
```

то `\ab` после закрывающей скобки `}` будет не определено. Вы, разумеется, не склонны вставлять `\define` внутри группы, подобной этой, но, возможно, вставьте `\define` внутри математической формулы, тем более, что тут дополнительные пробелы игнорируются. Но важно знать, что внутри математической формулы `\define` также будет забыто, как только формула закончится. На практике такое свойство `\define` обычно очень удобно — это означает, что можно ввести управляющую последовательность временно посредством `\define` только для данной математической формулы, а для другой формулы использовать ту же самую управляющую последовательность для чего-нибудь совсем иного.

Мы перечислили несколько предосторожностей, которые нужно соблюдать при пользовании `\define`, но имеется и более практический вопрос, когда новое `\define` действительно выгодно — вам же не захочется использовать все эти мощные новшества, для того чтобы завалить себя избытком бесполезных приспособлений.

Прежде всего, имеется немало специальных `\define` и `\redefine`, которые вы хотели бы использовать постоянно, вроде

```
\define\a{\alpha}
\predefine\barunder{\b}
\redefine\b{\beta}
\predefine\cedilla{c}
\redefine\c{\gamma}
\predefine\dotunder{d}
\redefine\d{\delta}
\define\e{\epsilon}
```

и остальные сокращенные названия греческих букв, или общеупотребительных математических символов, которыми вы часто пользуетесь. Их можно хранить в особом файле, скажем, `defs.tex`, и использовать ваш текстовый редактор для вставки содержимого этого файла в любой другой файл, с которым вам придется работать. Мой файл стандартных определений содержит

```
\define\left{
\define\right}
define\left[
\define\right]}
```

Так что я могу набирать формулы вроде

$$\left(\frac{1+\sqrt{5}}{2}\right)^n$$

просто как

$$\frac{(1+\sqrt{5})^n - (1-\sqrt{5})^n}{\sqrt{5}}$$

Кроме постоянно используемых `\define`, зачастую бывают нужны несколько специальных `\define`, которые могут сэкономить массу усилий на набор того или иного конкретного файла. Не надо слишком усердствовать в создании новых `\define` для каждой статьи — кроме всего прочего, это отнимает время на набор самих `\define` для отдельного символа, а если у вас слишком много новых управляющих последовательностей, требующих вашего внимания, вы просто снизите скорость набора, пытаясь вспомнить то или иное название. Но имеет смысл просмотреть всю статью, прежде чем начать ее набирать, чтобы понять, есть ли в ней формулы, встречающиеся десятки раз, или какие-нибудь особо сложные формулы, повторяющиеся раз 5–6. В частности, если автор вводит новые операторы типа `Tor` и `Res` (ср. со с. 95), то вам, наверное, пригодится

```
\define\Tor{\operatorname{Tor}}
\define\Res{\operatorname{withlimits{Res}}}
```

Наконец, всегда может встретиться какая-нибудь формула-монстр, ради которой не жаль ввести несколько локальных определений.

• Упражнение 18.11:

Как-то автору нужно было набрать формулу

$$\begin{aligned} a_n &= a_{n-1} + a_{n-2} \\ &= \frac{\left(\frac{1+\sqrt{5}}{2}\right)^{n-2} - \left(\frac{1-\sqrt{5}}{2}\right)^{n-2} + \left(\frac{1+\sqrt{5}}{2}\right)^{n-1} - \left(\frac{1-\sqrt{5}}{2}\right)^{n-1}}{\sqrt{5}} \\ &= \frac{\left(\frac{1+\sqrt{5}}{2}\right)^{n-2} \left(1 + \frac{1+\sqrt{5}}{2}\right) - \left(\frac{1-\sqrt{5}}{2}\right)^{n-2} \left(1 + \frac{1-\sqrt{5}}{2}\right)}{\sqrt{5}} \\ &= \frac{\left(\frac{1+\sqrt{5}}{2}\right)^{n-2} \left(\frac{1+\sqrt{5}}{2}\right)^2 - \left(\frac{1-\sqrt{5}}{2}\right)^{n-2} \left(\frac{1-\sqrt{5}}{2}\right)^2}{\sqrt{5}} \\ &= \frac{\left(\frac{1+\sqrt{5}}{2}\right)^n - \left(\frac{1-\sqrt{5}}{2}\right)^n}{\sqrt{5}}. \end{aligned}$$

— настоящее минное поле из потенциальных ошибок набора, особенно из-за всех этих `\frac` внутри `\frac`. Какие можно было использовать средства для того, чтобы избежать ошибок (при том, что управляющие последовательности `\(` и `\)` уже были доступны)?

Если вы опытный наборщик математических текстов, вы, по-видимому, задавались вопросом, как получать новые управляющие последовательности с аргументами, поскольку в статьях часто встречаются формулы почти идентичные, отличающиеся лишь несколькими подформулами или их расположением. Например, если вы набираете книгу по математическому анализу, то встретите мириады “производных” вроде

$$\frac{dy}{dx}, \quad \frac{dz}{dx}, \quad \frac{dw}{dx}$$

Хотя можно посредством `\define` определить несколько новых управляющих последовательностей вроде `\derivy`, `\derivz` и `\derivw`, было бы приятнее иметь единственную управляющую последовательность `\deriv`, чтобы `$$$ \deriv y $$$` давало $\frac{dy}{dx}$, а `$$$ \deriv z $$$` давало $\frac{dz}{dx}$, и т. д. Для получения такой управляющей последовательности `\deriv` с одним аргументом наберите

```
\define\deriv#1{\dfrac{d#1}{dx}}
```

Здесь `#1` обозначает аргумент, так что `\deriv f` заменяется на `\dfrac{df}{dx}`, и т. д.

• **Упражнение 18.12:**

Как бы вы набрали формулы

$$\frac{dx^2}{dx}$$

и

$$\frac{d(x^2 + x^3)}{dx}$$

используя `\deriv`?

• **Упражнение 18.13:**

Предположим, в рукописи идет речь о всевозможных ‘векторах’: (x_1, \dots, x_n) , (y_1, \dots, y_n) , и т. п. Какую аббревиатуру удобно для них ввести?

• **Упражнение 18.14:**

А как бы вы тогда получали $(\alpha_1, \dots, \alpha_n)$ и (x'_1, \dots, x'_n) ?

• **Упражнение 18.15:**

И, наконец, как быть с формулами (x'_1, \dots, x'_n) и (x_1', \dots, x_n') ?

• **Упражнение 18.16:**

Определите управляющую последовательность с одним аргументом `\power`, такую, что `$$$ \power x $$$` порождает 2^x , а `$$$ \power \alpha $$$` порождает 2^α , и т. д.

• **Упражнение 18.17:**

Как использовать управляющую последовательность с аргументом, чтобы избежать набора большей части `\frac` в выключной формуле из упр. 18.11?

Если мы продвинулись от анализа бесконечно малых к уравнениям математической физики, то будем непрерывно наткаться на “частные производные” вроде

$$\frac{\partial f}{\partial x}, \quad \frac{\partial f}{\partial y}, \quad \frac{\partial g}{\partial x}, \quad \frac{\partial h}{\partial z}$$

набирать которые нет никакого удовольствия. Нам бы захотелось ввести управляющую последовательность `\pd` с двумя аргументами, с тем, чтобы эти формулы набирались как `$$\pd fx$$`, `$$\pd fy$$`, `$$\pd gx$$`, `$$\pd hz$$`, и т. д. Как и следовало ожидать, это можно сделать при помощи

```
\define\pd#1#2{\dfrac{\partial#1}{\partial#2}}
```

(Обратите внимание, что пробел перед # в `\partial#1` и `\partial#2` не нужен, несмотря на то, что вместо #1 и #2 можно подставить буквы; как только `TeX` прочитает `\define`, он воспримет `\partial` и #1 как постоянно разделенные.)

• **Упражнение 18.18:**

Некоторые авторы используют x_1, x_2, \dots вместо x, y, \dots в выражениях типа $\frac{\partial f}{\partial x_1}, \frac{\partial g}{\partial x_2}$ и т. д. Как бы вы набрали эти формулы при помощи определенной выше управляющей последовательности `\pd`? Как модифицировать это определение, чтобы их было легче набирать?

• **Упражнение 18.19:**

Предположим, что в рукописи очень много формул вида $\sqrt{a} + \sqrt{b}, \sqrt{a} + \sqrt{z}, \sqrt{c} + \sqrt{d}$, и т. д., и вам нужна управляющая последовательность `\sqrts`, чтобы иметь возможность набирать их в виде `$$\sqrts ab$$`, `$$\sqrts az$$`, `$$\sqrts cd$$`, и т. д. Объясните, что неправильно в следующем определении:

```
\define\sqrts#1#2{\sqrt#1+\sqrt#2}
```

• **Упражнение 18.20:**

По-видимому, для получения векторов типа $(x_1, \dots, x_n), (y_1, \dots, y_n)$, и т. д. недостаточно управляющей последовательности с одним аргументом. Вам, вероятно, понадобятся также $(x_1, \dots, x_m), (y_1, \dots, y_{n+1})$ и т. п. Объясните, как определить `\vector` так, чтобы мы могли набирать это как `$$\vector xm$$` и `$$\vector y{n+1}$$`.

• **Упражнение 18.21:**

Вы набрали 50 страниц рукописи, в которой автор употребил пару сотен формул вида $R_i^j k_l$. А теперь О вдруг решает, что на самом деле обозначение $R_i^j k_l$ было бы лучше. С некоторым трепетом О обращается к вам с просьбой сделать эти изменения, возможно, при помощи того вашего небольшого хитроумного текстового редактора. На самом деле редактор здесь не поможет, но вы все же снисходительно усмехаетесь. Почему?

• **Упражнение 18.22:**

Вы одновременно набираете несколько рукописей разных авторов. Одни из них используют обозначения типа x_i^2, y_j^3, \dots , тогда как другие предпочитают x_i^2, y_j^3, \dots . Как можно было бы использовать одну управляющую последовательность, чтобы упростить себе жизнь?

Поскольку при определении управляющей последовательности требовались некоторые предосторожности, вас не должно слишком удивить, что для управляющей последовательности с аргументами понадобится еще пара предосторожностей.

Прежде всего, если вы ввели `\define\cs`, где `\cs` есть управляющая последовательность с аргументами, то после `\define\cs` ‘аргументы’ `#1`, `#2`, ... должны появляться именно в таком порядке, и вы можете пойти только до `#9`. Если порядок их появления будет нарушен, `TeX` выдаст сообщение об ошибке.

• **Упражнение 18.23:**

В упр. 18.20 мы так определили `\vector`, что `$(\vector xn)$` дает (x_1, \dots, x_n) и т. д. Но, может быть, вам это не нравится, и вы предпочитаете набирать `$(\vector px)$`, с ‘`n`’ на первом месте, а ‘`x`’ на втором. Как это организовать?

Когда задается `\define` для управляющей последовательности с аргументом, очень важно соблюдать следующую предосторожность, хотя это и покажется несколько странным. Если мы дали определение

```
\define\pd#1#2{\dfrac{\partial #1}{\partial #2}}
```

то можем пользоваться `$(\pd fx)$` для набора $\frac{\partial f}{\partial x}$ и можем также набирать `$(\pd f x)$`, поскольку `TeX` всегда игнорирует пробелы, когда ищет аргументы управляющей последовательности. Но вы **не должны** оставлять пробелы после `#1` и `#2`, когда определяете `\pd` посредством `\define`! Если бы вы набрали

```
\define\pd#1 #2 {\dfrac{\partial #1}{\partial #2}}
```

то `TeX` бы подумал, что вам *необходимы* пробелы после первого и второго аргументов. Так что если вы наберете

```
\pd fx =3 +x
```

`TeX` не станет рассматривать `f` и `x` как первый и второй аргументы. Вместо этого он предположит, что `fx` — это первый аргумент, поскольку `fx` — самая короткая вещь, после которой стоит пробел. Точно так же `TeX` подумает, что второй аргумент это `=3`. Получим воистину странный результат!

Так что если вы хотите избежать аномалий при определении управляющих последовательностей с аргументами, убедитесь, что вы не вставили никаких дополнительных пробелов после `#1`, `#2`, `#3`, ... , и у вас не возникнет никаких проблем. Если вы находите это несколько таинственным и хотите углубиться в изучение тонкостей, то объяснение подобному явлению можете найти во вспомогательном материале в конце главы.

⚠ Так как `TeX` игнорирует пробелы, когда ищет аргументы управляющей последовательности, запрещение оставлять пробелы в определении управляющей последовательности может показаться несообразным. Но на самом деле это частный случай весьма общего механизма `TeX`'а, который позволяет определять управляющие последовательности, ведущие себя особым образом.

Рассмотрим, например, снова определение

```
\define\pd#1#2{\dfrac{\partial#1}{\partial#2}}
```

и предположим, что нам нужны символы вроде $\frac{\partial x^2}{\partial x}$, $\frac{\partial f(x,y)}{\partial y}$ и т. д. Тогда понадобятся фигурные скобки:

```
$$\pd {x^2}x$$
$$\pd {f(x,y)}y$$
```

Возможно, вам удобнее набирать что-нибудь вроде

```
$$ \pd x^2.x $$
$$ \pd f(x,y).y $$
```

с точкой, отделяющей аргументы. Это можно устроить так:

```
\define\pd#1.#2{\dfrac{\partial #1}{\partial#2}}
```

Такой `\define` говорит Т_ЭX'у, что . должна следовать за аргументом #1. (Если вы после аргумента оставляете пробел, Т_ЭX так и считает, что за этим аргументом должен следовать пробел.)

• **Упражнение 18.24:**

Как определить управляющую последовательность `\vector`, так, чтобы при наборе `$$\vector x,n.$` получать (x_1, \dots, x_n) , а при `$$\vector y,m+1.$` получать (y_1, \dots, y_{m+1}) и т. д.

• **Упражнение 18.25:**

Предположим, вы определили

```
\define\powers#1,#2.{2^{#1}+3^{#2}}
```

Как бы вы набрали $2^3 + 3^4$ и $2^{10.03} + 3^{1.034}$ и, наконец, $2^{1.034} + 3^{10.03}$?

Вот другой пример, где можно с пользой применить это особое свойство процесса `\define`. Ранее мы рассмотрели определения

```
\define\deRham{de~Rham}
\define\Cartan{\'E.~Cartan}
```

и обратили внимание на то, что следует быть очень внимательным и набирать `_`, когда бывает нужен пробел после имен. Ну а теперь покажем, как избавиться от такого развлечения. Предположим, что вместо этого мы ввели определения

```
\define\deRham/{de~Rham}
\define\Cartan/{\'E.~Cartan}
```

Теперь Т_ЭX ожидает, что после `\deRham` и `\Cartan` всегда будет /, так что всегда набирайте `\deRham/` и `\Cartan/`. Но эти `\.../` отнимают не так уж много времени — во всяком случае смотрятся довольно неплохо — и нет нужды беспокоиться по поводу того, что Т_ЭX проглотит дополнительные пробелы: если вы оставите пробел после /, то и Т_ЭX здесь оставит пробел.

• **Упражнение 18.26:**

Снова выполните упр. 18.4, полагая, что `\deRham` и `\Cartan` определены теперь по-новому.

• **Упражнение 18.27:**

Здесь вопрос позаковыристей. Положим, вы сочли `\AmSTeX` не лучшим именем для фирменного знака $\mathcal{A}\mathcal{M}\mathcal{S}\text{-}\mathcal{T}\mathcal{E}\mathcal{X}$, и вам кажется более подходящим `\AmS-TeX`. По этому поводу вы определяете

```
\define\AmS-TeX{\AmSTeX}
```

Что происходит?

• **Упражнение 18.28:**

Что происходит, когда вы дадите такие определения?

```
\define \ab {\alpha^2+\beta^2}
\define \1  {\alpha^2+\beta^2}
```

• **Упражнение 18.29:**

Что произойдет, если отбросить фигурные скобки в `\define\alpha{\alpha}`, т. е. задать в вашем файле

```
\define\alpha
```

Глава 19. ТѢХнические приемы

Из предыдущих глав мы узнали, как готовить практически любые стандартные математические блюда. В этой главе большой выбор пикантных приправ — несколько тем, которых мы еще не коснулись, и всевозможные маленькие хитрости, которые нам бывают нужны для получения специальных эффектов и улучшения внешнего вида формул. Эта информация разбита на разделы и упорядочена по алфавиту. Воспользовавшись указателем, вы довольно просто сможете найти любой специальный ТѢХнический прием, какой пожелаете. Но, по-видимому, имеет смысл сначала бегло просмотреть эту главу, чтобы понять, какого рода вещи в ней содержатся, а затем приступить к серьезному изучению. Чтобы усвоить всю эту информацию, потребуется изрядное количество времени, но такое заострение внимания на деталях позволит вам набирать формулы, которые могли бы сделать честь типографскому наборщику экстра-класса.

АБЗАЦЫ

Посредством команды `\flushpar` можно получить абзац со строками, сдвинутыми к левому краю. Разумеется, если у вас несколько таких абзацев подряд, вам захочется поместить между ними что-то вроде `\smallpagebreak` (см. **вертикальные пробелы**). Но если вы хотите весь материал представить так, то вам придется слишком часто обращаться к `\smallpagebreak\flushpar`; вместо этого лучше создать новый стиль, который автоматически задаст такой формат.

Текстовые абзацы внутри выключных формул — это совсем иное дело. Мы уже знаем, что внутри формул текст получается посредством `\text`, но это единая, неразрывная строка текста. Но иногда вы можете сказать больше:

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

Для подобных ситуаций $\mathcal{A}\mathcal{M}\mathcal{S}$ -ТѢХ предоставляет средство `\foldedtext`. Если набрать

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

то текст ‘...’ оформляется в виде абзаца (за исключением того, что первая строка не имеет втяжки). Формат определяет ширину такого `\foldedtext`, но вы можете указать любой другой `{dimen}` (см. **единицы измерения**), набрав

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

[автоматическое форматирование, отмена

Таким образом, указанная выше выключная формула была набрана как

```

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

```

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.

Так же, как и в случае `\intertext` (см. **выравниваемые формулы**), в конструкциях `\foldedtext` нельзя пользоваться `\par` или пустой строкой, но можно вместо этого прибегнуть к помощи `\endgraf`.

АВТОМАТИЧЕСКОЕ ФОРМАТИРОВАНИЕ, ОТМЕНА

(См. также нумерация страниц, размер страницы и колонтитулы.)

Стоит только кому-нибудь создать продуманную систему, обеспечивающую вас множеством способов оформления, как обязательно найдется кто-то другой, кому захочется избавиться от некоторых из них. Например, для `\proclaim` выбран шрифт `\bf` и автоматическая установка нужного знака пунктуации в конце. Но вы можете пожелать чего-нибудь иного:

MAIN THEOREM!

Довольно просто сделать так, чтобы $\text{A}_M\text{S-}\text{T}_E\text{X}$ переключился на капитель — для этого достаточно набрать `\proclaim{\smc ...}`. Но знак препинания — это дополнительный элемент оформления, так что вы должны дать $\text{A}_M\text{S-}\text{T}_E\text{X}$ у указание удалить его, набрав `\nofrills` сразу после `\proclaim`:

```
\proclaim\nofrills{\smc Main Theorem!}
```

Аналогично, вы можете набрать

```
\demo\nofrills{Proof (?)}
```

чтобы получить

Proof (?)

Но следует быть осторожным, когда вы используете `\nofrills`, потому что пробел — это один из оформительских элементов, который вы воспринимаете как само собой разумеющееся. Наберете ли вы

```
\proclaim{Theorem} Here's a theorem.
```

акценты в математическом режиме]

или

```
\proclaim{Theorem}Here's a theorem.
```

стиль `amsppt` напечатает сначала ‘Theorem.’, а затем — ‘Here’s a theorem’ в наклонном начертании. Пробел после ‘Theorem.’ вставлен для свободы, и любые пробелы после } `AMS-TEX` игнорирует, так что нет необходимости беспокоиться о том, начинать ли формулировку теоремы сразу после закрывающей фигурной скобки или отступить. Но если вы используете `\nofrills`, этот пробел не появляется, так что на самом деле вам придется набирать что-то вроде

```
\proclaim\nofrills{Main Theorem! } Here's a theorem.
```

А еще лучше набирать

```
\proclaim\nofrills{Main Theorem!\usualspace}
Here's a theorem.
```

Это дает тот самый пробел, который оставляется после знака препинания в `\proclaim`.

Можно также помещать `\nofrills` после `\subheading`, чтобы избавиться от знака препинания, выбираемого автоматически; после `\abstract`, чтобы избавиться от слов “Abstract.”, “Summary.”, и т. д., автоматически проставляемых в этом стиле (это вам, несомненно, подойдет, если вы используете стиль `amsppt` для статьи, набранной на ином языке); и даже после `\keywords` и `\subjclass`. Во всех этих случаях вы также можете использовать `\usualspace` для получения пробела после знака пунктуации.

См. также `\Refs` и о наборе библиографических ссылок в приложении С.

АВТОР

См. приложение А.

АКЦЕНТЫ В МАТЕМАТИЧЕСКОМ РЕЖИМЕ

Обычные акценты типа `\``, `\^` и т. п., употребляемые в тексте, в математическом режиме не допускаются. Вместо них используются такие акценты:

<code> \$\hat a\$</code>	\hat{a}
<code> \$\check a\$</code>	\check{a}
<code> \$\tilde a\$</code>	\tilde{a}
<code> \$\acute a\$</code>	\acute{a}
<code> \$\grave a\$</code>	\grave{a}
<code> \$\dot a\$</code>	\dot{a}
<code> \$\ddot a\$</code>	\ddot{a}
<code> \$\ddd a\$</code>	\ddd{a}
<code> \$\dddd a\$</code>	\dddd{a}
<code> \$\breve a\$</code>	\breve{a}
<code> \$\bar a\$</code>	\bar{a}
<code> \$\vec a\$</code>	\vec{a}

[акценты в математическом режиме

Очень неплохая мысль — определить особые управляющие последовательности для тех снабженных акцентами букв, которые встречаются особенно часто. Если \hat{A} , \hat{c} , \check{s} , \tilde{x} и \bar{z} встретились в статье раз 6 и даже больше, вы можете ввести

```
\define\Ahat{\hat A}
\define\chat{\hat c}
\define\scheck{\check s}
\define\xtilde{\tilde x}
\define\zbar{\bar z}
```

в начале своего файла. Это не только сократит набор, но и облегчит чтение входного файла. Обратите внимание, что мы использовали дополнительные фигурные скобки, так что теперь имеем возможность набирать выражения вроде $\$2^{\hat{A}}\$$ (ср. со с. 69 и упр. 18.2).

Как и в текстовом режиме, когда акцент располагается над i или j , следует использовать форму “без точек” i и j . Это достигается набором \imath и \jmath в математическом режиме. Например, $\hat{\imath}$ дает \hat{i} .

TeX не станет возражать, если вы расположите математический акцент над всей формулой целиком, но обычно в этом нет большого смысла, потому что акцент расположится в самой середине над формулой. Например, $\hat{I+M}$ даст $I + M$. Когда математикам бывает нужно показать, что акцент относится ко всей формуле, они обычно прибегают к одному из следующих средств.

Прежде всего, у некоторых из акцентов есть более широкие варианты. Например, \widehat{x} и \widetilde{x} :

\widehat{x} , \widetilde{x}	\hat{x} , \tilde{x}
\widehat{xy} , \widetilde{xy}	\overhat{xy} , \overtilde{xy}
\widehat{xyz} , \widetilde{xyz}	\overhat{xyz} , \overtilde{xyz}
\widehat{xyzu} , \widetilde{xyzu}	\overhat{xyzu} , \overtilde{xyzu}
\widehat{xyzuv} , \widetilde{xyzuv}	\overhat{xyzuv} , \overtilde{xyzuv}

Эти акценты взяты из семейства `msbm`. Если `msbm` загружен, то \widehat{x} и \widetilde{x} автоматически будут выбирать эти более широкие варианты, когда потребуется. В противном случае наибольший из доступных размеров этих акцентов представлен в третьей строке. Если вы используете стиль `amsrpt`, шрифт `msbm` загружается автоматически. Если нет, то см. приложение G относительно инструкций, как его загрузить.

Другая возможность состоит в использовании обозначения типа $(I + M)^{\wedge}$ с акцентом \wedge в виде верхнего индекса. Чтобы получить “крышечку” в таком виде, нельзя набирать $\hat{\wedge}$ или $\sp{\hat}$, так как $\hat{\wedge}$ — это не символ, а команда для получения акцента над чем-либо. Но у `AMS-TeX`’а имеется $\sp{\hat}$, которая работает так же, как вы могли бы ожидать от $\sp{\hat}$, а также есть $\sp{\check}$, \dots , $\sp{\vec}$.

В некотором смысле у акцента \vec также имеется более широкий вариант, так как всегда можно воспользоваться \overrightarrow . Над отдельной буквой \vec обычно смотрится несколько лучше, чем \overrightarrow : ср. \vec{z} (\vec{z}) и \overrightarrow{z} (\overrightarrow{z}).

акценты в математическом режиме]

Аналогично, вместо акцента `\bar` над длинной формулой можно использовать `\overline`. Опять же `\bar` и `\overline` над отдельной литерой внешне различаются. Например, `\bar z` дает \bar{z} , тогда как `\overline z` дает \overline{z} . Акцент `\bar` — это специальный символ, аналогичный акценту `\B` в тексте. С другой стороны, `\overline` — прямая определенной толщины (обычно иной, чем `\bar`), и она полностью закрывает формулу. В формуле вроде \overline{z} (`\overline z`) она может показаться слишком длинной, а в `\overline{\phi}` и `\overline{A}` еще менее приемлемой: $\overline{\phi}$ и \overline{A} . А акцент в \bar{M} (`\bar M`), напротив, покажется слишком маленьким, так что здесь следует предпочесть \overline{M} (`\overline M`).

Когда вы начинаете работу над рукописью, вы можете не знать, что лучше будет поставить над конкретной буквой: `\bar` или `\overline`, поэтому иногда придется вернуться и заменить `\bar M` на `\overline M` или *vice versa*. Разумеется, если в рукописи много таких \bar{M} , то следует ввести `\define` для управляющей последовательности `\Mbar` в качестве аббревиатуры, и тогда надо будет только изменить определение.

На самом деле Трех можно проинструктировать рисовать прямые любой длины и ширины и в любом месте, так что будет получаться вполне приемлемое \bar{M} . Но для этого потребуются настоящая Трехпертиза и ряд проб и ошибок. С такого рода проблемами можно расправиться довольно лихо: если вы находите, что оба обозначения `\bar M` и `\overline M` не годятся, то временно определите `\define\Mbar{\bar M}` и потом попросите кого-нибудь знающего сделать более сложное определение.

Приступим теперь к более занимательным вещам: к двойным акцентам. В журналах обычно отговаривают и даже запрещают использовать двойные акценты вроде $\hat{\hat{A}}$, которые при наборе в типографии приходится собирать из очень мелких кусочков. Если математик использует множество двойных акцентов вроде $\hat{\hat{A}}$, $\hat{\hat{A}}$, $\hat{\hat{A}}$, ..., то читатели Е статьи вероятно будут так же недовольны, как и наборщики, и Е придется подобрать более удачные обозначения. Но иногда двойные акценты необходимы, и математики, наверное, впадут в экстаз, узнав что компьютерный набор с этим легко справится.

Но все дело в том, что даже у Трех'a возникают затруднения с двойными акцентами. Когда вы набираете `\hat A`, то получаете \hat{A} с акцентом справа, над острием буквы A , но при наборе `\hat{\hat A}` получаете $\hat{\hat{A}}$ (эх!), с верхней крышечкой, отцентрированной относительно всей формулы \hat{A} . Поэтому у $\mathcal{M}\mathcal{S}$ -Трех'a имеется альтернатива `\Hat`, которая усложняет работу Трех'a, но все делает, как надо: если вы набираете `\Hat{\hat A}`, то получаете $\hat{\hat{A}}$. Аналогично обстоит дело с `\Check`, `\Tilde`, `\Acute`, `\Grave`, `\Dot`, `\Ddot`, `\Breve`, `\Bar` и `\Vec`; в любой разумной статье они бывают нужны крайне редко, и $\mathcal{M}\mathcal{S}$ -Трех даже не побеспокоился о том, чтобы снабдить вас `\Ddot` или `\Dddot`.

$\mathcal{M}\mathcal{S}$ -Трех предоставляет и `\hat` и `\Hat`, потому что `\Hat` на самом деле пожирает компьютерное время. Вы всегда должны использовать `\hat` в качестве единичного акцента, оставляя `\Hat` только для тех случаев, когда нужно заставить Трех потрудиться над правильным двойным акцентом. Насколько больше приходится Трех'у работать над `\Hat`, чем над `\hat`? Ну, если $\hat{\hat{A}}$ встречается на

[большие ограничители `big` и `bigg`

одной странице раз 6–7, это может удвоить время ее обработки! Если определить

```
\define\Ahathat{\Hat{\Hat A}}
```

то вы облегчите жизнь себе, но не Т_EX'у, так как ваши `\Ahathat` просто транслируются в `\Hat{\Hat A}`. Так что \mathcal{AMS} -Т_EX предоставляет специальное средство, помогающее в ситуациях, подобных этой. Следует набирать так:

```
\accentedsymbol\Ahathat{\Hat{\Hat A}}
```

В этом случае Т_EX сначала проделывает всю работу, заключенную в наборе `\Hat{\Hat A}`, но затем он запоминает необходимую информацию. Так что как только вы набираете `\Ahathat` в своем файле, он сразу вспоминает все, что нужно, не проделывая заново всю работу. Это как если бы типографский наборщик имел специальную литеру \hat{A} и всегда пользовался этим одним наборным материалом, вместо того чтобы кропотливо подсоединять друг к другу крошечные кусочки набора. Обладая таким `\Ahathat`, полученным посредством `\accentedsymbol`, вы можете использовать его сколь угодно часто, не заботясь о том, замедлится ли работа Т_EX'а.

Не следует слишком часто прибегать к помощи `\accentedsymbol`, так как Т_EX обладает ограниченным пространством памяти для хранения новых символов; если окажется, что посредством `\accentedsymbol` придется воспользоваться более 10 раз для получения новых символов для одной статьи, значит система обозначений автора никуда не годится. Имеется еще одна маленькая ловушка: созданный вами посредством `\accentedsymbol` новый символ `\Ahathat` действительно аналогичен отдельной типографской литере — он не меняет размера в индексах. Если вам нужны верхние индексы \hat{A} в большом количестве, создайте более мелкий символ при помощи

```
\accentedsymbol\smallAhathat{\ssize\Hat{\Hat A}}
```

(обратите внимание на дополнительные фигурные скобки) и теперь набирайте, например,

$$\text{\$}\Gamma_1^{-1}\text{\smallAhathat}\text{\$} \quad \Gamma_1^{\hat{A}}$$

См. также шрифты в математическом режиме.

БОЛЬШИЕ ОГРАНИЧИТЕЛИ `big` И `bigg`

Мы уже познакомились с `\bigl`, `\bigr` и `\biggl`, `\biggr`, позволяющими получать различные размеры ограничителей, но имеются также `\big` и `\bigg`, которые порождают ограничитель, выступающий в качестве обычной переменной. Эти средства преимущественно применяются к косой и к обратной косой (см. ограничители), как в следующем примере.

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

большие операторы]

Имеются также `\bigm` и `\biggm`, используемые в середине формул и дающие ограничитель, который выступает в качестве бинарного отношения типа $=$, и поэтому ТрХ вставляет пробелы вокруг него.

$$\begin{aligned} & \mathbb{\bigl}(x \in A(n) \mathbb{\bigm} | x \in B(n) \mathbb{\bigr)} & (x \in A(n) \mid x \in B(n)) \\ & \mathbb{\bigcup}_n X_n \mathbb{\bigm} \mathbb{\bigcap}_n Y_n & \bigcup_n X_n \parallel \bigcap_n Y_n \end{aligned}$$

Кроме того, для всех этих вещей существует два еще бóльших размера: `\Big` — на 50% выше `\big`, и `\Bigg` — в 2.5 раза выше `\big`. Но эти управляющие последовательности труднее набирать, да и используются они реже, поскольку формулы обычно бывают высотой в одну или две строки, а не в $1\frac{1}{2}$ или $2\frac{1}{2}$ строки.

БОЛЬШИЕ ОПЕРАТОРЫ

(См. также составные символы.)

Хотя стиль `amsrpt` обычно использует “пределы” для `\sum` и больших операторов отношения, это соглашение можно изменить, набрав `\NoLimitsOnSums`. (Если набрать `\LimitsOnSums`, вернемся к старому соглашению, но на самом деле не следует включать и выключать стили.) Аналогично обстоит дело с `\LimitsOnInts` и `\NoLimitsOnInts` для `\int`, `\oint`, `\iint` и т. п. Стилиевые соглашения могут также влиять на операторы типа `\max` и `\min`, получаемые при помощи `\operatornamewithlimits`: в некоторых стилях допускается отсутствие “пределов” во всех подобных операторах. Набирая `\NoLimitsOnNames`, вы устанавливаете такие соглашения, а `\LimitsOnNames` означает возврат к прежним соглашениям. Разумеется, оператор `\sin` и ему подобные, которые получаются посредством `\operatorname` никогда не имеют “пределов”.

Эти управляющие последовательности “глобальны” — они действуют на весь последующий текст, даже если были использованы в группе или между знаками $\$$. Использование их в журнальном стиле бессмысленно — они будут проигнорированы.

Дальнейший текст этой статьи касается некоторых тонкостей расстановки пределов у больших операторов. Вы можете, по-видимому, его опустить, представив редактору журнала самому позаботиться о таких подробностях.

Мы уже упоминали, что когда мы заключаем `\sum` с пределами в скобки или ставим другие ограничители, то лучше использовать размер `\bigg`, но в упр. 13.18 приводится пример, когда на самом деле имеет смысл применить `\left` и `\right`. Однако, набрав

$$\mathbb{\left(\sum_{i=1}^n p_{ix_i} \right) \mathbb{\right)}$$

можете получить нечто неожиданное, вроде

$$\left(\sum_{i=1}^n p_i x_i \right)$$

Это происходит потому, что ТрХ добавляет дополнительный “буферный” пробел над и под пределами, чтобы они не послужили помехой окружающим формулам.

[большие операторы]

Шрифт, которым набран этот учебник, имеет буфер 1pt, но в формуле выше был использован буфер 3pt. Если вы хотите быть полностью защищены от подобных ситуаций, можете срезать буферный пробел при помощи `\shave`: ввод

```
$$\left( \shave{\sum_{i=1}^n} p_{ix_i} \right)$$
```

даст

$$\left(\sum_{i=1}^n p_i x_i \right)$$

каким бы ни был размер буферного пробела.

Также стоит использовать `\shave` для больших операторов под знаком корня:

```
$$\sqrt{\shave{\sum_{i=1}^n} a_i}$$
```

Кроме `\shave` существуют `\topshave` и `\botshave` для срезания только верхнего или только нижнего буферного пробела. Например, вы можете набрать

```
$$\sqrt{ \botshave{\sum_{0<i<n} } a_i }$$
```

чтобы получилось

$$\sqrt{\sum_{0<i<n} a_i}$$

А если бы вы набрали

```
$$\frac{\dspace 1+\sum_{i=1}^N a_i}{\dspace 1+\sum_{j=1}^M b_j}$$
```

то получили бы

$$\frac{1 + \sum_{i=1}^N a_i}{1 + \sum_{j=1}^M b_j}$$

Если вы захотите заключить эту формулу в скобки, то лучше всего набрать

```
$$\left( \frac{\dspace 1+\topshave{\sum_{i=1}^N } a_i}{\dspace 1+\botshave{\sum_{j=1}^M } b_j} \right)$$
```

— мы хотим срезать пробел над N в числителе и пробел под $j=1$ в знаменателе, но буферные пробелы ниже $i=1$ и выше M трогать нельзя.

У \LaTeX имеется также управляющая последовательность `\buffer` для указания размера буфера, которую вы можете использовать для отшлифовки того или иного вывода.

вертикальные пробелы]

◆ Используемый стиль должен на самом деле определять, насколько велик буфер, но, набрав

```
\ChangeBuffer{(dimen)}
```

вы можете изменить размер буфера на какой-нибудь иной $\langle \text{dimen} \rangle$ (см. **единицы измерения**). Это изменение “глобальное” — оно действует повсюду, даже если было набрано внутри группы или между знаками \$. Однако существует управляющая последовательность `\ResetBuffer` для восстановления прежнего размера буфера. Вообще говоря, `\ChangeBuffer` следует использовать только для постоянных изменений в стиле `amsrpt`, но большинство журнальных стилей будет, по видимому, игнорировать эту управляющую последовательность.

ВВОД

См. файлы.

ВЕРТИКАЛЬНЫЕ ПРОБЕЛЫ

Как правило, перед `\proclaim` оставляется небольшой пробел, который естественным образом исчезает, если `\proclaim` окажется вверху страницы. Т_ЭХ обеспечивает возможность расстановки таких исчезающих в нужных ситуациях вертикальных пробелов любого размера, но в таком приеме нет необходимости, если вы пользуетесь готовым стилем, в котором уже заложены пробелы перед такими вещами как `\proclaim`. Тем не менее в стиле `amsrpt` имеются управляющие последовательности `\smallpagebreak`, `\medpagebreak` и `\bigpagebreak` для получения таких вертикальных пробелов:

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

Их следует использовать только между абзацами (в самом начале абзаца или в самом конце абзаца считается “между абзацами”). Перед `\flushpar` можно использовать `\smallpagebreak` или `\medpagebreak` (см. **абзацы**).

Для `\smallpagebreak`, `\medpagebreak` и `\bigpagebreak` использованы такие названия потому, что они не только оставляют некоторое свободное пространство, но и подсказывают Т_ЭХ’у, что разрыв страницы в этом месте соответственно возможен, уместен и весьма желателен. Разумеется, если в этом месте произойдет разрыв страницы, пробел исчезнет. Вы не можете добавлять такие пробелы, набирая две и более таких управляющих последовательностей одна под другой — сухим остатком будет тот, что больше.

Для выключных формул у АМ_С-Т_ЭХ’а значительно большее разнообразие вертикальных пробелов. Если многострочная выключная формула создается при помощи `\align`, и т. п., то строки уже и так располагаются чуть свободнее, чем в обычном тексте. Но если в формуле есть большие символы, то бывает полезно добавить пробел и от себя лично.

В любых математических конструкциях, использующих `\` для разбивки на строки, таких как `\align`, `\aligned`, `\gather`, `\Sb`, `\matrix` и т. п., можно набирать

```
\vspace{(dimen)}
```

[вставка рисунков]

сразу после `\`, чтобы вставить пробел размера $\langle \text{dimen} \rangle$ сразу после этой строки. Хотя можно указать любой $\langle \text{dimen} \rangle$, обычно его выражают в виде `1\jot` или `.5\jot` т. п., и тогда ваше улучшение даст нужный эффект в любом формате.

Иногда полезно добавить вертикальные пробелы между всеми строками выключной формулы. Например, выключная формула из упр. 18.11 смотрелась бы лучше, если бы после каждого `\` был вставлен `1\jot`. Но тут масса работы, поэтому AMS-TeX предоставляет вам также средство `\spreadlines`. Если набрать

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

то все строки в этой формуле раздвинутся на $\langle \text{dimen} \rangle$. AMS-TeX позволяет пользоваться `\spreadlines` только в выключных формулах (между знаками `$$`), и его действие роаспространяется только на эту формулу.

`\spreadlines` дает результат в `\align` или `\aligned` или `\gather`, и т. п., но он не даст пробелов в таких (довольно специфических) конструкциях, как `\Sb` или `\Sp`, и в любых видах `\matrix`, поскольку `\matrix` обычно мыслится как подформула, пробелами в которой не стоит манипулировать. Разумеется, есть возможность изменять пробелы в матрице, если после каждого `\` вставить `\vspace{\langle \text{dimen} \rangle}`, но для этой цели у AMS-TeX 'а имеется управляющая последовательность `\spreadmatrixlines`. Набрав

```
\spreadmatrixlines{\langle \text{dimen} \rangle}
```

в выключной формуле, вы получите во всех `\matrix` этой формулы дополнительное пространство размера $\langle \text{dimen} \rangle$ после каждой строки (если нужно раздвинуть строки только в одной матрице, наберите

```
{\spreadmatrixlines{\langle \text{dimen} \rangle}\matrix
...\endmatrix}
```

только для этой `\matrix`).

Для `\cases` все происходит аналогично тому, как для `\matrix`: `\spreadlines` не дает никакого результата, но можно поместить `\vspace{\langle \text{dimen} \rangle}` после любого `\`, а `\spreadmatrixlines` увеличит пробелы после всех строк `\cases` точно так же, как и в случае `\matrix`.

ВСТАВКА РИСУНКОВ

См. вставки с подписями.

ВСТАВКИ С ПОДПИСЯМИ

Рисунки, таблицы и некоторые иные фрагменты текста или данных зачастую обрабатываются как вставки. Такие вставки иногда набираются как часть документа с использованием TeX 'а или AMS-TeX 'а, но чаще их подготавливают

вставки с подписями]

отдельно от основного документа и потом вклеивают в заранее оставленное в документе место. Эти вставки обычно имеют подписи либо над данными (в таблицах), либо под данными (в рисунках).

В таких вставках должны быть указания о размещении их либо сверху, либо в середине страницы, что достигается соответственно командами `\topinsert` либо `\midinsert`.

Подписи могут располагаться либо сверху, либо внизу вставки, что достигается при помощи соответственно `\topcaption` или `\botcaption`.

Типичная структура описания вставки с подписью сверху выглядит так:

```
\topinsert или \midinsert
\topcaption{( номер вставки)}
{ текст подписи}\endcaption
\vspace{(dimen)} или { иной код для тела вставки}
\endinsert
```

Для вставки с подписью внизу — аналогично:

```
\topinsert или \midinsert
\vspace{(dimen)} или { иной код для тела вставки}
\botcaption{( номер вставки)}
{ текст подписи}\endcaption
\endinsert
```

Если вы хотите оставить пустое место для чего-нибудь, что будет вклеено позже, то `\vspace{(dimen)}` должно указывать точную высоту материала, который вы собираетесь впоследствии вставить. Все дополнительные пробелы вокруг объекта и подписи определяются стилем, в котором подготовлен документ, и они будут выдержаны автоматически. (номер вставки) — это нечто вроде “Рис. 1” или “Таблица 2а”. Не ставьте точку в конце — она проставляется автоматически. (номер вставки) обычно изображается капителью.

{текст подписи} — это обычно любой описательный текст или ничего, в зависимости от того, что требуется. В стиле `amsrpt` этот текст будет представлен прямым шрифтом. Если даже никакого текста нет, метка `\endcaption` обязательна.

Если вы решили включить в ввод Т_ЭХ-код для рисунков, таблиц или иных вставок, то команду `\vspace{(dimen)}` включать не следует. Вместо этого наберите Т_ЭХ-код для вставки вслед за `\topcaption`, или перед `\botcaption`, судя по тому, что требуется. Размер будет вычислен автоматически, и соответствующая подпись будет вставлена в соответствующее место над или под объектом.

Ширина подписи определяется стилем документа, но она может быть задана и посредством `\captionwidth{(dimen)}` сразу перед подписью. Ввод

```
\midinsert
\vspace{.5in}
\captionwidth{17pc}
\botcaption{Рис. 20.1} Пример вставки с подписью внизу
\endcaption
\endinsert
```

Рис. 20.1. Пример вставки с подписью
внизу

ВЫРАВНИВАЕМЫЕ ФОРМУЛЫ

Межстрочные интервалы в `\align` можно менять при помощи управляющих последовательностей `\vspace` и `\spreadlines` (см. многострочные формулы).

Внутри выключных формул, содержащих `\align... \endalign`, не допускается переход на другую страницу. Но набрав `\allowdisplaybreak` после какого-нибудь `\\`, вы получите возможность разорвать формулу после этой строки и перенести ее на другую страницу. Можете также добиться разрыва выключной формулы после какой-нибудь строки, применив `\displaybreak`. Можно также набрать `\allowdisplaybreaks [sic]` перед `\align`, и это даст тот же эффект, как если бы вы набирали `\allowdisplaybreak` после каждой строки. Управляющую последовательность `\allowdisplaybreaks` разрешается помещать только между знаками `$$`, и ее действие распространяется лишь на ту выключную формулу, в которой она присутствует.

Можно также поместить

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

между двумя строками `\align`, чтобы на выходе получился текст, располагающийся между строками формулы и не нарушающий ее равенства. Если вы, например, набрали

```
We have
$$
\align X&= (-1)^{i+j-k/3+[\alpha,\beta]}Z_1
          +(-1)^{[\alpha/\beta-*[i+j/2,i+k/3]}Z_2\\
\intertext{which by properties (a)--(d) of $$*,
together with commutativity of the ring,}
&=\alpha Z_1+\beta Z_2,
\endalign
$$
which is the desired formula.
```


выравниваемые формулы]

We have

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

which by properties (a)–(d) of $*$, together with commutativity of the ring,

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

which is the desired formula.

с прекрасно выровненными знаками $=$. В `\intertext` нельзя вставлять `\par` или пустую строку, потому что в математическом режиме `\par` недопустим, но вместо этого есть управляющая последовательность `\endgraf`, которую можно использовать. Первая строка в `\intertext` всегда выровнена по левому краю (если только перед ней не стоит `\endgraf`).

За исключением редких случаев, использование `\intertext` считается дурным тоном. Даже если два разных `\align` тесно связаны между собой, нет такой традиции выравнивать их относительно друг друга; пользуйтесь `\intertext` лишь при особых обстоятельствах. (Если вы решили воспользоваться `\intertext` таким образом, комбинируя два или более `\align` в один длинный, будет, по-видимому, разумно до и после `\intertext` вставить `\allowdisplaybreak`. Но и тогда внутри `\intertext` переход на другую страницу недопустим.)

В конструкции `\aligned`, представляющей собой одно целое, нельзя использовать ни `\allowdisplaybreak[s]`, ни `\displaybreak` или `\intertext`. Но допускается `\vspace`; при этом `\spreadlines` действует на все `\aligned` внутри выключной формулы, где он употреблен.

Как показано на с. 99, различные `\aligned` вытягиваются вдоль их центральных строк. Имеются, однако, также `\topaligned` и `\botaligned`, которые вытягиваются вдоль верхних или нижних строк, если вам кажется, что в этом есть необходимость.

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

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

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

выравнивание по знаку $=$ проведено в трех местах. Это достигается при помощи

[выравниваемые формулы

средства \LaTeX 'а `\alignat`. Приведенная выше формула была набрана так:

```

$$
\alignat 3
V_i & =v_i-q_{iv_j}, & \quad X_i & =x_i-q_{ix_j}, \\
& \quad U_i & =u_i, \quad \text{\textit{for } $i \ne j$;} \tag 23 \\
V_j & =v_j, & \quad X_j & =x_j, \\
& \quad U_j & =u_j+\sum_{i \ne j} q_{iu_i}. \tag 24 \\
\endalignat
$$

```

Сразу после `\alignat` нужно указать количество пар формул, которые вы хотите выровнять.¹ 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, и между второй и третьей. Так что всего нужно 5 знаков `&` в строке. Обратите внимание также, что явно указывается пробел `\quad` между различными столбцами.

Чаще всего `\alignat` используется для формул вроде

$$\begin{array}{ll}
 x = y & \text{by (1)} \\
 x' = y' & \text{by (2)} \\
 x + x' = y + y' & \text{by Axiom 1.}
 \end{array}$$

Это было набрано так:

```

$$
\alignat2
x&=y &&\quad\text{\textit{by (1)}} \\
x'&=y' &&\quad\text{\textit{by (2)}} \\
x+x'&=y+y' &&\quad\text{\textit{by Axiom 1.}} \\
\endalignat
$$

```

Заметьте, что мы использовали `&&`, потому что хотели, чтобы условие в правой части рассматривалось как вторая формула пары и чтобы они были выровнены влево. Внутри `\alignat` можно использовать, как и в `\align`, вспомогательные управляющие последовательности `\vspace`, `\allowdisplaybreak`, `\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. Используя предыдущий

¹ Можете набрать `\alignat{10}`, если вам нужно 10 пар, но они, вероятно, не поместятся на странице.

выравниваемые формулы]

пример без пробелов между столбцами,

```

$$
\alignat 3
V_i &= v_i - q_{iv} v_j, & X_i &= x_i - q_{ix} x_j, \\
& U_i &= u_i, \text{\textfor $i \ne j$;} \tag 23 \\
V_j &= v_j, & X_j &= x_j, \\
& U_j &= u_j + \sum_{i \ne j} q_{iu} u_i. \tag 24
\endalignat
$$

```

получим

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

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

Средство `\xalignat` еще шире — первый и последний столбцы располагаются вдоль самых краев полосы набора. Здесь `\tag` не имеет смысла и в этой конструкции недопустим.

Существует также средство `\alignedat`, которое очень похоже на `\alignat`, за тем исключением, что здесь создается единый массив. 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 на строке не хватает места для размещения формулы и метки вместе в `\alignat`, метка не будет автоматически располагаться на отдельной строке. Метка может даже наложиться на формулу (!) без выдачи сообщения `Overfull box`. К счастью, такие конструкции встречаются достаточно редко, так что вам нужно просто убедиться, что все вышло хорошо. Если все-таки есть проблемы, нужно перенести метку на другую строку, введя пустую формулу.

На самом деле в случае слишком длинной формулы даже `\align` ведет себя несколько странно: черный прямоугольник появится не в конце слишком длинной строки, а в конце последней строки всего массива. Более того, будет выдано не одно, а два сообщения `Overfull \hbox`. Они, однако, почти идентичны, так что для беспокойства нет причин.

Наконец, имеется одна особая проблема выравнивания, для которой у $\mathcal{A}\mathcal{M}\mathcal{S}$ -Т_EX'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,

[горизонтальные пробелы

нельзя вкладывать `\align` или `\gather` внутрь другой конструкции, но \LaTeX 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.

ГОРИЗОНТАЛЬНЫЕ ПРОБЕЛЫ

Мы уже упоминали о том, что тонкие шпации `\`, следует использовать в формулах типа $f(x) dx$. Они должны также вставляться после восклицательного знака (который в математических выражениях имеет особый смысл), если за ним следует число, буква или левый ограничитель:

$$\begin{array}{ll}
 \$(2n)!/\bigl(n!, (n+1)!\bigr)$ & (2n)!/(n!(n+1)!) \\
 \$$\frac{52!}{13!13!26!}$$ & \frac{52!}{13!13!26!}
 \end{array}$$

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

$$\begin{array}{ll}
 \sqrt{2}, x & \sqrt{2}x \\
 \sqrt[O]{1/\sqrt{n}}, \bigl(r\bigr) & O(1/\sqrt{n})
 \end{array}$$

(Второй вид конструкции настолько общепринят, что я нахожу уместным определить

```
\define\ssqrt#1{\sqrt{#1}\,}
```

чтобы можно было просто менять `\sqrt` на `\ssqrt`, как только мне это понадобится.)

В иных случаях обычно можно полагаться на те пробелы, которые предоставляет \TeX , если только, посмотрев на распечатку, вы не решите, что следует для зрительного эффекта вставить дополнительный пробел, что бывает довольно редко. Когда вы дойдете до стадии улучшения внешнего вида распечатки, вам понадобится `\!` — это отрицательная тонкая шпация, которая удаляет ровно такой же пробел, который `\`, добавляет. Здесь приведены примеры того, как `\`, и `\!` позволяют несколько улучшить внешний вид:

горизонтальные фигурные скобки]

<code>\sqrt{\, \log x}</code>	$\sqrt{\log x}$
<code>[\, 0, 1]</code>	$[0, 1]$
<code>\log n\, (\log\log n)^2</code>	$\log n (\log \log n)^2$
<code>x^2!/2</code>	$x^2/2$
<code>n!/log n</code>	$n/\log n$
<code>\Gamma_2 + \Delta^2</code>	$\Gamma_2 + \Delta^2$
<code>R_i^j{}_k</code>	$R_i^j{}_k$
<code>\int_1^b \int_a^b</code>	$\int_1^b \int_a^b$

Тем не менее, обратите внимание, что `!` необязателен, если вы используете `\iint` и подобные ему операторы — эти управляющие последовательности сами вставляют необходимые `!`.

Синонимом для `\,` служит управляющая последовательность `\thinspace`, причем обе могут использоваться как в математическом режиме, так и вне его. Точно так же, `\negthinspace` это синоним для `!`, которые тоже используются в том и другом режимах. Гораздо реже применяются управляющие последовательности `\medspace`, `\negmedspace`, `\thickspace` и `\negthickspace`. Пробел `\medspace` обычно \TeX вставляет вокруг бинарных операторов, а `\thickspace` — вокруг бинарных отношений. Иногда все же `\thickspace` используется (см. точки), в качестве аббревиатуры применяется `\;`.

Если вы чрезвычайно придирчивы, вам может показаться, что даже `\,` и `!` недостаточно малы, чтобы удовлетворить вашему взыскательному вкусу. Поэтому $\mathcal{A}\mathcal{M}\mathcal{S}\text{-}\TeX$ предоставляет вам также `@`, для поистине крошечных горизонтальных пробелов ($\frac{1}{10}$ от `\,`), и `@!` для соответствующих отрицательных пробелов. Обратите внимание, что их использование в математическом режиме абсолютно отличается от их использования в тексте для получения правильного пробела после знака пунктуации, за которым следует прописная буква.

Для более длинных горизонтальных пробелов применяются `\quad` и `\qquad`; относительно других вариантов см. **фантомы**.

ГОРИЗОНТАЛЬНЫЕ ФИГУРНЫЕ СКОБКИ

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

<code>\overbrace{x+\dots+x}</code>	$\overbrace{x + \dots + x}$
<code>\underbrace{x+y+z}</code>	$\underbrace{x + y + z}$

Можно поместить какие-либо еще формулы над `\overbrace` или под `\underbrace`, если их набрать просто как верхний и нижний индексы, как если бы вы имели дело с большими операторами:

<code>\overbrace{x+\dots+x}^{\text{\text{\\$k\\$ times}}}</code>	$\overbrace{x + \dots + x}^{k \text{ times}}$
<code>\underbrace{x+y+z}_{\, \, 0}</code>	$\underbrace{x + y + z}_{> 0}$

[греческие буквы]

(Во втором примере была вставлена тонкая шпация, поскольку обычно `TeX` в `\ssize` не оставляет пробелов вокруг бинарных операторов.)

См. также составные символы.

• **Упражнение 19.1:**

Наберите следующее:

$$\frac{b^{p+1} - a^{p+1}}{\underbrace{1 + 1 + \dots + 1}_{p \text{ times}}} = \frac{b^{p+1} - a^{p+1}}{p + 1}.$$

• **Упражнение 19.2:**

Как бы вы набрали такую выключную формулу:

$$\underbrace{f\left(\frac{1}{n}\right) + \dots + f\left(\frac{1}{n}\right)}_{n \text{ times}} = f\left(\underbrace{\frac{1}{n} + \dots + \frac{1}{n}}_{n \text{ times}}\right) = f(1) = c.$$

ГРЕЧЕСКИЕ БУКВЫ

У `TeX`'а есть несколько греческих букв, которые никогда не встречаются в математических выражениях и только потому включены, что могут понадобиться для набора греческих текстов (в математическом режиме `$. . . $`). `\varsigma` (ς) не следует путать с `\zeta` (ζ), а `\upsilon` (υ) только самую малость шире чем υ). См. также приложение G.

ДВОЙНЫЕ АКЦЕНТЫ

См. акценты в математическом режиме.

ДЕКЛАРИРОВАНИЕ

См. автоматическое форматирование.

ДРОБИ И ИМ ПОДОБНЫЕ

Если при использовании `\thickfrac` вам нужно варьировать толщину черты дроби, то наберите

`\thickfrac\thickness{number}` . . .

`\thickness2` делает черту дроби вдвое толще, `\thickness{1.5}` делает ее толще в 1.5 раза, и т. п.

Если вам нужны ограничители вокруг `\frac`, используйте

`\fracwithdelims(left delimiter)(right delimiter)`

Например, “символ Лежандра” можно получить посредством

`$$\fracwithdelims()ab$$` $\left(\frac{a}{b}\right)$

Можно, конечно, поставить слева и справа `\frac` аб скобки `\left` и `\right`, но использование `\fracwithdelims` более предпочтительно, поскольку в этом

единицы измерения]

случае `TeX` проставляет особые пробелы, которые при обычных `\left... \right` недоступны.

Существует средство `\thickfracwithdelims` для более толстой черты дроби, и поместив сразу за двумя ограничителями `\thickness{(number)}`, можно варьировать толщиной. Например, набрав

```
$$\thickfracwithdelims<>\thickness0nk$
```

можно получить “число Эйлера” $\langle \frac{n}{k} \rangle$.

ЕДИНИЦЫ ИЗМЕРЕНИЯ

Иногда вам бывает нужно попросить `TeX` сделать пробел определенного размера, скажем, 3 дюйма, или 16 пунктов, или 2 сантиметра. Так что вам может понадобиться узнать о том, как `TeX` измеряет.

“Пункты” и “пайки” — вполне традиционные для полиграфии единицы измерения, так что `TeX` их понимает. Он также понимает дюймы и некоторые метрические единицы измерения, равно как и единицы измерения, принятые в континентальной Европе.² Каждая единица измерения задается двухбуквенной аббревиатурой:

<code>pt</code>	пункт (шаг строк в этом издании соответствует 12pt)
<code>pc</code>	пайка (1pc = 12 pt)
<code>in</code>	дюйм (1 in = 72.27 pt)
<code>bp</code>	большой пункт (72 bp = 1 in)
<code>cm</code>	сантиметр (2.54 cm = 1 in)
<code>mm</code>	миллиметр (10 mm = 1 cm)
<code>dd</code>	пункт Дидо (1157 dd = 1238 pt)
<code>cc</code>	цицero (1 cc = 12 dd)
<code>sp</code>	масштабный пункт (65536 sp = 1 pt)

Вот типичные размеры

```
.3 in
1pt
+ 29 pc
-0.01in
0cm
```

Пробел между числом и названием меры не имеет значения, равно как и пробел между числом и знаком + или -. Более того, знак + лишний, но он допустим.

Вам, конечно, не нужно запоминать все эти единицы. Если вы когда-нибудь обращались к `TeX`у с просьбой оставить место для рисунка (см. **вертикальные**

²В том числе и в нашей отечественной полиграфии. В настоящей книге под “пунктами” и “пайками” понимаются единицы измерения, принятые в США и Великобритании, а под “пунктами Дидо” и “цицero” — единицы измерения, принятые в континентальной Европе. Все изложение ведется применительно к англо-американской типометрии. — Прим. перев.

[жирный шрифт для бедных

пробелы) вам, по-видимому, хотелось задавать размеры в дюймах или сантиметрах. Очень важно помнить, что “размеры” всегда изображаются числом и следующим за ним названием меры. Чтобы изобразить один дюйм, следует написать 1in — только in недостаточно. А если вы хотите указать нулевую размерность, (например, см. **многострочные формулы**) следует набирать 0pt или 0cm или 0inch — 0 сам по себе не дает размерность. Обозначение (dimen) будет использоваться везде в тех местах, где должен стоять размер.

Для многих целей набора очень важны две единицы, которые мы еще не упоминали — em и ex. Эти единицы зависят от конкретного шрифта. В шрифте, использованном в этой книге, em-dash имеет длину em, а буква ‘x’ имеет высоту ex; но разработчик шрифта свободен в своем выборе размеров для em и ex. Преимущество использования таких единиц, как em и ex состоит в том, что все пробелы, которые вы укажете, будут автоматически соответствовать размеру шрифта, использованного в данной работе. Единицы em обычно больше подходят для горизонтальных мер, а ex — для вертикальных.

Имейте в виду, что \quad — это просто горизонтальный пробел, в точности равный 1em. Однако \quad нельзя использовать, когда вы хотите указать размер; \quad — это не размер, а команда Т_EX’у вставить пробел размера 1em.

Примеры использования различных размеров вы найдете всюду в этой главе.

ЖИРНЫЙ ШРИФТ ДЛЯ БЕДНЫХ

Теперь А_MS-Т_EX предоставляет полужирный вариант почти всех математических символов. Однако если вам понадобилась какая-то литера в жирном начертании, недоступная через А_MS-Т_EX, то вы всегда можете воспользоваться жирным шрифтом для бедных, применив команду \pmb:

$$\text{\pmb{\Bbb A}\pmb{\geqq}\pmb{x}} \qquad \mathbb{A} \geq x$$

Версия \pmb бинарного оператора или отношения представляет собой другой бинарный оператор или отношение. (Но если вы хотите создать другой большой оператор \sum , сделав \sum толще, вам нужно проконсультироваться с Т_EXнологом.)

ЗАГОЛОВОК

См. Приложение А.

КВАДРАТНЫЕ КОРНИ

См. корни.

КОЛОНТИТУЛЫ

Если вы используете стиль amsppt, то колонтитулы будут сформатированы так же, как в этом учебнике, т. е. текст располагается по центру, а цифры у внешних полей. (На первой странице колонтитул отсутствует, а номер страницы размещается внизу.)

Если колонтитулы не нужны, то в начале документа следует набрать управляющую последовательность \NoRunningHeads сразу после строки \documentstyle. Если колонтитулов нет, то нумерация страниц располагается внизу по центру.

комментарии]

В стиле `amsrpt` колонтитулами служат фамилия автора — на четных полосах, и название статьи — на нечетных полосах. Для монографий принято на четных полосах располагать названия глав, а на нечетных — заголовки разделов (которые вводятся посредством `\head ... \endhead`).

Если нужен какой-нибудь иной колонтитул, например, укороченная форма заголовка, вы можете переопределить текст колонтитулов, набрав следующее:

```
\leftheadtext{<левый колонтитул>}
\rightheadtext{<правый колонтитул>}
```

Эти команды могут появляться в произвольном месте файла, но лучше всего помещать их сразу после `\title` или `\author` или `\head`, чтобы отказаться от автоматически задаваемого колонтитула. Если `\leftheadtext` и `\rightheadtext` вводятся до `\topmatter`, то они не отменяются командами `\title` и `\author`.

Для журнальных статей такое соглашение относительно безболезненно, однако для монографий, в которых принято помещать в качестве колонтитула название раздела, должна отменяться каждая `\head`. Внутренняя команда, используемая управляющей последовательностью `\head`, называется `\headmark`. Если вы хотите предотвратить появление колонтитула от каждой команды `\head`, то вам следует уничтожить определение `\headmark`, переопределив его как

```
\redefine\headmark#1{}
```

(где `#1` есть номер аргумента, как это объясняется на с. 118).

По умолчанию колонтитулы даются прописными буквами. Это оформление можно отменить посредством `\nofrills`, например

```
\rightheadtext\nofrill{<текст колонтитула>}
```

КОММЕНТАРИИ

Вы можете “закомментировать” большой объем текста, набрав

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

Вы **должны** поместить `\endcomment` на отдельной строке.

Процесс просмотра ввода занимает некоторое время, так как Т_ЭХ просматривает массив построчно, так что если вы отметите посредством `\comment` действительно большой массив, Т_ЭХ может довольно долго не подавать признаков жизни.

Будьте внимательны и не вкладывайте конструкции `\comment` друг в друга — **не набирайте что-нибудь вроде**

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

[коммутативные диаграммы

КОММУТАТИВНЫЕ ДИАГРАММЫ

Если математики теряют рассудок от возможности набирать матрицы, то они просто впадут в экстаз, когда смогут набрать “коммутативную диаграмму” вроде такой:

$$(*) \quad \begin{array}{ccc} G & \xrightarrow{\alpha} & H \\ f \downarrow & & \uparrow g \\ G' & \xleftarrow{\beta} & H' \end{array}$$

Коммутативные диаграммы принимают самые разнообразные формы, среди которых

$$(**) \quad \begin{array}{ccccc} \text{Ext}_R^q & \xrightarrow{\quad} & \text{Ext}_R^q(B', B') & & \\ & \searrow & & \nearrow & \\ & & \text{Ext}_R^q(B', B) & & \\ & \nearrow & & \searrow & \\ \text{Ext}_R^q(B', R) & \xrightarrow{\beta} & \text{Ext}_R^q(B, R) & \xrightarrow{\quad} & \text{Ext}_R^q(B, B) \end{array}$$

представляет собой не самый трудный случай, но пока $\mathcal{A}\mathcal{M}\mathcal{S}\text{-T}\mathcal{E}\mathcal{X}$ умеет обрабатывать только “прямоугольные” коммутативные диаграммы вроде (*).

Коммутативная диаграмма (*) была получена посредством `\CD... \endCD`, с `\`, отделяющими отдельные строки, как обычно. Способ получения горизонтальных стрелок объясняется в разделе *стрелки*, так что стрелка с выражением над ней набрана так:

```
\CD
G      @>\alpha>>   H  \
```

а с выражением под —

```
G'    @<<\beta<<   H'
\endCD
```

Направленная вниз вертикальная стрелка представляется посредством `@VVV` — прописная буква *V* должна наводить на мысль об острие стрелы. Присутствие чего-либо между первой и второй буквами *V* дает выражение слева от стрелки, а между второй и третьей — справа. (Разумеется, следует аккуратно пользоваться фигурными скобками, если в выражении присутствуют *V*). Для вертикальных стрелок, указывающих вверх, используется `@AAA`, так как *A* больше всего похожа на перевернутую *V*.³ Таким образом, средняя строка формулы (*) была набрана как

```
@VfVV      @AAgA\\
```

³Вы, возможно, решите, что для направленных вверх стрелок больше подошли бы `@^^^`; увы, по Т_ЭХническим причинам этого делать нельзя.

КОММУТАТИВНЫЕ ДИАГРАММЫ]

а вся диаграмма —

```

$$
\CD
G      @>\alpha>>    H  \\\
@VVfVV          @AAgA \\\
G'     @<<\beta<<    H'
\endCD \tag {$*$}
$$

```

Обратите внимание, что знаки & для разделения столбцов не используются — различные конструкции со стрелками следят за этим сами. Заметьте также, что в средней строке, где имеются только вертикальные стрелки, ничего не следует набирать в “столбец”, содержащий горизонтальные стрелки.

Конструкции @VVV и @AAA можно использовать только внутри \CD, в отличие от @>> и @<<, которые можно использовать в математическом режиме где угодно (хотя обычно они дают более короткие стрелки, чем в коммутативных диаграммах).

Иногда бывает нужен только кусок коммутативной диаграммы:

$$\begin{array}{ccc}
 G & \xrightarrow{\alpha} & H \\
 & & \downarrow g \\
 & & H'
 \end{array}$$

Для отсутствующего “угла” можно набрать {} или даже ничего не набирать, но вместо отсутствующей стрелки обязательно нужно набрать ‘@.’ (по аналогии с \left.’ или \right.’ вместо отсутствующего ограничителя):

```

$$
\CD
G      @>\alpha>> H  \\\
@.          @VVgV \\\
      @.      H'
\endCD
$$

```

Здесь @. используется, разумеется, в совершенно ином смысле, чем в тексте, для получения правильного пробела после прописной буквы с точкой.

Надо сказать, что диаграмма (***) была набрана как

```

$$\define\Ext{\operatorname{Ext}}
\CD
\Ext_R^q @.          @. \Ext_R^q(B',B') \\\
      @. \Ext_R^q(B',B) @.          \\\
\Ext_R^q(B',R) @>\beta>> \Ext_R^q(B,R) @>> \Ext_R^q(B,B)
\endCD
$$

```

[коммутативные диаграммы

Комбинации `\\ \\` дают пустые строки, чтобы выдержать нужное расстояние, а диагональные стрелки и длинная стрелка наверху были нарисованы потом.

Иногда некоторые стрелки в коммутативных диаграммах заменяются на горизонтальные и вертикальные знаки `=`:

$$\begin{array}{ccc} G & \xlongequal{\quad} & G' \\ \parallel & & \downarrow \\ H & \longrightarrow & K \end{array}$$

Нужно только воспользоваться `=` для горизонтального длинного знака `=` и `\@` (или `\@vert`) для вертикального.

К сожалению, в настоящей версии `\CD` стрелки в одних и тех же столбцах автоматически не устанавливаются одной и той же длины. Например,

```


$$\begin{array}{ccc} G & \xrightarrow{\text{Clifford multiplication}} & H \\ \downarrow f & & \uparrow g \\ G' & \xleftarrow{\beta} & H' \end{array}$$


```

дает

$$\begin{array}{ccc} G & \xrightarrow{\text{Clifford multiplication}} & H \\ \downarrow f & & \uparrow g \\ G' & \xleftarrow{\beta} & H' \end{array}$$

Чтобы заставить `AMS-TeX` растянуть нижнюю стрелку, вы должны обеспечить, чтобы текст `\beta` был той же ширины, что и `Clifford multiplication`; вы можете добиться этого, используя `\pretend... \haswidth...`. Если вы наберете

```


$$\begin{array}{ccc} G & \xrightarrow{\text{Clifford multiplication}} & H \\ \downarrow f & & \uparrow g \\ G' & \xleftarrow{\text{Clifford multiplication}} & H' \end{array}$$


```

то получите в результате

$$\begin{array}{ccc} G & \xrightarrow{\text{Clifford multiplication}} & H \\ \downarrow f & & \uparrow g \\ G' & \xleftarrow{\beta} & H' \end{array}$$

матрицы]

Иногда бывает нужно укоротить стрелки в длинной коммутативной диаграмме, чтобы она разместилась на странице. Если вы набрали

```
\minCDarrowwidth{<dimen>}
```

то минимальная длина горизонтальных стрелок будет $\langle \text{dimen} \rangle$ (см. единицы измерения), более длинные стрелки будут использоваться, когда это вызвано длиной выражений при них. Вы можете использовать `\minCDarrowwidth` только внутри знаков `$$`, и применяется это только к выключному материалу.

КОРНИ

Довольно просто получить нечто вроде $\sqrt[3]{x}$ или даже более экстраординарные вещи, например

$$\alpha + \beta \sqrt[3]{1 + \frac{a}{b}}$$

Вы просто набираете

```
$$
\root\alpha+\beta\of{1+\frac ab}
$$
```

$\mathcal{A}\mathcal{M}\mathcal{S}\text{-T}\mathcal{E}\mathcal{X}$ стремится расположить корень правильно, но если вы хотите несколько подправить его расположение, то воспользуйтесь `\uproot{<number>}` после `\root`, чтобы передвинуть корень вверх на $\langle \text{number} \rangle$ единиц, а чтобы сдвинуть его влево на $\langle \text{number} \rangle$ единиц, надо набрать `\leftroot{<number>}` после него. Можно одновременно использовать управляющие последовательности `\uproot{<number>}` и `\leftroot{<number>}` в произвольном порядке, пока ничего иного между ними и `\root` не находится:

```
$$\root \uproot 3 \leftroot{-2}
\alpha+\beta \of{1+\frac ab}$$$
```

$$\alpha + \beta \sqrt[3]{1 + \frac{a}{b}}$$

Доли, на которые перемещается `\root`, чрезвычайно мелкие, так что легко добиться желаемого лоска.

МАТРИЦЫ

Использование `\matrix` в тексте не приводит к изменению размера матрицы.

Так что вы можете столкнуться с вещами вроде $\begin{pmatrix} a & b \\ c & d \end{pmatrix}$, что представляет собой не что иное, как сложную матрицу, которую автор не захотел сделать выключной. Но можно также получить в тексте $\begin{pmatrix} a & b \\ c & d \end{pmatrix}$, если набрать

```
$$\left(\smallmatrix a&b\
c&d\endsmallmatrix\righ)$
```

Нет таких вещей как `\smallmatrix`, и т. д. — нужно просто ставить соответствующие ограничители. Для `\smallmatrix` допускается использование `\format` и `\vspace`, а `\spreadmatrixlines` (см. вертикальные пробелы) не допускается.

Если набрать нечто вроде

```


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


```

то получится

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

Строки, содержащие меньше членов, чем максимальное число знаков `&`, снабжаются пустыми местами в остальных столбцах.

`\matrix` и родственные им конструкции допускают любое количество столбцов, но когда после `\matrix` вы набираете строку `\format...\\`, количество знаков `&` обычно определяет максимальное число знаков `&`, допустимое в каждой строке. Тем не менее, имеется особое средство для работы с форматами в периодических структурах. Если вместо `&` набрать `&&`, то часть первой “строки”, которая идет дальше, будет повторяться снова и снова. Например,

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

определяет матрицу, в которой первый столбец представляет собой выровненное влево множество, за которым следует произвольное число столбцов, каждый из которых выровнен по левому краю и предваряется пробелом `\quad`. Точно так же знак `&` в самом начале приводит к периодической структуре

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

дающей произвольное число столбцов, каждый из которых выровнен по левому краю и начинается с пробела (что, возможно, совсем не то, чего вы ожидали). Вы можете только иметь один `&&` или начальный `&` в первой “строке”.

Иногда матрица

$$\begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{pmatrix}$$

представляется несколько иначе:

$$\begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{pmatrix}$$

матрицы]

Этого можно добиться, набрав

```

$$
\pmatrix a_{11} & a_{12} & \& \hdots & a_{1n} \\
a_{21} & a_{22} & \& \hdots & a_{2n} \\
\hdotsfor 4 \\
\hdotsfor 4 \\
a_{m1} & a_{m2} & \& \hdots & a_{mn} \\
\endpmatrix
$$

```

Здесь `\hdotsfor 4` указывает Т_EX'у, что следует напечатать горизонтальные точки на протяжении 4-х столбцов; число столбцов должно указываться, потому что Т_EX не знает наверняка, сколько столбцов в матрице, пока не закончит работу с ней. Убедитесь, что вы поставили `\\` после `\hdotsfor 4`, если только далее не следует '&...'.
 Увы, `\hdotsfor` не работает так четко, если вы хотите поместить точки после первого столбца. Если вы наберете

```

$$
\pmatrix a_{11} & a_{12} & \& \hdots & a_{1n} \\
a_{21} & a_{22} & \& \hdots & a_{2n} \\
a_{31} & \hdotsfor 3 \\
\hdotsfor 4 \\
a_{m1} & a_{m2} & \& \hdots & a_{mn} \\
\endpmatrix
$$

```

то получите

$$\begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ a_{31} & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{pmatrix}$$

Это смотрится неважно, потому что `\hdotsfor 3` протянулось через *весь* второй столбец, а часть этого столбца содержит пробел `\quad`, отделяющий его от предыдущего столбца. Если вам действительно нужны точки внутри столбцов, правильно будет набрать

```

$$
\pmatrix a_{11} & a_{12} & \& \hdots & a_{1n} \\
a_{21} & a_{22} & \& \hdots & a_{2n} \\
a_{31} & \innerhdotsfor 3 \after \quad \\
\hdotsfor 4 \\
a_{m1} & a_{m2} & \& \hdots & a_{mn} \\
\endpmatrix
$$

```

[многострочные формулы]

Конструкция `\innerdotsfor... \after{...}` дает точки для указанного числа столбцов, но начинаются они после `\quad` (или иного использованного вами интервала, если применяется команда `\format`). Не забудьте поставить `*` перед `\innerdotsfor`.

Однако, если вам не нравятся пробелы между точками, которые дает средство `\hdotsfor`, можете воспользоваться конструкцией `\spacehdots <number>\for... \for... \for... \for...` для некоторого `<number>` (в десятичном виде). Обычно в `\hdotsfor` применяется `<number> = 1.5`; меньшее `<number>` даст более тесно расположенные точки, а большее `<number>` расставит их более редко. Здесь также имеется

```
\spaceinnerhdots<number>\for... \after...
```

МНОГОСТРОЧНЫЕ ФОРМУЛЫ

Для формул, состоящих из нескольких строк, на самом деле достаточно воспользоваться `\multiline`, не забывая при этом проставлять между строками, как обычно, `\\`. Все строки между первой и последней обычно в этом случае располагаются по центру. Однако, одну из этих строк можно отодвинуть влево или вправо, набрав

```
\shoveleft{...}\\
```

или

```
\shoveright{...}\\
```

Обратите внимание, что `\shoveleft` и `\shoveright` представляют собой управляющие последовательности с одним аргументом; вся строка, которую вы хотите отодвинуть, заключается в фигурные скобки, и в конце все так же ставится `\\`.

Для изменения межстрочных пробелов в `\multiline` можете пользоваться управляющими последовательностями `\vspace` и `\spreadlines` (см. вертикальные пробелы).

Можно также менять расстояния до левого и правого краев, которые оставляет управляющая последовательность `\multiline`. Это осуществляется при помощи `\multlinegap{<dimen>}` (см. единицы измерения). Например, если ваша формула `\multiline` несколько шире, чем нужно, чтобы уместиться в отведенное ей место, можно удалить пробелы слева и справа, набрав

```
$$
\multlinegap{Opt}
\multiline ...
... \endmultiline
$$
```

Когда формула завершится, прежние пробелы будут восстановлены. Обратите внимание, что следует указывать `Opt`, а не просто `0`. Для `\multlinegap{Opt}` имеется также синоним `\nomultlinegap`.

Управляющая последовательность `\multlinegap` допускается только внутри знаков `$$`. Но имеется также `\MultLineGap` для изменения расстояния до полей на все время работы; таким образом, создается новый стиль. Естественно, большинством журнальных стилей `\MultLineGap` игнорируется.

непрерывные дроби]

МОДУЛЬ (MOD)

Слово ‘mod’ обычно встречается в формулах в прямом начертании, но то, что за ним скрывается, функционально отличается от таких операторов как `sin` и `cos`. Обычно ‘mod’ используется в двух различных смыслах; если ‘mod’ стоит между двумя величинами и используется как бинарный оператор вроде `+`, то употребляется `\bmod`, а в конце формул для ‘mod’, заключенного в скобки, употребляется `\pmod`:

$$\begin{aligned} \text{\$}\gcd(m,n)=\gcd(n,m\bmod n)\text{\$} & \quad \gcd(m,n) = \gcd(n, m \bmod n) \\ \text{\$}x\equiv y+1\pmod{m^2}\text{\$} & \quad x \equiv y + 1 \pmod{m^2} \end{aligned}$$

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

Отдельные авторы предпочитают во второй из приведенных формул ставить ‘mod’ без скобок: `mod m2`. Поэтому А_MS-Т_EX также обладает средством `\mod` для этой цели:

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

Некоторые предпочитают формулу

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

вовсе без ‘mod’, но зато в скобках. Для них предпочтительнее команда `\pod`:

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

НАГРОМОЖДЕНИЕ СИМВОЛОВ

См. составные символы.

НЕПРЕРЫВНЫЕ ДРОБИ

Математики иногда любят вставить “непрерывную дробь” вроде

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

которую даже трудно себе вообразить, не то что набрать! Но А_MS-Т_EX предоставляет вам простой способ их описания. Приведенная выше выключная формула была набрана так:

```

\text{\$}a_0 + \cfrac{1}{
  a_1 + \cfrac{1}{
    a_2 + \cfrac{1}{
      a_3 + \cfrac{1}{
        a_4\endcfrac
      }
    }
  }
}

```

Поняли мысль? Каждый раз, как только вы начинаете новую “поддробь”, набирайте `\cfrac` и используйте, как обычно, `\\` чтобы разбить на отдельные строки. Затем все это заканчиваете единственным `\endcfrac`.⁴

Некоторые математики предпочитают что-то вроде

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

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

```

$$a_0 + \lcfrac{1\\
  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`.

НОВЫЕ СИМВОЛЫ

См. составные символы и дроби и им подобные (а также акценты, команда `\accentedsymbol`).

Когда вы пользуетесь стилем `amspt`, все символы в шрифтах `msam` и `msbm` можно получить при помощи управляющих последовательностей; например, `\lpprox` даст \lesssim . Если вы не пользуетесь стилем `amspt`, то те же последовательности можно определить, набрав `\loadmsam`, `\loadmsbm` и `\UseAMSsymbols`. Эти команды определяют все управляющие последовательности, перечисленные в приложении G.

Если вам нужно несколько из этих дополнительных символов и вы не пользуетесь стилем `amspt`, то все еще нужно набрать `\loadmsam` и `\loadmsbm`, но вместо `\UseAMSsymbols` достаточно воспользоваться командой `\newsymbol`, чтобы создать управляющую последовательность с соответствующим названием нужного вам символа. Управляющие последовательности имеют либо “стандартные” названия, либо перечисленные в приложении G, либо одно из выбранных вами.

В списке символов в приложении G указан сам символ, его 4-х литерный идентификатор, и “стандартное” название символа. (Первая литера идентификатора обозначает шрифтовое семейство, которому символ принадлежит. Символы из семейства `msam` в качестве первой литеры имеют 1, а из семейства `msbm` — 2.) Например, символ \leslant описан как

```

1 230A \nleqslant

```

⁴Если бы вы набрали `\cfrac a+1 \b+1 \endcfrac`, то получили бы в точности то же самое, как если бы набрали `\dfrac{a+1}{b+1}`. Но для обычных дробей использование `\cfrac` не рекомендуется!

нумерация формул]

Чтобы получить управляющую последовательность с таким именем, следует набрать команду

```
\newsymbol\nleqslant 230A
```

После чего управляющая последовательность `\nleqslant` будет давать символ \nleqslant в математическом режиме, который будет действовать точно так же, как “бинарное отношение”.

В plain Т_EX’e некоторые символы получаются посредством комбинации символов из семейства Computer Modern. Некоторые из этих комбинированных символов в шрифтах `msam` и `msbm` заменены единичными символами. Это `\angle` (\angle) и `\hbar` (\hbar) из группы “Мозаика символов” и `\rightleftharpoons` (\rightleftharpoons) из группы “Стрелки” в приложении G.

Новые символы правильно меняют свои размеры в верхних и нижних индексах, при условии, что вы пользуетесь подходящими переопределениями. Чтобы использовать `\newsymbol` для изменения существующего определения, это имя должно быть сначала не определено (`undefined`). Далее приводятся строки, которые вы должны поместить в свой файл, если вы не используете стиль препринта или `\UseAMSsymbols` (в котором переопределение производится автоматически):

```
\undefine\angle
\newsymbol\angle 105C
\undefine\hbar
\newsymbol\hbar 207E
\undefine\rightleftharpoons
\newsymbol\rightleftharpoons 130A
```

Список этих символов приводится в приложении G с пометкой “(U)”, напоминающей нам, что они должны быть неопределенными (`undefined`).

Обратите внимание, что в этих таблицах некоторые символы представлены под двумя названиями. В подобных случаях любое из них можно использовать для получения этого символа.

НУМЕРАЦИЯ СТРАНИЦ

См. постраничная нумерация.

НУМЕРАЦИЯ ФОРМУЛ

В стиле `amspt` нумерацию формул слева можно заменить нумерацией справа, набрав управляющую команду `\TagsOnRight`. Чтобы вернуть нумерацию слева, надо набрать `\TagsOnLeft`. Эти команды “глобальные”: они действуют на весь последующий текст, даже если используются внутри группы `{...}` или между знаками `$`. Разумеется, журнальные стили такие инструкции преимущественно игнорируют.

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

[ограничители

Если у вас масса формул, нумерованных как (A_1) и (A') или $(*)$ и $(**)$, но черточки и дефисы в нумерации используются редко, то вы можете предложить \LaTeX трактовать номера формул как математические формулы, а не как текст. Для этого наберите `\TagsAsMath`; а `\TagsAsText` вернет противоположное соглашение (эти команды также “глобальные”). Журнальные стили такие инструкции не игнорируют, поскольку они никак не отражаются на выводе, а только на способе набора текста. Определенно нельзя назвать удачной мыслью постоянно переключаться с `\TagsAsMath` на `\TagsAsText`, поскольку вы можете запугаться и набрать `\tag A_2`, когда следовало бы набирать `\tag {\$A_2\$}`, чем усложните себе жизнь.

В некоторых форматах номера формул располагаются по центру даже для формул, разбитых на несколько строк. Вы можете сделать это и в стиле `amsrpt`, если наберете

```
\CenteredTagsOnSplit
```

А затем можете вернуться к обычной расстановке номеров:

```
\TopOrBottomTagsOnSplit
```

Это также “глобальные” команды. Если вам почему-то нужно получить центрированный номер только для одной разбитой на несколько строк формулы (скажем, чтобы лучше расположить ее на странице), эти управляющие последовательности использовать не нужно; просто наберите такую формулу как `\aligned`.

Если внутри текста нужно сослаться на формулу (17), не забывайте об одной небольшой тонкости: в разных стилях это может делаться по-разному, например, `[17]` или `17` (наиболее распространен способ заключения номера в круглые скобки). По этому поводу \LaTeX предлагает вам набирать `\thetag{17}`; тогда вы получите именно то, что принято в данном стиле.

Вам может понадобиться отказаться от стиля для какой-нибудь конкретной формулы. Например, по той или иной причине обязательно нужно получить для данной формулы полужирный номер в полужирных скобках (3). Для этого используйте свойство `\tag'a` “аргумент в виде литеры”: наберите

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

и номер появится точно в том виде, как вы его набрали между двойными кавычками “ (сравните сноски и список).

ОГРАНИЧИТЕЛИ

Ограничители `|` и `\|` довольно специальные, поскольку те же символы служат и левым, и правым ограничителем. Когда ограничителям `|` и `\|` не предшествуют `\left` или `\right`, у \TeX нет способа узнать, в каком смысле они используются, так что он трактует их просто как обычные символы. Это вполне удовлетворительно работает для формул типа $|x|$, но пробелы могут оказаться неправильными, если туда входят бинарные операторы, так что даже для формул обычного размера могут понадобиться `\left` и `\right`:

```


$$|-x|=|+x|$$


$$\left|-x\right|=\left|+x\right|$$


```

переносы]

В первом случае пробелы неправильные, поскольку Трех здесь набирает сумму и разность элементов ‘|’ и ‘x’; во втором случае `\left` и `\right` предоставляют Трех’у необходимую информацию для правильного набора формулы.

Кроме ограничителей, упомянутых в гл. 13, имеются следующие:

обратная косая: \	<code>\backslash</code>
стрелка вверх: ↑	<code>\uparrow</code>
двойная стрелка вверх: ⇑	<code>\Uparrow</code>
стрелка вниз: ↓	<code>\downarrow</code>
двойная стрелка вниз: ⇓	<code>\Downarrow</code>
стрелка вверх-вниз: ⇕	<code>\updownarrow</code>
двойная стрелка вверх-вниз: ⇔	<code>\Updownarrow</code>

См. также большие ограничители и теоретико-множественные обозначения.

∎ Существует еще несколько довольно специальных ограничителей. Например, `\lgroup` и `\rgroup` позволяют получать ограничители, составленные из повторяющихся частей вертикальных стрелок и двойных вертикальных стрелок, но без самой стрелки. Результат аналогичен `\vert` или `\Vert`, но эти линии несколько отличаются по толщине, и пробелы между ними чуть-чуть шире. Средство `\bracevert` дает ограничитель, составленный из повторяющихся частей больших фигурных скобок, но без закруглений. Можно также воспользоваться `\lgroup` и `\rgroup`, которые построены из фигурных скобок без средней части, и `\lmoustache` и `\rmoustache`, которые дают верхние и нижние половины больших фигурных скобок. Здесь представлены варианты `\bigg` для `\vert`, `\Vert`, `\lgroup`, `\rgroup` и пять других специальных ограничителей.

$$\dots \left| \dots \right| \left| \dots \right| \left| \dots \right| \left| \dots \right| \left| \dots \right| \left(\dots \right) \dots \int \dots \int$$

Скобки `\lgroup` и `\rgroup` иногда оказываются лучше круглых скобок для больших матриц; они доступны только в размере `\Big` или больше.

ОПЕРАТОРЫ

Мы уже упоминали, что если оператор состоит из двух частей, как, например, `\lim inf`, между этими частями должна быть тонкая шпация. При использовании `\operatorname` или `\operatornamewithlimits` символы ‘,’, ‘*’, ‘-’, ‘/’, ‘:’, ‘;’ и ‘,’ трактуются как обычные текстовые символы, при этом ‘‘’ и ‘’’ дают “и”, а ‘--’ и ‘---’ дают – и —. После точек, запятых и двоеточий будет крошечный пробел.

ПЕРЕНОСЫ

Если вы имеете дело с вызывающими сомнения словами вроде `databases` или `Treemunch`, и вам бы хотелось знать, что думает Трех по поводу их переноса, наберите просто

```
\showhyphens{databases Treemunch}
```

(либо в самом файле, либо сразу после запроса Трех’a). В результате Трех выведет на ваш терминал следующее:

[переход на другую строку или страницу

```
Underfull \hbox (badness 10000) detected at line ...
[] \tenrm databases Treemu-nch
```

из чего заключаем, что Т_ЕХ не нашел ни одного места для переноса `databases`, а относительно `Treemunch` полагает, что перенос следует делать после `u`. Не обращайтесь внимание на сообщение `Underfull \hbox` — это побочный эффект работы `\showhyphens`.

См. также разрыв строк.

ПЕРЕХОД НА ДРУГУЮ СТРОКУ ИЛИ СТРАНИЦУ

См. разрыв строк и разрыв страницы.

ПОДПИСИ

См. вставка с подписью.

ПОЛУЖИРНЫЙ МАТЕМАТИЧЕСКИЙ ШРИФТ

См. шрифты в математическом режиме.

ПОСТРАНИЧНАЯ НУМЕРАЦИЯ

Если вы пользуетесь стилем `amsprt`⁵ и наберете `\NoPageNumbers` в начале своего документа (после строки `\documentstyle`), нумерация внизу страниц исчезнет.

ПРОВЕРКА СИНТАКСИСА

Пропуская первый раз через Т_ЕХ свой файл, вы, вероятно, хотите лишь выловить все сделанные вами ошибки и не рассчитываете получить окончательную распечатку. Если в начале своего документа, после строки `\documentstyle` вы наберете `\syntax`, то Т_ЕХ не выдаст никакого продукта, но станет работать раза в 2–4 быстрее и проверит все синтаксические ошибки.

Есть только одна закладка: `\syntax` не даст ни одного сообщения `Overfull box` о переполнении. Так что попользовавшись `\syntax`, можете заменить его на `\galley`s. Это приводит к тому, что Т_ЕХ имитирует режим типографской гранки — так что вы получите сообщение о том, где и что не помещается — но окончательные страницы никогда не смонтируются, и вы не получите никакой распечатки. Вы, вероятно, сочтете `\galley`s достаточно бесполезным, поскольку скорость Т_ЕХ'a увеличивается лишь чуть-чуть, а зачастую вам бывает трудно осмысленно обработать `Overfull box`, пока вы не увидите распечатку или ее экранный вариант.


Если вы не хотите возвращаться в файл и менять или удалять `\syntax`, то можете воспользоваться другим свойством АМ_S-Т_ЕХ'a. Наберите `\printoptions`, и АМ_S-Т_ЕХ спросит вас

```
Do you want S(yntax check), G(alley) or P(ages)
Type S, G or P, follow by <return>
```

⁵ Если вы применяете АМ_S-Т_ЕХ, но не используете стиль `amsprt`, то автоматически получите нумерацию внизу страниц (стиль `plain` Т_ЕХ'a). Чтобы отменить нумерацию страниц, когда не используется стиль `amsprt`, можно набрать команду `\nopagenumbers` `plain` Т_ЕХ'a. Однако команда `\nopagenumbers` не работает со стилем `amsprt` — в этом случае следует использовать `\NoPageNumbers`.

размер страницы]

(Что вы желаете: проверку синтаксиса (S), гранки (G) или распечатку (P)? Наберите S, G, или P и затем нажмите `<return>`.) Можете нажать S или s и т. п., но если вы наберете что-нибудь совсем неподходящее, вопрос повторится. Можно нажать нужную клавишу не дожидаясь вопроса, пока Т_ЕХ тратит свое время на чтение файла `amstex.tex` и стилевого файла; Т_ЕХ сделает для себя пометку о вашем ответе, хотя ответ не появится на экране до подходящего момента.

 Это исключительно для толмачей и зануд. А_МS-Т_ЕХ занимает довольно много боксов, размерностей и пр., но эти вещи обычно в `transcript`-файле отсутствуют, поскольку не представляют интереса для обычного пользователя. Если набрать в начале файла `\showallocations`, то в `transcript`-файле появятся сведения о распределении переменных.

ПРОБЕЛЫ

См. горизонтальные пробелы и вертикальные пробелы.

ПРЯМОУГОЛЬНИКИ, ПЕРЕПОЛНЕНИЕ И НЕДОГРУЗКА

Уже возникали разнообразные примеры сообщений `Overfull \hbox`, попадались и сообщения `Underfull \hbox`. Мы ссылались на них обычно просто как на сообщения `Overfull box`, потому что у Т_ЕХ'a имеются также и сообщения `Overfull \vbox` и `Underfull \vbox`. Вы, скорее всего, можете получить такое сообщение, когда Т_ЕХ не способен найти хороший разрыв при переходе на следующую страницу. Например, если вы задали `\aligned` так, что он оказался длиннее, чем страница, то Т_ЕХ все-таки разместит это на странице и затем пожалуется на `Overfull \vbox`. В этом случае `\vbox` — это вся страница, которую Т_ЕХ складывает из различных выровненных вертикально посредством `\aligned` строк. И разумеется, на предыдущей странице может оказаться недостаточно материала, чтобы ее заполнить, в связи с чем вы получите сообщение `Underfull \vbox` о странице, непосредственно предшествующей `Overfull \vbox`. Даже если материал с `\aligned` не занимает всю страницу, его, возможно, придется расположить с самого верха новой страницы до того, как текущая страница будет заполнена, и вы снова получите сообщение `Underfull \vbox`.

РАЗМЕР СТРАНИЦЫ

Можно изменить размер страницы в стиле `amsrpt`, набрав

```
\pagewidth{<dimen>}
\pageheight{<dimen>}
```

указав подходящий `<dimen>` в качестве ширины и высоты страницы (*см. единицы измерения*). В высоту страницы включается и строка для номера. Другие стили эти команды, по-видимому, проигнорируют.

Когда вы меняете размеры страницы, вы, по-видимому, можете захотеть поменять и ее расположение на стандартном листе формата А4 ($8\frac{1}{2}$ на 11 дюймов)⁶. Набрав

```
\hcorrection{<dimen>}
```

⁶Указаны размеры для формата А4, принятого в США. У нас принят формат 210 на 297 мм. — *Прим. перев.*

[размер шрифта набора

вы передвинете всю страницу по горизонтали вправо на $\langle \text{dimen} \rangle$, а команда

```
\vcorrection{ $\langle \text{dimen} \rangle$ }
```

опустит страницу вниз на $\langle \text{dimen} \rangle$.

РАЗМЕР ШРИФТА НАБОРА

В стиле `amsrpt` весь текст набирается шрифтом размера `\tenpoint`, за исключением подстрочных примечаний и библиографии (см. приложение В), которые автоматически настроены на шрифт размера `\eightpoint`. Чтобы уменьшить размер шрифта, можете самостоятельно задать `\eightpoint`. Это бывает полезно для выключных формул, которые не умещаются на страницу по ширине.

РАЗРЫВ ВЫКЛЮЧНЫХ ФОРМУЛ

См. выравниваемые формулы.

РАЗРЫВ СЛОВ

Чтобы дать понять \TeX 'у, как разорвать слово в тексте, можно воспользоваться знаком возможного переноса `\-` или `\hyphenation`. Но иногда бывает нужно запретить перенос. Один из способов заключается в том, чтобы набрать `\text{strangeword}`, потому что `\text` всегда дает неразрываемый массив, даже в обычном тексте. Или же можно набрать возможный перенос `\-` в самом конце слова. Это действует потому, что \TeX никогда не добавит перенос в то слово, где уже есть дефис или возможный перенос. Или же можно вставить в файл `\hyphenation{strangeword}`.

РАЗРЫВ СТРАНИЦЫ

Аналогично средству `\linebreak`, \LaTeX - \TeX предоставляет команду `\pagebreak`. Если она применяется между абзацами, то приводит к разрыву текста на странице, причем оставшийся на данной странице текст растягивается до нижнего поля. При использовании внутри абзаца `\pagebreak` приводит к переносу текста на следующую страницу *в конце той строки, на которой он ставится*. В обычном математическом режиме `\pagebreak` не допускается, но он допустим в *выключных* формулах. На самом деле правильный способ вызвать разрыв страницы после выключной формулы состоит в использовании `\pagebreak` *внутри* этой выключной формулы (если вы используете `\pagebreak` сразу после выключной формулы, то на странице появится дополнительное пустое пространство после этой формулы).

Точно таким же способом для предотвращения разрыва страницы используется `\nopagebreak`. В частности, если нужно избежать разрыва страницы после выключной формулы, эту команду следует набирать *внутри* нее (набранная после выключной формулы, она не даст никакого эффекта).

Команда \LaTeX - \TeX 'а `\newpage` может использоваться только между абзацами; она вызывает разрыв текста и перенос его на следующую страницу, а оставшаяся часть данной страницы остается пустой (по аналогии с `\newline`).

У \LaTeX - \TeX 'а нет никаких `\allowpagebreak` для разрешения разрывать страницу, поскольку страницу обычно можно рвать между любыми строками текста. (Относительно разрыва страницы между строками выключной формулы см. *выравниваемые формулы*.)

разрыв строк]

РАЗРЫВ СТРОК

Мы уже изучали управляющую последовательность `\linebreak` для принудительного перехода на другую строку. У \LaTeX -а также имеется `\newline`. Используя `\newline` вместо `\linebreak` вы вызываете переход на следующую строку, но текст в этой строке не простирается до правого края, так что вы получаете вещи вроде этих, набирая

```
... но текст в этой строке не простирается до правого
края,\newline так что вы получаете вещи\newline
```

Если набрать `\newline\newline`, то образуется пустая строка. В конце абзаца нет необходимости помещать `\newline`, но можно и поместить. (Если вы делаете что-нибудь достаточно странное, например, помещаете в конце абзаца `\newline\newline`, то получите ложное сообщение `Underfull \hbox`, которое можете просто проигнорировать.)

Связка `~` предотвращает разрыв строки, но при этом вставляет обычный межсловный интервал. Иногда для предотвращения разрыва строки может понадобиться команда `\nolinebreak`, которая не дает такого побочного действия. Например, вы можете захотеть поместить `\nolinebreak` в конце предложения, если по той или иной причине нежелательно заканчивать предложение в конце строки. Тогда следует оставить пробел между точкой и следующей за ней командой `\nolinebreak`.

И, наконец, управляющая последовательность `\allowlinebreak` разрешает сделать пробел там, где обычно \TeX его не делает. Например, обычно разрывы строк допускаются только после дефисов и тире; вы не можете начать строку с дефиса, тире или короткого тире (если только перед ним не было чего-либо вроде выключной формулы). Но иногда может понадобиться, чтобы тире попало в начало следующей строки, например потому, что оно не умещается в конце предыдущей строки. Для этого достаточно набрать `'... \allowlinebreak---...'` и здесь будет разрешен разрыв строки.

Пробелы перед `\allowlinebreak` не игнорируются, но использование управляющей последовательности `\allowlinebreak` между словами не даст никакого эффекта, поскольку между словами всегда разрешается разрыв строки. В частности, `\allowlinebreak` не поможет избавиться от `Overfull \hbox`; это происходит потому, что \TeX не в состоянии растягивать пробелы как угодно широко, и только `\linebreak` может вынудить \TeX преодолеть свое упорство.

Кроме использования `\allowlinebreak` перед тире, вам может понадобиться воспользоваться этим средством после слэша. \TeX не переносит слова типа "ввод/вывод", в которых содержатся небуквы, поэтому он не разорвет слово после `/`. Так что вы можете прибегнуть к помощи `'ввод/\allowlinebreak вывод'`, если это слово оказалось камнем преткновения.

У \TeX -а имеется также управляющая последовательность `\slash` для указания `/`, где \TeX может разорвать строку, так что можно просто набирать `'ввод\slash вывод'`. Это несколько отличается от `\allowlinebreak`, поскольку \TeX рассматривает разрыв строки после `\slash` как нечто нежелательное, столь же дурного тона, как и разрыв перед дефисом (насколько это дурной тон определяется в стилевом файле). \LaTeX игнорирует любые пробелы перед `\slash`.

РАЗРЫВ ФОРМУЛ

Для представления “факультативного умножения” \TeX предоставляет средство $\backslash*$ в формулах — если строка разрывается в этом месте, то в конце строки появится знак \times .

См. также выравниваемые формулы.

РАСПОРКИ

Когда вы берете $\backslash\sqrt$ от некой формулы, то размер $\sqrt{\quad}$ зависит от формулы: сравните \sqrt{a} и \sqrt{d} и \sqrt{y} . Если в вашей формуле только один $\backslash\sqrt$, все прекрасно, но вот вместо формулы $\sqrt{a} + \sqrt{d} + \sqrt{y}$ вам бы хотелось иметь более равномерно расположенное выражение: $\sqrt{a} + \sqrt{d} + \sqrt{y}$. Этого можно достичь посредством

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

\mathstrut представляет собой невидимый символ, который простирается над и под строкой на расстояние, достаточное для того, чтобы перекрыть любую букву. См. фантомы.

СВЕДЕНИЯ ОБ ИЗДАНИИ

Журнальная статья может начинаться со специального материала, вроде замечания о том, что данная статья была представлена как приветственная речь и т. п. Стиль amsrpt не предвосхищает подобные вещи, но имеет

```
\pretitle{...}
\preauthor{...}
\preaffil{...}
\predate{...}
\preabstract{...}
\prepaper{...}
```

для оформления вспомогательного материала, который может быть помещен до $\backslash\text{title}$, $\backslash\text{author}$, и пр. Но вы не сможете этим воспользоваться до тех пор, пока вы не узнаете достаточно о самом \TeX 'е; внутри каждой такой управляющей последовательности должны быть непосредственно указаны все вертикальные пробелы, замены шрифтов и т. д.

СВЯЗКИ

В гл. 5 мы уже упоминали, что будет правильно набирать связку ~ после аббревиатур с точкой в середине предложения. Далее приводится список ситуаций, где такая связка уместна (следуя *The \TeX book*).

- (1) В ссылках на нумерованные разделы документа:
- | | |
|------------------------|-------------------------------|
| Глава $\text{~}12$ | Теорема $\text{~}1.2$ |
| Приложение $\text{~}A$ | Таблица $\text{~}B\text{--}8$ |
| Рисунок $\text{~}3$ | Леммы 5 и $\text{~}6$ |

(Использование -- объяснялось в замечании в конце гл. 5.)

сноски]

- (2) Между фамилией или именем и инициалами или другими именами:
 Donald E. Knuth Luis I. Trabb-Pardo
 Bartel Leendert van der Waerden Charles XII
 Арнольд И. В. Майкл Д. Спивак

Но если полное имя очень длинное, вы не можете рискнуть связать его воедино:

Charles Louis Xavier Joseph de la Vall\`ee Poussin.

- (3) Между математическими символами и определяющим их словом:
 dimension`\$d\$ width`\$w\$ function`\$f(x)\$
 размерность`\$d\$ ширина`\$w\$ функция`\$f(x)\$
 string`\$\$ of length`\$l\$
 цепочка`\$\$ длины`\$l\$

Но сравните последний пример с таким:

string`\$\$ of length \$l\$`or more.

цепочка`\$\$ длины \$l\$`или больше

- (4) Между символами при перечислении:
 1, 2, или 3
 \$a\$, \$b\$, и \$c\$.
 1, 2, \dots, \$n\$

- (5) Когда символ плотно примыкает к предлогу:
 of`\$x\$
 from 0 to`1
 increase \$z\$ by `1
 in common with `\$m\$`.
 от 0 до`\$\infty\$

Это правило неприменимо к составным объектам, но:

of \$u\$`and`\$v\$.

- (6) Когда математическое выражение представлено словесно:
 equals`\$n\$ less than`\$\epsilon\$ (given`\$\$)
 mod`2 modulo`\$p\$`e\$ for all large`\$n\$

Сравните 'is`15` с 'is 15`times the height`.

- (7) Когда перечисление происходит внутри абзаца:
 (b) Покажите, что \$f(x)\$ (1)`непрерывна; (2)`ограничена.

У-ухх!

СИМВОЛ НАД СИМВОЛОМ

См. составные символы.

СНОСКИ

В стиле `amsprt` в качестве номеров сносок используются цифры в верхних индексах: ¹, ², ..., но иногда может понадобиться что-нибудь иное. Например, в стилях некоторых журналов к `\title` или `\author` делаются примечания для указания места работы или благодарностей за поддержку, которые помечаются посредством *, **, и т. п., несмотря на то, что в самом тексте используются

другие маркеры.⁷ Чтобы получить ту пометку, какую вы хотите, воспользуйтесь `\footnote` с аргументом в виде “литеры”: просто заключите этот маркер в двойные кавычки и поместите его непосредственно после `\footnote`. Например, для сноски с пометкой * наберите

```
\footnote"*{...}
```

Пробел после первой " или перед второй не игнорируется. (С другой стороны, если вы набрали `\footnote"*{...}`, то не получите совсем никакого знака сноски, а просто примечание в конце страницы.)

Хотя \LaTeX специально заботится о том, чтобы сноски можно было использовать внутри таких вещей, как `\title`, `\author`, и т. п. иногда вы можете обнаружить, что `\footnote` просто исчезает, поскольку вы ввели ее внутри какой-то иной конструкции вроде выключной формулы

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

Но вы можете справиться даже и с этими странными вещами, так как `\footnote` можно разложить на пару управляющих последовательностей: `\footnotemark{...}` эквивалентно набору

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

Управляющая последовательность `\footnotemark` даст пометки сноска^{1, 2} и т. д. в основном тексте, тогда как `\footnotetext` породит примечание внизу, знак сноски у которого будет тот же, что и у самого последнего `\footnotemark` или у команды `\footnote`. Так что вы можете набрать

```
$$
y=x^2 \quad \text{для некоторого } x > 0
$$
\footnotetext{Вот уж, действительно, не место для сноски!}%
Но вы можете справиться ...
```

где `\footnotetext` появляется вслед за `\footnotemark`, как только такая возможность возникает, чтобы примечание оказалось на той же самой странице.⁹ Обратите внимание, что после `\footnotemark` нужен `_`; пробел не вставляется автоматически, поскольку вы можете не пожелать этого, если за `\footnotemark` стоит запятая, например. (С другой стороны, пробел перед `\footnotemark` всегда игнорируется.) Следует также быть внимательным — не вводить дополнительный пробел перед текстом, который следует за `\footnotetext`.

Даже `\footnotemark` и `\footnotetext` не избавят вас от проблем, если вы захотите сделать два примечания к одной и той же выключной формуле: по видимому, тогда следует дважды внутри формулы набрать `\footnotemark` и

⁷ В идеале каждый журнальный стиль должен сам об этом заботиться, но пока это не так.

⁸ Вот уж, действительно, не место для сноски!

⁹ Чтобы не произошло перехода на другую страницу, вставьте `\pagebreak` непосредственно перед `\footnotetext`, но если вы имеете дело с выключной формулой, не забудьте поместить `\pagebreak` внутрь выключной формулы (см. разрыв страницы).

сноски]

сразу же после них дважды набрать `\footnotetext`. Но теперь первая управляющая последовательность `\footnotetext` получила инструкцию дать сноске номер второго `\footnotemark`! Так что А_МС-Т_ЕХ предоставляет также средство `\adjustfootnotemark`, позволяющее изменить текущий номер сноски. Набрав

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

вы уменьшите текущий номер на единицу, что даст правильный номер для первой `\footnotetext`, затем, увеличив его на 1 снова, получите правильный номер для второй `\footnotetext`.

И `\footnotemark`, и `\footnotetext` допускают использование аргумента в виде “литеры” “...”. Так что^{10,11} вы можете даже проделывать такие вещи: надо просто набрать

```
Так что\footnotemark"${12,13}$ вы можете даже проделывать такие
вещи:
\footnotetext"${12}$ {Примечание.}%
\footnotetext"${13}$ {Еще одно.}%
надо просто набрать ...
```

(Это, разумеется, работает только в том случае, если вы знаете, какие номера вам нужны. Если вы в действительности набираете книгу с обилием сносок, да еще и с такими двойными, то вам лучше обратиться к Т_ЕХ_Нологу с просьбой разработать стилевой файл, предоставляющий некую разновидность управляющей последовательности `\doublefootnote` для автоматического выполнения этих вещей.)

В предыдущих частях этой книги нумерация сносок на каждой странице начиналась заново. В этой же главе сноски нумеруются подряд. В стиле `amsrpt` сноски по всей статье нумеруются подряд. Такого рода устройство сносок предпочитают традиционные полиграфисты, потому что первый набор текста делается в виде “гранок” и разбивается на страницы позже, поэтому нельзя заранее знать постраничную нумерацию сносок, пока не будут расставлены номера страниц.

В некоторых журнальных стилях нумерация сносок начинается заново на каждой странице, но это может вызвать известные трудности, поскольку у Т_ЕХ’а такие же точно проблемы со сносками, что и у типографских наборщиков. Хотя Т_ЕХ в явном виде гранок не делает, тем не менее, прежде чем решить, где закончится страница, он должен иметь несколько больше текста, чем помещается на одной странице. Следовательно, Т_ЕХ может решить, что сноска имеет номер 3, тогда как фактически текст будет перенесен на следующую страницу и сноска будет иметь первый номер.

Если вы пользуетесь таким стилем, то можете набрать

```
\adjustfootnotemark{-2}
```

¹⁰Примечание.

¹¹Еще одно.

[сообщения об ошибках]

чтобы получить правильную нумерацию. Но журнальные стили, которые все-рез заботятся о нумерации сносок, должны бы решить эту задачу, используя “двухпроходный” алгоритм: Сначала, когда вы обрабатываете файл, ТрХ сам может написать файл с комментариями относительно того, какие примечания занумерованы неправильно, и затем он может использовать этот файл для получения верной нумерации, когда будет обрабатывать файл вторично.

Кроме аргумента в виде “литеры” для `\footnote` стиль `amsprt` предоставляет возможность воспользоваться “факультативным аргументом”, позволяющим привести в порядок нумерацию сносок. Если набрать

```
\footnote [2]{...}
```

получится примечание, которое воспринимается, как если бы это было второе примечание. Это означает, что в стиле `amsprt` меткой этого примечания будет `2`, тогда как в других стилях это будет `[2]`, или `‡`, и т. п. Такая особая нумерация прерывает стандартную, и как только вы снова воспользуетесь `\footnote`, прерванная нумерация продолжится, как будто `\footnote[...]` не вводилось. Управляющие последовательности `\footnotemark` и `\footnotetext` также имеют “факультативный аргумент”.

СООБЩЕНИЯ ОБ ОШИБКАХ

(См. также проверка синтаксиса.)

В ответе к упр. 4.5 мы видели, какие действия возможны после сообщения об ошибке:

```
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.
```

Нажатие **S** в ответ на сообщение об ошибке аналогично тому, как если бы вы нажали `<carriage-return>` после этого сообщения и после всех последующих. Эти сообщения об ошибках промелькнут на вашем терминале, возможно, слишком быстро, чтобы их можно было прочесть, но они также появятся и в файле протокола. Вы можете также назначить режим `\scrollmode` в вашем файле; это равносильно нажатию клавиши **S** после первого сообщения об ошибке с той разницей, что позже можно будет назначить режим `\errorstopmode`, чтобы вернуться к нормальному режиму.

Нажатие клавиш **R** и **Q** — это еще более крутая мера; по-видимому, вам не стоит об этом беспокоиться, если только вы не прогоняете ТрХ в `batch`-режиме, но тогда все подробности можете узнать у вашего системного программиста.

(Нажатие клавиши **I** позволяет сделать вставку, а **E** — редактировать файл. — *Перев.*)

Имеется также возможность набрать `1, ..., 9`, чтобы ТрХ удалил следующие `1, ..., 9` элементов ввода (элемент (“token”) — это либо литера (например **A**), либо `+`, либо `$`, либо управляющая последовательность, которая рассматривается как один элемент, независимо от того, насколько длинное у нее название). Это средство исправления ошибки требует некоторого навыка и изощренности, но

составные символы]

может быть крайне полезным. Например, предположим, что у нас имеется файл с предложением

Нужно набрать $8=2^3$ в математическом режиме.

После обработки его Т_ЕX'ом получаем сообщение об ошибке вроде такого:

```
! Missing $ inserted.
<inserted text>
$
<to be read again>
```

Нужно набрать $8=2^$

3 в математическом режиме.

?

Теперь Т_ЕX вставит \$ сразу перед ^ (и готов снова читать ^). Литеры '8=2' были уже набраны не в математическом режиме, так что вокруг знака = нет нужных пробелов. Вы не можете сразу это исправить, но и не хотите сразу нажимать (carriage-return) потому что Т_ЕX начнет изображать оставшийся текст как математическую формулу и ошибка вероятно не будет обнаружена до тех пор, пока абзац не закончится, а закрывающий знак \$ так и не появится. С другой стороны, если набрать 2 и потом нажать (carriage-return), то Т_ЕX удалит две литеры, а именно \$, которую он вставил, и ^ из файла, который он собирался снова читать, чем уберет себя от перехода в математический режим.

СОСТАВНЫЕ СИМВОЛЫ

(См. также дроби и им подобные.)

Математики часто любят создавать новые символы, помещая нечто над или под символом, вместо того чтобы пользоваться верхними или нижними индексами. Для этого у $\mathcal{A}\mathcal{M}\mathcal{S}$ -Т_ЕX'a есть средство `\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>	$\overset{s}{X}$ $\underset{A}{A}$

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` всего лишь часть "синтаксиса", а не правая стрелка \rightarrow , которая тоже имеет имя `\to`.

• Упражнение 19.3:

Наберите следующее:

$$\widehat{BC} = \theta, \quad \overline{AB} = \sqrt{2 + 2 \cos \theta} \quad (\text{by the law of cosines}).$$

`\underset` и `\overset` не так тщательно располагают надписи, как это имеет место в случае математических акцентов (см. **акценты в математическом режиме**). К счастью, эти конструкции используются не так уж часто, и вы сами можете приспособиться, немного поэкспериментировав.

$$\text{\$}\overset{\,}{\,},\,s\to\{\underset{A}{\to} X\}\text{\$}$$

$$\overset{s}{X}_A$$

Когда вы применяете `\overset` или `\underset` к бинарной операции или отношению, в результате получается бинарная операция или отношение.¹² Таким образом, $\overset{\text{def}}{\rightarrow}$ и $\overset{\alpha\beta}{\rightarrow}$ будут новыми бинарными отношениями. Разумеется, лучше, если надписи и подписи будут короткими, но со стрелками возможны и длинные надписи и подписи, поскольку Т_ЭХ может делать произвольно длинные стрелки. Однако такие стрелки с надписями или подписями получаются совсем по-другому. См. **стрелки**.

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

$$\text{\$}\overset{\alpha}{\,},\,\alpha\text{ph}\to X_i^j\text{\$}$$

$$\overset{\alpha}{X}_i^j$$

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

$$\text{\$}\overset{+}{\to}=\overset{+}{j}\text{\$}$$

Вы можете улучшить результат, набрав

$$\text{\$}\overset{+}{\to}=\{\overset{+}{j}\}\text{\$}$$

но тогда после символа $\overset{\pm}{\rightarrow}$ появится дополнительный пробел. Если вам когда-нибудь понадобится что-то в этом роде, то обратитесь к помощи Т_ЭХнолога.

Нет необходимости применять `\underset` или `\overset` к формулам типа

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

поскольку для этого существуют `\overbrace` и `\underbrace` (см. **горизонтальные скобки**). Но \mathcal{M}_S -Т_ЭХ также предоставляет вам в качестве альтернативы `\undersetbrace... \to` и `\oversetbrace... \to`:

$$\begin{aligned} \text{\$}\oversetbrace \text{\textit{\$k\$ times}} \to \{x+\dots+x\}\text{\$} & \quad \overbrace{x + \dots + x}^{k \text{ times}} \\ \text{\$}\undersetbrace >\,0 \to\{x+y+z\}\text{\$} & \quad \underbrace{x + y + z}_{> 0} \end{aligned}$$

¹²Если по той или иной причине вы хотите, чтобы результирующая формула воспринималась как отдельный символ, просто заключите ее в фигурные скобки.

список]

Иногда новые символы строятся совсем по-другому, из больших операторов. Например, \sum^* может использоваться в качестве разновидности \sum . Здесь * появляется в качестве верхнего индекса, но при этом другие формулы могут использоваться в качестве “пределов” суммирования:

$$\sum_{x \in A}^* f(x) = \sum_{0 \neq x \in A} f(x)$$

AMS-TeX имеет средство `\sideset` для получения верхних индексов справа и слева:

```
\define\sumstar{\sideset \and^* \to\sum}
$$\sumstar_{x \in A} f(x) = \sum_{0 \neq x \in A} f(x)$$
```

Набрав `\sideset^*\and \to\sum`, вы получите * слева.

СПИСОК

`\roster` часто используется внутри абзаца, так что материал, следующий за `\endroster` не начнется с нового абзаца, если вы не оставите пустую строку или не наберете `\par` после него. Таким образом, `\roster... \endroster` прерывает абзац наподобие выключной формулы. (В противоположность этому, `\endproclaim` и `\enddemo` всегда начинают новый абзац.)

Хотя `\roster` нумерует каждый `\item` автоматически, `\item` может также иметь “факультативный аргумент” (см. сноска). Если вы набираете

```
\item[5] ...
```

с 5 в квадратных скобках, то этот элемент списка получит номер 5 независимо от его положения в списке. Последующие `\item` будут тогда нумероваться начиная с 6 (в отличие от ситуации с `\footnote`, где факультативный аргумент не нарушает нумерацию последующих `\footnote`). Это может быть полезно, например, в том случае, если вы создали `\roster` с четырьмя `\item` и теперь вам нужен другой `\roster` с продолжающей предыдущую нумерацией. Для этого просто представьте первый `\item` второго `\roster` как `\item[5]`.

Разумеется, `\item[5]` на самом деле даст пометку типа (5), или возможно (v) или [e], и т. п. Но вы можете обозначить пометку конкретно, как вам нужно, используя опцию “аргумент в виде литеры” для `\roster` (снова сравните со сносками.) Если вы наберете

```
\item"{\bf 3}"
```

с `{\bf 3}` между парами двойных кавычек ", то элемент списка получит номер 3, точно как набрано.

AMS-TeX также предоставляет вам `\therosteritem` для удобства ссылаться на конкретный `\item` независимо от того, какой стиль вы используете. Если набрать `\therosteritem7`, то получится (7) или (vii) или [g] ..., независимо от стиля.

Некоторые авторы предпочитают первое условие помещать внутри абзаца, и лишь последующие располагают со специальным отступом: Таким образом,

[стрелки

они хотят: (1) Первое условие, которое мы хотим установить, когда используем `\roster` для получения списка,

(2) Второе условие, которое не имеет дополнительной пустой строки перед ним,

(3) Последнее.

за которым следует небольшая отбивка. Чтобы получить это, следует набрать `\Runinitem` вместо `\item` для первого элемента списка.

Когда первый элемент списка трактуется подобным образом, вы можете пожелать, чтобы последующие строки первого элемента списка имели тот же отступ как и все остальные элементы. Это сопровождается некоторыми ухищрениями. Если вы хотите чтобы это произошло, наберите `\Runinitem` непосредственно перед абзацем, в котором появился `\roster\runinitem`. `\Runinitem` будет действовать только на первый `\roster` в этом абзаце. Если вам нужны два подобных списка в одном и том же абзаце, наберите

```
\Runinitem ... \roster\runinitem
...
\endroster\Runinitem
... \roster\runinitem ...\endroster
```

СТРЕЛКИ

AMS-TeX предоставляет общий способ размещения маленьких формул друг над другом для получения новых соотношений вроде $\stackrel{\text{def}}{=}$ (см. **составные символы**), и этот же метод может быть использован для получения новых соотношений типа $\xrightarrow{\alpha\beta}$, которые получаются размещением $\alpha\beta$ над `\longrightarrow`. Но имеется также особый механизм получения более длинных стрелок, которыми бывает необходимо снабжать длинные выражения.

Если набрать в математическом режиме `>>>`, получится правая стрелка \rightarrow , а `<<<` даст левую стрелку \leftarrow . Здесь используются именно тройки `>>>` и `<<<`, потому что если формула помещается между первым и вторым символами, получается выражение над стрелкой, а если между вторым и третьим, то под стрелкой.

```
$>>> \alpha+\beta+\gamma >>>$
```

```
$>>> \alpha > \gamma^{-1} >>>$
```

$$\begin{array}{c} \alpha+\beta+\gamma \\ \xrightarrow{\hspace{1cm}} \\ \leftarrow{\hspace{1cm}} \\ \alpha+\beta+\gamma \\ \alpha \\ \xrightarrow{\hspace{1cm}} \\ \gamma^{-1} \end{array}$$

Конечно, если необходима стрелка вроде $\xrightarrow{x > y}$, снабженная выражением со знаком $>$ или $<$, то лучше заключить это выражение в скобки: `>>>\{x\,>\,\},y>>>`. (Здесь добавлены тонкие шпации, потому что TeX обычно не делает пробелов вокруг бинарных операций в выражениях размера s-size.)

СТЯЖКА

Если набрать `\smash{...}`, то можно убедить TeX, что `'...'` не выдается над строкой и не провисает под ней. Имеются также две разновидности: `\topsmash`

теоретико-множественные обозначения]

и `\botsmash`, в зависимости от того, хотите ли вы, чтобы Т_ЭХ проигнорировал часть текста над строкой, или под ней. Объединив `\smash` и `\phantom`, можно заставить Т_ЭХ думать, что любой символ равнозначен любому другому в смысле расстановки пробелов.

Вот пример того, когда `\smash` применяется для улучшения внешнего вида окончательного продукта. В упр. 10.2 мы набирали

Hölder's Inequality: Let $0 \leq p, q \leq \infty$ with $\frac{1}{p} + \frac{1}{q} = 1$. If $f \in L^p(\mu)$

and $g \in L^q(\mu)$, then $fg \in L^1(\mu)$ and

$$\|fg\|_1 \leq \|f\|_p \|g\|_q.$$

Подобно любому хорошему наборщику, Т_ЭХ вставляет дополнительную отбивку после первой строки, так чтобы дроби $\frac{1}{p}$ и $\frac{1}{q}$ не наложились на что-нибудь из следующей строки. Но это тот самый случай, когда такая уловка дает осечку, так как следующая строка недостаточно длинная, и возникает впечатление излишнего пустого пространства. Если мы набираем

`\$ \botsmash{\dfrac{1}{p}} + \botsmash{\dfrac{1}{q}} = 1 \$`

то Т_ЭХ проигнорирует тот объем, который остается под строкой из-за дробей, и мы получим

Hölder's Inequality: Let $0 \leq p, q \leq \infty$ with $\frac{1}{p} + \frac{1}{q} = 1$. If $f \in L^p(\mu)$

and $g \in L^q(\mu)$, then $fg \in L^1(\mu)$ and

$$\|fg\|_1 \leq \|f\|_p \|g\|_q.$$

См. упр. 19.2 в статье **горизонтальные скобки** относительно иных способов использования `\smash`.

ТЕКСТ В ФОРМУЛАХ

См. абзацы.

ТЕОРЕТИКО-МНОЖЕСТВЕННЫЕ ОБОЗНАЧЕНИЯ

Простые формулы, вроде $\{a, b, c\}$, набираются обычным образом:

`\${a,b,c}\$` $\{a, b, c\}$
`\${1,2,\dots,n}\$` $\{1, 2, \dots, n\}$

Но когда вам встретятся вертикальная черта `|` или двоеточие `:` посреди такой формулы, то они лучше смотрятся с тонкими шпациями с внутренней стороны фигурных скобок и с `\mid` вместо `|` (или `\vert`), который автоматически проставляет пробелы вокруг `|` (вокруг двоеточия пробелы устанавливаются автоматически, так что вам незачем об этом беспокоиться):

`\${\,x\mid x>5\,}\$` $\{x \mid x > 5\}$
`\${\,x:x>5\,}\$` $\{x : x > 5\}$

Когда используются ограничители `\bigl` и `\bigr` больших размеров, как в

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

то они должны сопровождаться (с.м. **большие ограничители**) `\bigr|` для изображения |:

```
$$\bigl\{\, \bigr(x, f(x)\bigr)\bigr|x \in d\, \bigr\}$$
```

ТОЧКИ

У `AMS-TeX`'а есть `\ldots` и `\cdots` для получения точек на строке (...) и точек посередине (...), но обычно вам достаточно просто набрать `\dots`, и нужные точки появятся чудесным образом. Однако иногда бывает нужно знать немного больше относительно того, как работает `\dots`, чтобы иметь возможность ими манипулировать.

`\dots` работает как управляющая последовательность с одним аргументом — на основе следующего символа в формуле решает, какого рода точки надо использовать. Это несколько проблематично, потому что правила расположения точек на строке и точек посередине не всегда одни и те же: некоторые наборщики любят набирать формулы типа a_1, \dots, a_n и $a_1 + \dots + a_n$ (ужас!). Так что на самом деле `\dots` работает одним из четырех способов:

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

Например, когда вы набираете `'\dots'` в математическом режиме, `AMS-TeX` транслирует это в `'\dotsc'`, и тот стиль, который вы используете, определяет, какими эти `\dotsc` должны быть — в этом учебнике и в стиле `amsppt` получатся точки на строке, но в каком-нибудь ином стиле `\dotsc` могут дать точки посередине строки.

Таким образом, когда вы используете `\dots` в математическом режиме, от стиля и следующего символа формулы зависит конкретный вид точек. К сожалению, существует одна ситуация, в которой такая схема порочна, а именно, когда нет следующего символа. Если вы набираете

```
Consider the infinite sequence $a_1, a_2, a_3, \dots$.
Let $a_1 + a_2 + a_3 + \dots$ be its sum, and
let $a_1 a_2 a_3 \dots$ be its product.
```

то нельзя рассчитывать, что `\dots` знает, какого рода точки вам нужны — он может только заглядывать вперед, и у него нет способа узнать, какой символ был набран ранее — и во всех этих случаях `\dots` просто выбирает `\dotso`. Так что если формула заканчивается точками, самым правильным будет сказать `AMS-TeX`'у, какого именно вида точки нужны:

```
Consider the infinite sequence $a_1, a_2, a_3, \dotsc$.
Let $a_1 + a_2 + a_3 + \dotscb$ be its sum, and
let $a_1 a_2 a_3 \dotso$ be its product.
```

Если вы будете помнить, что в конце формул нужно явно указывать, какие именно точки нужны, то в 99% случаев будете получать точки нужного вида.

точки]

◇ Есть одна ситуация, в которой даже вполне здравомыслящие люди могут не сойтись во мнениях относительно расположения точек. Рассмотрим предложение

The quadrilateral $A_1A_2A_3A_4$ clearly has area $x_1x_2x_3x_4$, and in general the n -gon $A_1A_2A_3\dots A_n$ has area $x_1x_2x_3\dots x_n$.

В формуле $A_1A_2A_3A_4$ расположение букв A_1 , A_2 , A_3 , и A_4 не несет особой смысловой нагрузки — они 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 $x_1 \times x_2 \times x_3 \times x_4$ или $x_1 \cdot x_2 \cdot x_3 \cdot x_4$. Аналогично, в последней формуле этого предложения расположение символов x_1 , x_2 , ... представляет собой бинарную операцию умножения, которую некоторые авторы¹³ 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 `\dotso` вместо `\dotso`, но $\mathcal{A}\mathcal{M}\mathcal{S}\text{-}\mathcal{T}\mathcal{E}\mathcal{X}$ не имеет возможности понять, имеется ли в виду умножение, так что от `\dotso` не следует ожидать, что он сам справится с ситуациями, подобными рассмотренной. Однако $\mathcal{A}\mathcal{M}\mathcal{S}\text{-}\mathcal{T}\mathcal{E}\mathcal{X}$ может предоставить в помощь 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 `\dotso` или в `\dotso`.

Тем не менее вы можете вставить `\dotso`, если набираете что-то вроде

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

потому что перед бинарной операцией `\times` команда `\dotso` автоматически дает `\dotso`. Кроме того, $\mathcal{A}\mathcal{M}\mathcal{S}\text{-}\mathcal{T}\mathcal{E}\mathcal{X}$ намеренно делает одно крошечное исключение из того правила, что `\dotso`, за которым следует бинарная операция, дает `\dotso`:

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

(относительно “толстой шпации” `\;`; см. горизонтальные пробелы).

• Упражнение 19.4:

Наберите следующее соотношение:

$$\begin{aligned} G(z) &= e^{\ln G(z)} = \exp\left(\sum_{k \geq 1} \frac{S_k z^k}{k}\right) = \prod_{k \geq 1} e^{S_k z^k / k} \\ &= \left(1 + S_1 z + \frac{S_1^2 z^2}{2!} + \dots\right) \left(1 + \frac{S_2 z^2}{2} + \frac{S_2^2 z^4}{2^2 \cdot 2!} + \dots\right) \dots \\ &= \sum_{m \geq 0} \left(\sum_{\substack{k_1, k_2, \dots, k_m \geq 0 \\ k_1 + 2k_2 + \dots + mk_m = m}} \frac{S_1^{k_1}}{1^{k_1} k_1!} \frac{S_2^{k_2}}{2^{k_2} k_2!} \dots \frac{S_m^{k_m}}{m^{k_m} k_m!} \right) z^m \end{aligned}$$

¹³В том числе и автор этого учебника, но не автор самого $\mathcal{T}\mathcal{E}\mathcal{X}$ ’а.

♦ `\dots` знает, как обращаться с математическими символами, уже имеющимися в АМ_S-ТЭХ, подобными `\alpha`, `\iint`, `\cap`, `<` и `\leq` (`\dots` также выбирает `\dotscb`, когда далее следует `\not`, поскольку `\not` используется преимущественно для “отрицания” следующего за ним бинарного отношения.) Но что делать с новыми символами, которые вы определяете, `\dots`, вероятно, не знает. Например, если в статье часто встречается символ $\int_{-\infty}^{\infty}$, очень удобно определить

```
\define\Int{\int_{-\infty}^{\infty}}
```

Но если вы набираете

```
\Int\dots\Int
```

то не сможете получить `\dotsi` между `\Int`, а просто получите `\dotso`. Чтобы разобраться с этим вопросом, АМ_S-ТЭХ предоставляет две другие управляющие последовательности `\DOTSI` и `\DOTSB`; эти управляющие последовательности не выводят никаких символов на печать, но появившись в начале определения, предупреждают `\dots` о том, что нужно будет использовать `\dotsi` или `\dotscb`. Таким образом, можно набрать

```
\define\Int{\DOTSI\int_{-\infty}^{\infty}}
```

и тогда получится `$$\Int\dots\Int$$` — именно так, как нужно.

Имеется еще одна тонкость, связанная с `\dots`, о которой вы можете захотеть узнать (обещаю — это последняя). Когда в конце формулы возникает `\dots` или какие-либо специфические точки, `\dotsc`, ..., АМ_S-ТЭХ автоматически вставляет дополнительную тонкую шпацию, поскольку это лучше смотрится. Когда за одной из этих управляющих последовательностей следует правый ограничитель типа `)` или `\bigger` и т. д., то также автоматически вставляется тонкая шпация. Опять же, когда вы определяете какой-нибудь новый правый ограничитель, нужно в `\define` включать `\dots`. Например, в предыдущей главе говорилось, что автор любит определять

```
\define\{\left{}
\define\}\right)}
```

Но на самом деле `\}` было определено как

```
\define\}\DOTSX\right)}
```

Управляющая последовательность `\DOTSX` не порождает печатный символ, но даст команду `\dots` или `\dotsc` или ... вставить дополнительную тонкую шпацию.

ФАЙЛЫ

У ТЭХ'a есть управляющая последовательность `\input`, о которой мы упоминали только в одном контексте: если вы не пользуетесь специальной программой `amstex`, ваш файл должен начинаться строкой `\input amstex`, чтобы сообщить ТЭХ'у, что надо читать файл `amstex.tex`. На самом же деле `\input` можно использовать и в других случаях. Например, если у вас есть файл `defs.tex` с часто используемыми `\define`, нет никакой необходимости переписывать его в начало другого файла `paper.tex`, который вы обрабатываете. Вместо этого можно

фантомы]

просто набрать `\input defs` в начале `paper.tex`, и тогда ТрХ прочтет `defs.tex`, как только начнет обрабатывать `paper.tex`. Это обычно несколько упрощает управление файлом: каждый файл будет избавлен от нагромождения `\define` в начале, и если вы решите сделать какие-то изменения в ваших `\define`, вы можете сделать это только в `defs.tex`, а не в каждом файле, где используются эти `\define`. С другой стороны, вы не можете при такой организации прогнать ваш файл через ТрХ, не имея поблизости файла `defs.tex`. Так что если вы посылаете на другой компьютер свой файл, следует не забыть и о файле `defs.tex`.

У вас могут быть файлы вроде `defs.1`, `defs.2`, ..., с иными расширениями, чем `.tex`, но тогда следует набирать `\input defs.1`, `\input defs.2`, ...

Когда вы имеете дело с большой рукописью, `\input` также может быть полезен. Например, этот учебник составлен из нескольких файлов: `jot1.tex`, `jot2.tex`, ..., каждый из которых сначала был обработан отдельно (со своей для каждой главы нумерацией страниц). Файл `jot.tex`, который дает окончательный вариант книги, не содержит почти ничего, кроме `\input jot1`, `\input jot2`, ...

ФАНТОМЫ

Иногда бывает нужен пробел, не такой как `\quad`, или `\smallvspace`, или ..., но просто пробел, “размером с эту формулу”. При помощи средства `\phantom` с той или иной строкой можно проделывать чудеса. Если набрать `` (в математическом режиме или вне его), то вы получите пробел точно такого же размера, как и ‘...’. Если нужно, чтобы новый, еще не определенный символ был того же размера, что и X , очень удобно воспользоваться ``. (Но если вам нужно что-то размера \sum , и действие этого совпадает с действием большого оператора, обратитесь за помощью к ТрХнологу.)

Если набрать `\hphantom{...}`, то получится “горизонтальный фантом”, ширина которого точно совпадает с ‘...’ а высота — нет. Так что это эффективно действует как пробел требуемой ширины. Например, управляющую последовательность `\dwidth`, упомянутую в ответе к упр. 17.5, можно определить следующим образом:

```
\define\dwidth{\hphantom0}
```

• Упражнение 19.5:

Объясните, как набрать следующую выключную формулу:

$$m' = m_1 + 2n_1 = 3m + 4n,$$

$$n' = m_1 + n_1 = 2m + 3n.$$

• Упражнение 19.6:

Наберите такую выключную формулу:

$$f(x) = \sum_{n=1}^{\infty} a_n x^{n-1} = 1 + x + 2x^2 + 3x^3 + \dots,$$

$$xf(x) = \sum_{n=1}^{\infty} a_n x^n = x + x^2 + 2x^3 + \dots,$$

$$x^2 f(x) = \sum_{n=1}^{\infty} a_n x^{n+1} = x^2 + x^3 + \dots.$$

Имеется также `\vphantom{...}`, который дает пустое пространство без ширины, но с тем же просветом над (и под строкой), что и `'...'`. Например, `\mathstrut` (см. **распорки**) не что иное, как `\vphantom{.}`. Можно заставить Т_ЭX думать, что формула имеет тот размер, который вам хочется, если вы объедините `\phantom` с различными видами `\smash` (см. **стяжка**).

ФИГУРНЫЕ СКОБКИ

См. теоретико-множественные обозначения.

ФИГУРНЫЕ СКОБКИ НАД И ПОД ФОРМУЛАМИ

См. горизонтальные фигурные скобки

ФИГУРНЫЕ СКОБКИ СВЕРХУ

См. горизонтальные фигурные скобки.

ФИГУРНЫЕ СКОБКИ СНИЗУ

См. горизонтальные фигурные скобки.

ФОРМУЛЫ В РАМОЧКЕ

Формулу можно заключить в рамочку при помощи управляющей последовательности `\boxed`. Например,

`$$\boxed{1+1=2}$$`

дает

$$1 + 1 = 2$$

В журнальных статьях почти никогда не бывает формул в рамочке, но в книгах они иногда появляются. (Уровень полиграфического исполнения находится в обратном пропорциональной зависимости от уровня излагаемого материала.)

Взятая в рамочку формула представляет собой просто новый математический символ, так что он может быть использован внутри другой формулы и даже может быть взят в рамочку. Таким образом,

`$$\text{Fundamental result:}\quad\boxed{\boxed{1+1=2}}$$`

дает

Fundamental result:

$$1 + 1 = 2.$$

шрифты в математическом режиме]

Кроме того, формула, обработанная `\boxed`, может иметь метку `\tag`. Например,

```
$$\boxed{1+1=2}\tag 3--1$$
```

дает

(3-1) 1 + 1 = 2

Но не пытайтесь помещать `\tag` внутрь `\boxed` — вы только получите таинственное сообщение об ошибке.

ЧЕРНЫЕ ПРЯМОУГОЛЬНИКИ

Вы можете заставить Трех прекратить печатать черные прямоугольники после каждого `Overfull \hbox`, набрав `\NoBlackBoxes` (как поступил автор с формулой на с. 58, чтобы она вышла за поле). Это бывает полезно, когда мы имеем дело с выключной формулой, которая шире лишь на один-два пункта (см. **единицы измерения**) — если над и под формулой много чистого пространства, дополнительная ширина будет, вероятно, незаметна, и можно позволить себе избежать переноса формулы на другую строку.

Это “глобальная” команда — она действует на весь последующий материал, даже если она заключена в фигурные скобки или в знаки \$. Имеется также средство `\BlackBoxes` для возврата черных прямоугольников, но на самом деле в одном файле оба эти средства не следует употреблять. Только убедитесь, что вы не использовали `\NoBlackBoxes` до самого последнего этапа, когда вы уверены, что все имеющиеся `Overfull \hbox` именно там, где нужно.

ЧИСЛА, СТАРИННОЕ НАПИСАНИЕ

Если набрать `\oldnos{5,283.06}`, то получится 5, 283.06. Можно также использовать `\oldnos` в математическом режиме; числа, стоящие в индексах, будут иметь правильный размер, и хотя тонкая шпация в математическом режиме обычно вставляется после запятой, внутри `\oldnos` никаких пробелов после запятых не будет.

ШРИФТЫ В МАТЕМАТИЧЕСКОМ РЕЖИМЕ

Математики часто используют в своих формулах полужирный шрифт, например $ax + by^n$. В *AMS-TeX*’е это набирается как

```
$a\bold x + b\bold y^n$
```

Обратите внимание, что `\bold` в математическом режиме работает совсем по-другому, чем `\bf` в тексте: это управляющая последовательность с аргументом, а не команда изменить шрифт. Лучше всего это выражается в математических формулах, потому что полужирные буквы часто перемежаются с обычными курсивными. Буквы, к которым был применен `\bold`, автоматически правильно ставят размер, так что `$x^{\bold y}$` дает x^y .

Если в статье, которую вы набираете, полно x , y и z , вам, по-видимому, захочется ввести определения

```
\define\x{\bold x}
\define\y{\bold y}
\define\z{\bold z}
```

[шрифты в математическом режиме

и тогда выражение $ax + by + cz$ можно набрать так:

```
$a\bold{x}+b\bold{y}+c\bold{z}$
```

Может появиться искушение набрать

```
$a\bold{x}+b\bold{xy}+c\bold{xyz}$
```

но результат бывает совершенно непредсказуемым. Например, `\bold{ff}` даст на самом деле лигатуру ff! Во всяком случае набор вида `\bold{xy}` считается плохим стилем. Если в статье лишь несколько полужирных букв, не так уж обидно набрать при необходимости `\bold{x}\bold{y}`, но если полужирных букв слишком много, то следует создать новые управляющие последовательности вроде `\x`, `\y` и `\z`, которые облегчат ваш труд.

Кроме `\bold` иногда может оказаться полезным `\goman`, по-видимому не так уж нужен `\slanted`, и наконец есть `\italic`, позволяющий получать обычный курсив, используемый в тексте, в противоположность используемому в формулах. Но буквы, к которым были применены `\slanted` и `\italic`, не меняют размера в индексах.¹⁴

`\bold` следует применять только к буквам. Если набрать `\bold{+}` и `\bold{<}`, то получатся обычные знаки `+` и `<`, которые, как правило, и ставятся в формулах с полужирными буквами, вроде $x < y + z$. Но если вам действительно необходимы полужирные математические символы, АМ_S-Т_EX предоставит их в ваше распоряжение, если в начале своего файла после строки `\input amstex` вы наберете `\loadbold`. Для двух видов символов используются различные управляющие последовательности:

<code>\boldkey</code>	для символов, имеющих на клавиатуре, типа <code>></code> или <code>=</code>
<code>\boldsymbol</code>	для символов, которые вводятся одной управляющей последовательностью, например, <code>\Gamma</code> или <code>\gtrapprox</code> .

Например,

```
$\bold{x} \boldkey+ \boldsymbol{\Omega}$
```

даст

$$x + \Omega$$

Точнее говоря, `\boldkey` в математических формулах можно использовать в следующих комбинациях:

- С любым из символов

`+ - = < > () [] | / * . , : ; ! ?`

¹⁴Иногда по всей книге формулы могут быть прямыми или полужирными или еще какими-нибудь столь же непривычными. Это, по всей видимости, вопрос стиля — кто-то может сам решить, использовать ли прямые или полужирные буквы вместо обычных математических курсивных. Если это именно такой случай, вам не следует снова и снова указывать `\goman` или `\bold`. Разработчик стиля может устроить все так, что Т_EX будет автоматически выбирать нужный шрифт в математическом режиме.

шрифты в математическом режиме]

вы получите

$$+ - = < > () [] | / * . , : ; ! ?$$

Использование с любым из перечисленных символов `\bold` не даст полужирного начертания, т. е. в результате `\bold +$` получится обычный `+`, и т. д.

Полученные в результате применения `\boldkey` символы `+` и `-` представляют собой бинарные операции, такие же как обычные `+` и `-`, полужирный `=` будет бинарным соотношением, соответствующим обычному `=`, и т. д.

- С буквами:

$$\begin{array}{ll} \text{\boldkey a\$}, \dots, \text{\boldkey z\$} & a, \dots, z \\ \text{\boldkey A\$}, \dots, \text{\boldkey Z\$} & A, \dots, Z \end{array}$$

Обратите внимание, что эти буквы набраны шрифтом *bold math italic*; они отличаются от прямых полужирных букв `a, ..., z, A, ..., Z`, которые получаются посредством `\bold` в математическом режиме.

- С цифрами:

$$\text{\boldkey 0\$}, \dots, \text{\boldkey 9\$} \quad 0, \dots, 9$$

Однако при помощи `\bold 0$, ..., \bold 9$` получатся точно такие же цифры.

Конструкцию `\boldsymbol` можно использовать в следующих комбинациях:

- Со строчными и прописными греческими буквами:

$$\begin{array}{ll} \text{\boldsymbol\Gamma\$}, \dots, \text{\boldsymbol\Omega\$} & \Gamma, \dots, \Omega \\ \text{\boldsymbol\varGamma\$}, \dots, \text{\boldsymbol\varOmega\$} & \Gamma, \dots, \Omega \\ \text{\boldsymbol\alpha\$}, \dots, \text{\boldsymbol\omega\$} & \alpha, \dots, \omega \end{array}$$

- Из соображений удобства за `\boldsymbol` может следовать обычная буква (но не цифра или иная литера), что даст тот же результат, что и `\boldkey`.

- `\boldsymbol` можно применять также ко всем иным стандартным символам, которые представляются управляющими последовательностями. Например, для получения жирного штриха:

$$\begin{array}{ll} \text{\boldsymbol\prime\$} & / \\ \text{\boldsymbol A^{\boldsymbol\prime}\$} & A' \end{array}$$

(Команда `\boldsymbol\prime`, использующая краткое обозначение `\prime`, не работает.)

- `\boldsymbol` можно применять и к ограничителям:

$$\begin{array}{ll} \text{\boldsymbol\{...\boldsymbol\}\$} & \{...\} \\ \text{\boldsymbol\langle...\boldsymbol\rangle\$} & \langle...\rangle \\ \text{\boldsymbol|, \boldkey|, \boldsymbol|\$} & |, |, |, | \\ \text{\boldsymbol\vert, \boldsymbol\vert, \boldsymbol\Vert, \boldsymbol\| \$} & |, |, |, | \end{array}$$

(Обратите внимание, что для `|` следует использовать `\boldkey`.)

После `\left` и `\right`, однако, `\boldsymbol` не применяется. В частности, набрав `\left\boldsymbol| ... \right\boldsymbol|`, вы получите только сообщение об ошибке.

[шрифты в математическом режиме

- Некоторые символы нельзя получить в полужирном начертании ни с помощью `\boldkey`, ни с помощью `\boldsymbol`. Это относится к полужирному начертанию “рукописных букв” A, \dots, Z , которые набираются как `\Cal A, \dots, \Cal Z`, и полужирному начертанию “старинных” цифр $0, \dots, 9$, которые получаются посредством `\oldnos`. Если они вам все-таки очень нужны, то обратитесь за помощью к Трехнологу или воспользуйтесь `\rmb`. См. жирный шрифт для бедных.

Как уже упоминалось, имеется специальное семейство “рукописных” букв: A, \dots, Z , которые получаются посредством `\Cal A, \dots, \Cal Z`. Прописная буква C в `\Cal` напоминает вам о том, что бывают только прописные рукописные буквы (для единственной строчной буквы ℓ используется `\elll`).

В настоящее время у `AMS-TeX` есть и более экзотический шрифт `\frak` — готический, который предоставляется в ваше распоряжение управляющей последовательностью `\loadelf`, помещенной в начале файла после строки `\input amstex`. Если вы пользуетесь стилем `\amsppt`, то этот шрифт подключается автоматически, так что нет необходимости особо это оговаривать. Итак, набрав `\frak g`, вы получите g а `\frak G` и `\frak S` дадут \mathfrak{G} и \mathfrak{S} (отличить которые смогут разве что \mathfrak{Goethe} и $\mathfrak{Schiller}$). Этот шрифт математики обычно называют “готическим”, поэтому `AMS-TeX` использует `\goth` в качестве альтернативного названия для `\frak` (что довольно удобно, если вы не хотите беспокоиться о случайной путанице между `\frac` и `\frak`).

`AMS-TeX` в качестве уступки неудачным, но принятым обозначениям, предоставляет также еще один шрифт. Чтобы выделить полужирный шрифт, математики мелом на досках обычно рисуют буквы вроде \mathbb{R} (и в своих рукописях тоже). Некоторые из них предпочитают видеть такие же буквы и на печати, поэтому `AMS-TeX` предлагает ажурный шрифт, или “жирный мелом на доске” (`blackboard bold`) `\Bbb`. Подобно `\Cal`, он применим только к прописным буквам. В стиле `amsppt` вы автоматически получаете доступ к шрифту `\Bbb`, в других стилях достаточно набрать `\loadmsbm` после строки `\input amstex`.

Даже если акценты ставятся над полужирными буквами, обычно используются светлые акценты:

$$\hat{\boldsymbol{x}}$$

Но если есть необходимость получить $\hat{\boldsymbol{x}}$, следует просто набрать `\bold{\hat{x}}` — это единственный случай, когда `\bold` удобнее применять к целой группе. Большинство специальных математических шрифтов не имеют своих собственных акцентов, так что в результате получится обычный акцент. Например, `\Cal{\hat{A}}` даст просто \hat{A} , так как шрифт `\Cal` не имеет своего акцента `\hat`.

ПРИЛОЖЕНИЯ

Приложение А. Стиль AMS для журнальных статей

Обычно при работе с AMS-TeX ’ом элементы форматирования того или иного вида публикаций определяются “стилевым файлом”. Описываемые далее элементы форматирования представляют собой составную часть стиля “препринт”, разработанного Американским математическим обществом (AMS — American Mathematical Society). Этот стиль разрабатывался специально для *Journal of the American Mathematical Society*, полиграфические особенности которого он наиболее полно отражает.

ПОДГОТОВИТЕЛЬНЫЙ ЭТАП

Прежде чем приступить к набору статьи, необходимо обычно задать некоторую информацию: длину строки, длину полосы набора; определиться, нужна ли колонцифра или колонтитул; и даже решить, будет ли файл содержать статью или монографию. Если вы намереваетесь строго придерживаться стиля AMS для журнальных статей, то вся эта информация автоматически будет задана управляющей последовательностью `\documentstyle{amsprpt}`.

Стиль `amsprpt` задает длину строки равной 30 pica, а длину полосы набора — 47 pica. Предполагается, что файл содержит статью, а не главу книги (в противном случае следует набрать `\Monograph`), и каждая полоса, за исключением первой, будет снабжена колонтитулом: на четных полосах будет проставлен номер страницы и текст, который был введен посредством `\title ... \endtitle`, а на нечетных — номер страницы и текст из `\author ... \endauthor`.

Некоторые команды простирают свое действие на весь документ. Они должны быть изменены в начале файла, сразу после `\documentstyle` и перед `\topmatter`. Эту часть файла называют “преамбулой”. В преамбулу входят: `\TagsOnRight`, `\NoPageNumbers`, `\NoRunningHeads`, `\Monograph`, `\pagewidth`, `\pageheight`, `\pageno`, `\define`, а также команды загрузки шрифтов. (Если вы используете стиль “препринт” для файла, который в дальнейшем намереваетесь переслать в AMS для электронного воспроизведения, не переопределяйте никакие команды TeX ’а или AMS-TeX ’а — это может вызвать известные трудности при издании ваших работ в AMS).

Преамбула предваряет любой документ, независимо от того, статья это или монография.

Например, вы можете изменить длину строки, задаваемую стилем `amsrpt`, набрав в преамбуле `\pagewidth{(dimen)}`. Точно так же можно пререклеститься на другую длину полосы при помощи `\pageheight{(dimen)}` набранной перед строкой `\topmatter`.

Можно со всех полос документа удалить колонцифру и (или) колонтитул, если в преамбулу соответственно включить `\NoRunningHeads` и `\NoPageNumbers`. Если же колонтитул нужен, но следует изменить его содержание, см. раздел “колонтитулы” в гл. 19.

Если вы готовите не статью, а главу книги, то наберите `\Monograph`, чтобы установить соответствующие книге параметры. (Более полно об использовании стиля `amsrpt` для подготовки монографий см. с. 191).

СВЕДЕНИЯ ОБ ИЗДАНИИ

Статья должна начинаться с элегантно оформленного названия, расположенного так, чтобы привлечь внимание читателя, за которым следует автор(ы) в таком виде, чтобы его (их) можно было заметить. `AMS-TeX` специфицирует название статьи конструкцией `\title... \endtitle`. Например, в стиле `amsrpt` ввод

```
\title Treemunch's Tribulations \\ Strange
Names\\ And People
\endtitle
```

даст на выходе

Treemunch's Tribulations Strange Names And People

Обратите внимание, что `\\` разбивает название на отдельные строки; `AMS-TeX` не обязан разбивать для вас название на отдельные строки, поскольку для этого необходимо эстетическое решение (включая такие вещи, как прописная буква в слове “And”). Нет необходимости заключать отдельные строки в фигурные скобки, так как `TeX` знает, что первая строка всегда находится между `\title` и первым `\\`, вторая — всегда между `\\` и следующим `\\`, а третья — это все вплоть до `\endtitle` (после последней строки `\\` не требуется). Во всех конструкциях `AMS-TeX`'а такого рода, где переход на другую строку регламентируется наборщиком, используется специальная управляющая последовательность `\\` (в основном в тех случаях, когда все строки центрируются по отдельности, а не объединены в один абзац).

Нет необходимости набирать название прописными буквами — это делается автоматически. Однако, если вы предпочитаете, чтобы каждое слово названия было набрано с прописных букв, а далее строчными, то воспользуйтесь опцией “`nofrills`”, набрав

```
\title\nofrills ... \endtitle
```

⊠ *AMS-TeX* также позаботился о том, чтобы не обращать внимания на дополнительные пробелы в конструкциях типа `\author` и `\title`. Несмотря на то, что было набрано

```
\title Treemunch's Tribulations \ Strange
...
```

эти строки воспринимаются как 'Treemunch's Tribulations' и 'Strange Names', а не 'Treemunch's Tribulations,' и '\ Strange Names'. Так что не следует беспокоиться о том, что случайно пробравшиеся внутрь дополнительные пробелы могут мешать правильному центрированию каждой строки.

Авторство устанавливается конструкцией `\author ... \endauthor`. Разрыв строк (`\`) здесь также допускается, чтобы примирить между собой тройку, четверку и даже большее чило капризных расположений.

И `\title... \endtitle`, и `\author... \endauthor` должны появляться внутри особой области `\topmatter... \endtopmatter`, которая используется для выделения всех сведений об издании (они могут появиться в конце титульной страницы или даже в конце всей работы). Таким образом, начало *AMS-TeX*-файла должно выглядеть следующим образом:

```
\input amstex
\documentstyle{amspt}
< команды из преамбулы типа \define,
\NoRunningHeads, \Monograph, и т. п.>
\topmatter
{
\title... \endtitle
\author... \endauthor
\affil... \endaffil
\address... \endaddress
\email... \endemail
\date... \enddate
\dedicatory... \enddedicatory
\thanks... \endthanks
\translator... \endtranslator
\keywords... \endkeywords
\subjclass... \endsubjclass
\abstract... \endabstract
}
\endtopmatter
\document
```

Хотя все управляющие последовательности, стоящие в фигурных скобках { }, являются факультативными, издателям следует определиться с тем, как именно должно задаваться то или иное вхождение в `\topmatter`. Американское математическое общество, например, требует, чтобы в материалах для публикации были представлены заголовок (`\title`), автор (`\author`), адрес (`\address`), тематика (`\subjclass`) и для публикации в журнале — аннотация (`\abstract`). Они могут набираться в любом порядке, но обязательно между `\documentstyle{amspt}` и `\document`. Синтаксис всех управляющих последовательностей из `\topmatter` обязательно имеет начальный и конечный элемент.

Если вы забудете поставить ограничители `\topmatter... \endtopmatter`, весь этот материал попросту исчезнет. Если вы забудете любой другой элемент с `\end...`, появится сообщение об ошибке в `\endtopmatter`, а может и раньше. Если это были `\endtitle`, `\endkeywords` или любой другой `\end...`, имеющие `\nofrills`, то появится следующее сообщение об ошибке: “! Paragraph ended before `\next@` was complete.”

Такие вещи, как `\title`, `\author`, `\affil(iation)` и `\dedicatory` являются просто подзаголовками, для разбивки которых на отдельные строки прибегают к `\\`. В других составляющих `\topmatter`, которые оформляются в виде абзацев, разбивка на отдельные строки достигается посредством `\linebreak`.

В стиле `amsprt` управляющая последовательность `\affil` находится непосредственно под фамилией автора. В большинстве журнальных стилей (в том числе и в журналах AMS) `\affil`, по всей видимости, будет опускаться, но зато будет использоваться `\address... \endaddress` для указания более полного адреса, который зачастую будет помещаться в самом конце статьи. Стиль `amsprt` также помещает адреса в конце. В `\address` не допускается вводить `//` для указания особых разбивок на строки. Следует просто набрать `\address`(полный адрес)`\endaddress`, и `AMS-TEX` воспримет это как обычную строку, в которой допустимы обычные разбивки. Вы можете использовать `\address` более одного раза для разных адресов разных авторов: убедитесь только, что вы набрали их в том же порядке, что и самих авторов.

Электронный адрес теперь становится существенной компонентой адреса автора, так что стиль `amsprt` предоставляет конструкцию `\email... \endemail`. Можно использовать несколько электронных адресов, но каждая из конструкций `\email... \endemail` должна сочетаться с соответствующей `\address`. Иными словами, каждый электронный адрес `\email` должен следовать за обычным адресом `\address`. Электронный адрес будет напечатан в конце статьи в виде “*E-mail*: (сетевой адрес)” сразу за соответствующим ему обычным адресом. Если предшествующий ему `\address` отсутствует, то электронный адрес не будет напечатан.

Элемент `\date... \enddate` представляет собой другой кусок информации, который может быть удален или изменен в зависимости от журнала, который может иметь, по-видимому, свои собственные выходные данные. В стиле `amsprt` можно использовать `\date` для предоставления приоритета тому или иному элементу.

Посвящение `\dedicatory... \enddedicatory` применяются для таких вещей, как “Посвящается профессору Е. Johnson в связи с Е восьмидесятилетием.” Посвящение будет напечатано по центру курсивом непосредственно перед аннотацией.

Благодарность за финансовую помощь, независимо от того благодарны вы или нет, выражается при помощи `\thanks... \endthanks`. В стиле `amsprt` благодарности помещаются на первой странице на месте подстрочного примечания. В отличие от `\address` и `\email` опция `\thanks` не может более одного раза появляться в `\topmatter`. Иногда, однако, целесообразно поблагодарить за финансовую поддержку отдельно от имени каждого из соавторов статьи. Поскольку `\thanks` оформляется как нумерованное подстрочное примечание, каждую из благодарностей можно оформить в виде отдельного абзаца в одном примечании.

Дополнительные абзацы, однако, нельзя просто указать посредством `\par` или оставив пустую строку — это вызовет сообщение об ошибке. Поэтому нужно внутри текста примечания в начале нового абзаца набирать `\endgraf`.

Для статей, у которых наряду с автором есть и переводчик, информация об этом в стиле `\amsrpt` задается посредством `\translator... \endtranslator`. Эти сведения будут напечатаны в конце статьи восьмым кеглем прямого начертания в виде “Translated by” и следующей за этим фамилией переводчика, набранной прописными буквами.

Ключевые слова и предметная классификация AMS для вашей статьи оформляются как `\keywords... \endkeywords` и `\subjclass... \endsubjclass`. В стиле `\amsrpt` это просто распечатывается в конце первой страницы статьи в виде нумерованной сноски. Пометка “Keywords.” автоматически появляется перед словами, которые набираются в качестве ключевых, а пометка “1980 Mathematics Subject Classification. (1985 Revision)” появляется перед предметной спецификацией. В разных журналах могут использоваться разные пометки, но в любом случае они будут появляться автоматически, так что их не следует набирать.

В тексте главы книги все, что набрано между `\keywords... \endkeywords` и `\subjclass... \endsubjclass` вовсе не пропечатается, поскольку считается, что это обрабатывается отдельно, как часть материала, выносимого на оборот титульной полосы книги.

Аннотация статьи набирается при помощи `\abstract... \endabstract` восьмым кеглем и оформляется с втяжками от левого и правого поля на 2 пайки. Поскольку в журнале *Journal of American Mathematical Society*, стиль которого отражает `\amsrpt`, аннотации предшествует слово “ABSTRACT.”, оно автоматически будет проставлено в этом стиле перед текстом аннотации, так что набирать его не следует. Если вы не хотите, чтобы это слово было в вашей аннотации, наберите `\nofrills` сразу после `\abstract`. Например, вы считаете, что у вас *Résumé*, а не *Abstract*. Тогда нужно поступить так:

```
\abstract\nofrills R'ESUM'E. Текст
резюме. \endabstract
```

Вы получите:

RÉSUMÉ. Текст резюме.

Обратите внимание, что если вы примените `\nofrills`, будет использован обычный прямой шрифт восьмого кегля, если вы специально не оговорите иного.

Иногда статья сопровождается содержанием, которое тоже набирается в разделе `\topmatter` посредством `\toc... \endtoc`. (Оглавление книги оформляется несколько иначе — см. раздел “Макетирование книги” на с. 191.) Команда `\toc` автоматически выдаст заголовок “CONTENTS” по центру страницы. Если вам нужен иной заголовок, можно сразу после `\toc` применить `\nofrills` и набрать то, что вам больше нравится.

Поскольку грамматически названия разделов статьи должны быть идентичны соответствующим названиям в содержании, содержание можно сконструировать посредством втягивания соответствующих строк из основного текста. Однако если некоторые заголовки были разбиты на строки посредством `\`, то эти знаки из текста в `\toc` следует удалить.

Как правило, в содержании статьи, которое достаточно короткое, страницы не указываются. Но если это нужно сделать, следует поступать так же, как и в случае монографии; см. раздел “Макетирование книги”.

Следующий ввод

```
\toc
\widestnumber\subhead{3.2.1}
\specialhead {} Introduction\endspecialhead
\head 1. Notation and terminology\endhead
\head 2. Some of the basic properties of tight closure for
ideals in characteristic  $p$ \endhead
\subhead 2.1. Notation and terminology\endsubhead
\head 3. Major constraints on systems of parameters
in characteristic  $p$ \endhead
\subhead 3.1. Test elements\endsubhead
\subhead 3.2. More test element with wrap around
to a second line\endsubhead
\subsubhead 3.2.1. Notation and terminology\endsubsubhead
\endtoc
```

даст такую распечатку:

CONTENTS

Introduction

1. Notation and terminology
2. Some of the basic properties of tight closure for ideals in characteristic p
 - 2.1. Notation and terminology
3. Major constraints on systems of parameters in characteristic p
 - 3.1. Test elements
 - 3.2. More test elements with a really, really, really long line that will wrap around to a second line
 - 3.2.1. Notation and terminology

Предполагается, что все вхождения в `\toc` состоят из двух частей: из номера заголовка и текста заголовка. Как явствует из строки “Introduction”, если заголовков не имеет номера, следует вместо него набрать `{}` и оставить пробел перед текстом заголовка. Вставка `{}` вместе со следующим далее пробелом приведет к тому, что этот заголовок будет выровнен по левому краю с соответствующими ему по значимости заголовками, но имеющими номер.

В содержании задано только два уровня выравнивания, хотя в статье может быть и 4 уровня значимости заголовков. Вхождения `\specialhead` и `\head` печатаются с одинаковым отступом, а `\subhead` и `\subsubhead` с другим, но тоже с одним и тем же.

Для `\specialhead` и `\head` нумерация принята в виде “1.”, а для `\subhead` и `\subsubhead` — в виде “1.1.” Размер отступа для `\head`, `\subhead` и `\subsubhead`

определяется посредством `\widestnumber`:

```
\toc
\widestnumber\head{10}
\widestnumber\subhead{10.1}
\widestnumber\subsubhead{10.1.1}
...
```

Если требуется, это можно повторить несколько раз для разных разделов содержания.

Если в содержание входит несколько разных по уровню значимости заголовков, вы можете захотеть, чтобы они внешне несколько отделялись друг от друга. Этого можно добиться, если сразу после строки `\toc` воспользоваться

```
\openup.5\baselineskip
```

Размер отделяющего пространства может быть иным, чем `.5\baselineskip` — см. **единицы измерения** в гл. 19 относительно деталей указания конкретных размеров.

Если вы готовите монографию, формат и все оформительские элементы будут иными. См. раздел “Макетирование книги”.

ТЕЛО СТАТЬИ

Даже после того как этапы преамбулы и сведений об издании пройдены, в самой статье остается еще немало специальных элементов оформления для привлечения внимания читателя.

- (1) Для организации материала используются заголовки и подзаголовки.
- (2) Математики любят устанавливать новые результаты под звуки фанфар и свои **Теоремы** и **Леммы** выделяют специальными шрифтами.
- (3) Взаимосвязанные сведения часто представляют списками.
- (4) Возможно, включают любопытные (или не очень) цитаты.
- (5) И, хоть это обычно считается дурным тоном, математики иногда используют подстрочные примечания.¹

Заголовки. Как уже говорилось, в стиле `\amsppt` четыре уровня заголовков (не считая `\title`):

```
\specialhead... \endspecialhead
\head... \endhead
\subhead... \endsubhead
\subsubhead... \endsubsubhead
```

В большинстве статей заголовком первого уровня по всей видимости будет `\head... \endhead`, потом `\subhead... \endsubhead`. Заголовок уровня `\head` печатается капителью, `\subhead` — полужирным, выровненным по левому краю с текстом в подбор.

¹Держу пари, вы удивлялись, видя их!

Первый заголовок в этой главе был набран как

```
\head Подготовительный этап\endhead
```

а подзаголовок этого раздела —

```
\subhead Заголовки \endsubhead Как уже...
```

Обратите внимание, что точка после *Заголовки* была поставлена автоматически, так что нет необходимости ее специально набирать.

Другие заголовки. Если в статье нужны еще и другие заголовки, для этого имеются `\specialhead... \endspecialhead` и `\subsubhead... \endsubsubhead`. Управляющая последовательность `\specialhead` используется в довольно длинных статьях, где требуется дополнительный раздел выше уровня `\head`. В стиле `amsppt` для `\specialhead` используется полужирный шрифт, и заголовок выравнивается по левому краю. Как понятно из названия, `\subsubhead` используется для подзаголовков по уровню значимости ниже `\subhead`. Текст между `\subsubhead` и `\endsubsubhead` печатается с абзацного отступа курсивом с автоматически проставляемой точкой. Подзаголовок в начале нашего абзаца был набран как

```
\subsubhead Другие заголовки\endsubsubhead Если в статье...
```

Явный разрыв строки в `\head` и `\specialhead` задается посредством `\`, но для `\subhead` и `\subsubhead`, представляющих собой часть абзаца, можно пользоваться лишь `\linebreak`, как для обычного текста, оформленного в абзац.

Если вы готовите монографию, заголовки будут выполнены в ином стиле. Относительно подробностей см. раздел “Макетирование книги”.

Теоремы и доказательства. Математики любят провозглашать свои утверждения в возвышенных тонах посредством `\proclaim`. В стиле `amsppt` ввод

```
\proclaim{Теорема (Народная мудрость)} Два плюс два --- четыре,  
а не пять и не три.  
\endproclaim
```

даст следующее:

Теорема (Народная мудрость). *Два плюс два — четыре, а не пять и не три.*

В соответствии со стилем журнала *Journal of the American Mathematical Society*, заголовки в `\proclaim` и в аналогичных конструкциях печатаются полужирным шрифтом. Однако в отличие от *Journal of the American Mathematical Society* в стиле `amsppt` текст в `\proclaim` печатается не курсивным, а наклонным шрифтом. (Стиль `amsppt` использует наклонный шрифт потому, что многие авторы предпочитают, чтобы в материале, содержащем большое количество математических знаков, текст отличался по написанию от формул.)

Заметим, что точка после пометки `{Теорема (Народная мудрость)}` печатается как неотъемлемая часть стиля. Дополнительный пробел, предшествующий

утверждению и `\endproclaim`, добавляет также некий дополнительный пробел. Но самой важной функцией `\endproclaim` является отмена наклонного шрифта. Если вы забудете `\endproclaim`, то остальная часть статьи будет напечатана наклонно — разве только вы наберете другой `\endproclaim`. У `AMS-TeX`'а нет способа узнать, когда утверждение следует завершить, но если вы забудете `\endproclaim` и вставите позже другой `\proclaim`, то `AMS-TeX` решит, что у вас что-то не в порядке, и это приведет к сообщению об ошибке.

Время от времени такую помпезность `\proclaim` следует подкреплять демонстративными доказательствами. Если вы наберете

```
\demo{Proof} See the {\it Collected Poems\} of
A. E. Housman.
\enddemo
```

то стиль `amsprt` преобразует это в

Proof. See the *Collected Poems* of A. E. Housman.

Двоеточие или иной знак препинания после слова, обозначающего “доказательство” (в нашем случае “Proof”), будет проставлен автоматически. В стиле `amsprt` слово, заключенное в фигурные скобки, печатается курсивным шрифтом, а `\enddemo` не означает ничего иного, кроме разрешения чуть-чуть увеличить вертикальный пробел в конце. Конец доказательства может быть отмечен специальным знаком \square , которому предшествует пробел размера `\quad`, что достигается посредством команды `\qed`.

Если вы забудете `\endproclaim`, то как только доберетесь до следующего `\demo`, `AMS-TeX` выдаст сообщение об ошибке. Пропуск же `\enddemo` никогда не приведет к сообщению об ошибке, поскольку никогда нельзя быть уверенным, что `\demo` завершилось: иногда `\demo` содержит внутри себя другое `\proclaim`, а иногда даже `\demo` для этого `\proclaim`.

Кроме обычных утверждений и доказательств, математики могут высказывать предложения иного рода, которые редакторы предпочли бы видеть оформленными в другом стиле. Стиль `amsprt` обеспечивает следующие опции:

```
\definition... \endddefinition
\example... \endexample
\remark... \endremark
```

`\definition` и `\example` в сущности идентичны: оба оставляют дополнительное пустое пространство перед заголовком, и сам заголовок печатается полужирным шрифтом. Точка в конце заголовка проставляется автоматически. Конструкция `\remark` аналогична `\demo` за тем исключением, что в отличие от `\enddemo` в конце замечаний `\endremark` не проставляет дополнительный вертикальный пробел.

Например, следующий ввод

```
\remark{Remark {\rm 9.12}} The progression formulas
(9.13) are easily derived from formula (9.23).
\endremark
%
\definition{Definition} The shadow  $F$  of form  $x^n \cdot k$ 
\text{(Irrational\ Junction)}\times \frac{(a_i-b_i k+c_i)}{(a_i'n+b_i'+c_i')} is obtained by placing every factor
 $(an+k+c)$  by  $(1)^{a+bk}/(-n+bk-1)$ , if  $a+b \neq 0$ , and
leaving it alone if  $a+b=0$ .
\enddefinition
\example{Example} In one variable,  $n$  ( $n \geq 0$ ) is
equivalent to  $(-1)^n/(-n-1)$  (defined for  $n < 0$ ). In two
dimensions  $(an+k-c)$ , where  $a$  and  $k$  are integers, is
equivalent to  $\frac{(-1)^{a+bk}}{(-an-bk-1)}$ .
\endexample
```

даст приведенный ниже текст. (Горизонтальные линейки были добавлены для того, чтобы отделить примеры от основного текста; они не предусмотрены стилем `amsppt`. Обратите внимание на использование `\rm` в заголовке `\remark`. В большинстве издательств, в том числе и в AMS, требуется, чтобы даже в курсивном тексте цифры были прямыми. Заголовки в `\remark` автоматически устанавливаются курсивными, так что приходится обращаться к `\rm`, для того чтобы выпрямить цифры.)

Remark 9.12. The double progression formulas (9.13) are easily derived from formula (9.23).

Definition. The shadow F of the form $x^n \cdot k$ (Irrational Junction) $\times \frac{(a_i-b_i k+c_i)}{(a_i'n+b_i'+c_i')}$ is obtained by placing every factor $(an+k+c)$ by $(1)^{a+bk}/(-n+bk-1)$, if $a+b \neq 0$, and leaving it alone if $a+b=0$.

Example. In one variable, n (for $n \geq 0$) is equivalent to $(-1)^n/(-n-1)$ (defined for $n < 0$). In two dimensions $(an+k-c)$, where a and k are integers, is equivalent to

$$\frac{(-1)^{a+bk}}{(-an-bk-1)}.$$

Еще ряд опций. Вот пример теоремы с перечнем свойств в формулировке:

Теорема. Предположим, что имеет место следующее:


- (1) Выполняются все стандартные законы аристотелевой логики, в том виде как они формулируются в классических учебниках;
- (2) Все люди смертны;
- (3) Сократ — человек.

Значит, Сократ смертен.

Для получения такого перечня у AMS-TeX'a есть средство `\roster`, в котором каждому элементу перечня предшествует `\item`. Предыдущий пример был набран следующим образом:

```
\proclaim{Теорема} Предположим, что имеет место следующее:
\roster
\item Выполняются все ...
\item Все люди смертны;
\item Сократ --- человек.
\endroster
\noindent
Значит, Сократ смертен.
\endproclaim
```

Обратите внимание, что `\roster` автоматически помечает различные `\item`. В стиле `amspt` элементы перечня будут помечены как (1), (2), Размер отступа может быть отрегулирован в соответствии с шириной пометок. Прежде чем открыть `\roster`, наберите, например, `\widestnumber\item{(viii)}`. Такая регулировка временная: соглашение “по умолчанию” будет восстановлено посредством `\endroster`.

 В гл. 19 объясняется, как отказаться от автоматического выбора вида пометок `\item` в `\roster` и назначить свои собственные пометки.

Если в статье содержится цитата из некоего источника, ее обычно стараются выделить, чтобы она отличалась от основного текста. В стиле `amspt` это достигается посредством `\block... \endblock`, который пропустит небольшой интервал по вертикали, напечатает цитируемый “блок” с отступом от правого и левого поля и оставит еще небольшой пробел по вертикали. Эта опция предназначена для использования внутри абзаца, так что не следует оставлять перед ней пустую строку или набирать `\par`, иначе вы получите дополнительный пробел перед цитатой.

Таблицы. В AMS-TeX'e нет специальных макро для создания таблиц. Такие макро доступны в других пакетах. Однако имеются команды (`\topcaption` и `\botcaption`), которые можно использовать для набора подписей под рисунками или в таблицах. Полное описание использования этих команд, равно как и способ включения в текст рисунка или таблицы в виде вставки вы можете найти в гл. 19.

Сноски. Чтобы получить сноску,¹ наберите просто

```
Чтобы получить сноску, \footnote{Наконец-то!}
наберите просто ...
```

и почти наверняка знак сноски (или цифра) появится непосредственно после предшествующей ей литеры, а само примечание — внизу страницы. В зависимости от конкретного стиля, используемого вами, будут задействованы те или

¹Наконец-то!

иные значки для сносок; в стиле `amsprt` приняты нумерованные сноски. Более подробно с подстрочными примечаниями можно ознакомиться в гл. 19.

Управляющая последовательность `\footnote` игнорирует пробелы и возвраты каретки, ей предшествующие, так что `\footnote` может набираться на отдельной строке:

```
Чтобы получить сноску,  
\footnote{Наконец-то!}  
наберите просто ...
```

Но пробел или возврат каретки после закрывающей скобки `}` в `\footnote` не игнорируется. Если вы не хотите пробела², вам следует его убрать:

```
Если вы не хотите пробела  
\footnote{Как в этом случае.},  
вам следует его убрать:
```

ССЫЛКИ НА ЛИТЕРАТУРУ

В библиографических сведениях обычно содержатся самые разнообразные виды информации. Следовательно, управляющие последовательности для работы с ними содержат некоторые включения, и их можно найти в приложении С. Математики, готовящие свои собственные работы, могут предпочесть оставить эти подробности кому-нибудь другому. Даже амбиции наборщика, занятого `TeX`нической работой, будут таковы, что он, вероятно, захочет отложить эти детали в долгий ящик. Но на литературу ссылаются и внутри самой статьи. В большинстве математических статей ссылки на литературу представляют собой цифру или иной идентификатор, заключенный в квадратные скобки:

В силу теоремы 4 из [10] и результатов работы [K-N] имеем ...

Иногда в квадратных скобках содержится дополнительная информация:

В силу [10, Theorem 4] и результатов работы [K-N] имеем ...

Хотя не составляет никакого труда набрать это вручную, в `AMS-TeX`'е имеется управляющая последовательность `\cite`, которая позволяет делать это еще проще. Приведенный выше пример был набран так:

```
В силу \cite{10, Theorem 4} и результатов работы \cite{K-N}  
... ..
```

В стиле `amsprt` средство `\cite` автоматически возьмет `\cite` в квадратные скобки.

• Упражнение 19.7:

Превратитесь в самиздатовца. Запишите несколько собственных мыслей и затем добавьте весь этот джентльменский набор, чтобы представить их в виде настоящей статьи. Дайте броское название и не упустите возможности проинформировать читателей о том, что автор — вы: пусть они узнают, где находится

²Как в этом случае.

ваш компьютер, и можете даже дать свой адрес, чтобы писали *вам*. Можете напечатать дату завершения своего опуса, поблагодарить всех, кто терпел ваши игры с компьютером, или снабдить краткой аннотацией для своих крайне занятых друзей. Найдите какое-нибудь шокирующее высказывание и провозгласите его как можно более помпезно. Создавая, наслаждайтесь.¹

МАКЕТИРОВАНИЕ КНИГИ

Если вы готовите монографию, то для этой цели в стиле `\amspt` имеется набор средств, позволяющих сформатировать ваш выходной файл в виде главы, а не в виде статьи. Макет каждой главы подготавливается отдельно и также по отдельности прогоняется через `AMS-TeX`.

Прежде всего сообщите в преамбуле о вашем намерении набирать монографию, поместив `\Monograph` сразу после строки `\documentstyle`.

Раздел `\topmatter` для главы монографии обычно выглядит следующим образом:

```
\documentstyle{amspt}
\Monograph
\topmatter
\title{chapter{4} Matrix Algebras\endtitle
\endtopmatter
```

Это приведет к такому заголовку:

CHAPTER IV

MATRIX ALGEBRAS

Обратите внимание, что номер главы автоматически будет дан римскими цифрами и будет добавлено слово "CHAPTER". Для глав, имеющих заголовки другого вида, следует применять `\nofrills`:

```
topmatter
\title{chapter\nofrills{Appendix B} The Poisson Integral\endtitle
\endtopmatter
```

В этом случае получается

Appendix B

THE POISSON INTEGRAL

Как вы могли видеть из предыдущего примера, при помощи `\nofrills` вы отмечаете еще и такой оформительский элемент, как прописные буквы для заключенного в фигурные скобки текста.

И наконец, для таких вещей, как предисловие и введение, у которых вообще нет никакого предваряющего заголовка текста, команда `\chapter` исключается:

```
\topmatter
\title Preface\endtitle
\endtopmatter
```

¹Можете дать сноску.

Оглавление книги. Обычно в монографиях оглавление рассматривается как отдельная глава, которую следует начинать с набора заглавия “Contents”, как было бы, если бы вы набирали предисловие или введение. Затем используется конструкция `\toc ... \endtoc` как тело этого документа (вместо того чтобы помещать это в `\topmatter`, как в случае журнальных статей). Например,

```
\topmatter
\title Contents\endtitle
\endtopmatter
\bigskip
\document
\toc
\title Preface\page{vii}\endtitle
\title\chapter{1} Matrix Algebras\page{1}\endtitle
\head Continuous complex-valued functions\page{1}\endhead
...
\title Bibliography\page{307}\endtitle
\endtoc
\enddocument
```

Заголовки глав, перечисляемые в оглавлении, воспроизводятся точно в таком же виде, как они были набраны в основном тексте. Для получения номеров страниц в оглавлении используйте `\page` как показано, сразу после самого элемента. Эта опция доступна для всех уровней заголовков.

Колонтитулы книги. В стиле `amsprt` в качестве левого колонтитула используется название главы, а в качестве правого — название раздела (из `\head`). Довольно стандартна ситуация, когда название раздела слишком длинно и не помещается в пространство, отведенное для колонтитула; в таких случаях используют `\rightheadtext` для указания сокращенной формы заголовка, предназначенной для колонтитула:

```
\head Fourier coefficients of continuous periodic functions
of bounded entropy norm\endhead
\rightheadtext{Fourier coefficients of periodic functions}
```

Это должно находиться сразу за `\head`, чтобы и то, и другое простирало свое действие на одну и ту же страницу. Если слишком длинным оказалось название главы, то сокращенная форма может быть представлена аналогичным образом с `\leftheadtext` сразу после `\title`. См. также раздел “Колонтитулы” в гл. 19.

Особенности макетирования книги. Стиль глав книг в некоторых деталях отличается от стиля статей. Текст внутри `\head` делается полужирным, а не капителью; заголовки теорем, предложений, определений, замечаний и пр. набираются капителью, а не полужирным, и с абзацного отступа, а не в левый край.

Шрифты

Специально для версии `AMS-TeX 2.0` был создан ряд новых шрифтов, причем семейство `Computer Modern` было дополнено недостающими размерами и были

созданы новые алфавиты и наборы символов, используемых для математических обозначений. Эти шрифты и составляют коллекцию AMSFonts, версия 2.0. Прежде чем использовать стиль `amsppt`, их следует инсталлировать на компьютер. Обратите внимание, что шрифты AMS версии 2.0 не могут быть использованы с более ранними версиями `AMS-TEX`'а, чем 2.0, и обратно. Некоторые из этих шрифтов автоматически доступны в стиле `amsppt`; они перечислены ниже.

Шрифты, загружаемые со стилем `amsppt`. Несколько шрифтов автоматически загружаются для целей общего характера.

- `cmcs8` представляет собой новый размер капители в Computer Modern.
- `cmex8` и `cmex7` являются новыми размерами шрифта математических расширений в Computer Modern, причем `cmex8` используется в стиле `amsppt` для аннотаций и других вспомогательных текстов, набираемых 8 кеглем, а `cmex7` — для верхних и нижних индексов.

Математические шрифты, загружаемые со стилем `amsppt`.

- `msam` и `msbm` содержат дополнительные символы. Эти символы и названия, которые позволяют их получить, перечислены в приложении G.
- `eufm` представляет собой шрифт Euler Fraktur (готический) среднего размера.

Приложение В. Ответы ко всем упражнениям

ППП.1. 1, 2, 4 или 8, в зависимости от ...

1.1. В первом случае на выходе получилось бы следующее:

In addition to the upper- and lower- case letters, and the numerals:

Нужный пробел после "upper-" получился, так как после `upper-` (carriage-return) дает тот же самый эффект, что и пробел. Но появился совершенно ненужный пробел в слове "lower- case". Строка исходного ТрХ-файла не должна заканчиваться на дефисе, разве что после этого дефиса действительно нужен пробел.

Во втором случае исходный текст с точки зрения машинистки выглядит вполне корректно, но он дает на выходе куда более серьезную ошибку:

In addition to the upper- and lower-case letters, and the numerals:

Никогда не следует вставлять дефисы в слова, чтобы выполнить их перенос на экране, что любят делать машинистки, поскольку перенос слов на экране не имеет ничего общего с разрывом слов, которым пользуется ТрХ. Кроме того, у него нет возможности понять, что это не является дефисом.

1.2.

О говорит: "Я все еще набираю два пробела после точки --- никак не могу отвыкнуть, но я всегда забочусь о том, чтобы в списках литературы при каком-нибудь перечислении, вроде страниц '480--491', вместо дефиса стояло короткое тире."

Обратите внимание, что ни пробелов, ни (carriage-return) вокруг короткого тире -- нет, а вокруг длинного есть (поскольку здесь текст на русском языке). Также заметьте, что нигде в этом ответе не использовались ". Тем не менее это упражнение следует стандартным правилам грамматики, по которым знак препинания помещается внутри кавычек, но в этом учебнике автор придерживается логического размещения знаков препинания и кавычек, так что подобное предложение он бы закончил так: ... короткое тире". Разумеется, для этого надо было бы набирать так:

... короткое тире''.

1.3. -- даст '- ', короткое тире, после которого следует пробел и затем дефис, а ---- даст '—', т. е. длинное тире, за которым следует дефис, и смотрится это ужасно.

Между прочим, - и — суть "лигатуры", новые символы, которые ТрХ вставляет автоматически, когда видит комбинации -- и ----. Точно так же он заменяет комбинацию букв fi на лигатуру 'fi'. Лигатуры, впрочем, не встроены в ТрХ, а определяются используемыми шрифтами. Если мы — для иллюстративных

ЦЕЛЕЙ — ВЫДЕЛИМ (SET OFF) ЭТО ПРЕДЛОЖЕНИЕ ШРИФТОМ “SMALL CAPS”, то увидим, что **ff** не дает лигатуры в этом шрифте, хотя --- все еще дает emdash, а “ и ” по-прежнему превращаются в двойные кавычки. Какой-нибудь разработчик шрифтов мог бы захотеть, например, чтобы в Е новом шрифте было бы сверхдлинное тире ‘—’ и О мог бы задать его в виде новой лигатуры, которую Т_ЭХ распознавал бы посредством ввода ----. (Обратите, однако, внимание на то, что в стиле **amsprt** шрифт “small caps” можно получать при помощи управляющей последовательности `\smc`, но другие стили могут не распознать эту управляющую последовательность.)

1.4. Т_ЭХ различает прописные и строчные буквы в названиях управляющих последовательностей, так что слово `\Par` рассматривается как совершенно отличное от слова `\par`. `\Par` не относится к управляющим словам, которые Т_ЭХ умеет распознавать, поэтому он остановится и выдаст сообщение об ошибке, заявив, что `\Par` не определено. Подробнее о сообщениях об ошибках вы узнаете в гл. 4.

1.5. Это абсолютно то же самое, как если бы вы оставили две пустые строки. Поскольку Т_ЭХ воспринимает две пустые строки как одну, эта цепочка просто будет означать конец абзаца. Такой способ обозначения конца абзаца может быть полезен в том случае, если вы хотите просмотреть свой файл поабзацно, а ваш текстовый редактор не умеет распознавать пустые строки. Тогда поиск абзацев по слову `\par` вам позволит организовать ваш текстовый редактор, а наличие пустых строк даст возможность различать в файле абзацы зрительно.

1.6.

Многие сносно печатают ‘`\$\$`’, гораздо труднее напечатать U.S. `\$\$` так, чтобы они выглядели сносно.

Обратите внимание, что после четвертого `\$` был оставлен пробел, потому что на выходе после четвертого `$` такой пробел нужен. Но после остальных `\$` никаких пробелов нет.

1.7.

`\AmSTeX` --- это просто специализированная версия пакета `\TeX`, но как только вы ее освоите, то приобщитесь к `\TeX` нической революции.

(В конце первой строки после слова `\TeX` можно было бы оставить пробел.)

1.8. `\$_` даст ‘\$ 1.00’ с двумя пробелами после знака \$. Ввод `\$_` даст абсолютно то же самое; первый пробел получится от пробела после `\$`, который принимается во внимание, поскольку `\$` представляет собой управляющий символ, а второй пробел даст стоящий за ним `_`.

2.1. Вы можете поступить так:

`{\AmSTeX}` --- это просто специализированная версия пакета `{\TeX}`, но как только вы ее освоите, то приобщитесь к `{\TeX}`нической революции.

Входной файл в этом случае выглядит изящнее, но потребуются несколько дополнительных ударов по клавишам. Обратите внимание, что после `{\TeX}` не нужно оставлять пробелы.

А можно поступить иначе:

`\AmSTeX{}` --- это просто специализированная версия пакета `\TeX`, но как только вы ее освоите, то приобщитесь к `\TeX{}`нической революции.

2.2. Пробелы в группе `{\P_L}` будут проигнорированы. Из следующих двух пробелов будет учтен только один в соответствии с тем правилом, что в последовательности пробелов `TEX` игнорирует все пробелы, кроме первого. Но каждая из следующих двух групп `{_}{_}` внесет по одному дополнительному пробелу.

2.3. `{-}{-}` или `-{-}` от `-{-}`, и т. п.

2.4.

`{\sl Восхитительный \tex\}` объясняет, как вставлять в исходный файл текст, который исчезнет `[{\sl sic\}]` при печати.

Если вы пользуетесь `\lbrack` и `\rbrack`, то надо набирать

`\lbrack{\sl sic\}\rbrack\` при печати.

2.5.

Закурсивленное `{\it одно\}` слово или даже `{\it пре\}`фикс или `{\it суф}`фикс слова смотрятся неплохо, но не `{\it перебарщивайте}`.

Этот пример демонстрирует, что важно вставлять необходимые пробелы после `\}`; `AMS-TeX` не расставляет их автоматически, поскольку в некоторых случаях они бывают не нужны. С другой стороны, нет никаких оснований оставлять пробел перед `\`, так что если вы наберете `{\it одно \}`, `AMS-TeX` просто проигнорирует этот пробел, считая, что вы хотели набрать `{\it одно\}`. Во всяком случае, если не прибегать к группированию, это предложение можно было бы набрать так:

Закурсивленное `\it одно\ \gm` слово или даже `\it пре\ \gm` фикс или `\it суф\ \gm` фикс слова смотрятся неплохо, но не `\it перебарщивайте\gm`.

Здесь для правильной расстановки пробелов требуется повышенное внимание.

2.6.

Большинство грамотеев скажут, что знаки препинания следует ставить в том же начертании, что и `{\it основной текст:\}`, но многие пишущие предпочитают `{\it переключиться\}`; это особенно верно в случае точки с запятой.

Поправка на курсив после двоеточия кажется не столь существенной; без нее вы получили бы '... что и *основной текст*: но многие ...', что кому-то, возможно, больше понравится.

2.7. Если бы вы вместо “`{\it переход/}`” набрали “`{\it переход}`”, то получили “переход”, с буквой *d*, запутавшейся в кавычках.

2.8. Мы можем набрать

```
W. Ambrose,
{\sl Higher order Grassmann bundles/} {\bf3}
(1964), 199--238.
```

либо вставить `\bf` внутри группы `\sl`

```
W. Ambrose,
{\sl Higher order Grassmann bundles/ \bf3}
(1964), 199--238.
```

Во втором случае, когда `TeX` натолкнется на `}`, шрифт уже будет переключен на `\bf`, который `}` соответственно отключит.

(На самом деле эти два ответа не эквивалентны, поскольку в первом случае пробел перед `3` представлялся в обычном прямом шрифте, тогда как во втором он был взят из шрифта `\sl`, но вряд ли кто-нибудь заметит разницу.)

2.9.

```
{\it Объясните, как набрать/ {\gm прямое} слово ... предложения.}
```

2.10.

```
{\sl Если все предложение набрано наклонным шрифтом, можно
подчеркнуть для выделения одно слово в/ \underbar{\gm прямом}
начертании.}
```

2.11.

```
Everyone will be happy when \TeX\ finally arrives,
and Dr. Treemunch can start typing his own so-called
scholarly manuscripts. His last {\it opus/}---%
which The Amer. Jour. Recr. Drugs recently published---%
created quite a sensation, particularly the material
that he has expounded on pages 22--23.
Including this material cost an extra
\$,1,000, but it did make \P\P\ 1 and 2 quite popular.
```

The head of the department, our university's own I. M. Stable, attributes Treemunch's recent aberrant behavior to the much-publicized "research" for his paper; notwithstanding, others say that Treemunch's name isn't on the computers' databases, so his name often gets hyphenated quite strangely, which may account somewhat for his feeling of being ill-used.

3.1. Теперь вы получите сообщение

```
Underfull \hbox (badness 1226) ...
```

Эта строка оказалась слишком разреженной, в результате чего и появилось сообщение `Underfull \hbox`, и “порочность” такого растяжения составила 1226. Посмотрев на распечатку, вы сами можете решить, насколько серьезен этот порок. Стил `amsprt` обычно допускает разрыв строки, в результате которого “порочность” не превышает 200, при этом требуется, чтобы пробелы между словами были одинаковыми.

3.3. Это хуже, чем было раньше! Добавление `\hyphenation{data-base}` в начале этого файла не поможет Т_ЭX’у перенести слово “databases”. Если мы собираемся часто использовать слово “database”, то следует добавить

```
\hyphenation{data-base}
\hyphenation{data-bases}
```

и тогда Т_ЭX будет знать, как переносить все возможные формы этого слова; для Т_ЭX’а “database” и “databases” — просто два разных слова, и он не распространяет перенос с одного на другое. Не разбитое на переносы слово “databases” теперь еще больше вылезет на поле, потому что Т_ЭX знал, как перенести “Treemunch’s”. Это связано с тем, что Т_ЭX считает “словом” последовательность букв, так что “Treemunch’s” не является другим словом — это прежнее слово “Treemunch”, после которого стоит апостроф. (В качестве интересного упражнения вы можете добавить

```
\hyphenation{Tree-munch's}
```

в свой файл и посмотреть, что произойдет.)

4.1. Оттранслированный файл начнется с:

```
This is TeX, Version 1.1 (preloaded format=plain 84.8.1)
**goof
(goof.tex (amstex.tex) (amsprt.sty)
! Undefined control sequence.
1.4 Here is a word in {\it
                                italics\} and here is one
? i\it
```

Обратите внимание, что ваше предложение `i\it` включено в сообщение об ошибке. (Заметьте также, что если бы вы вызывали Т_ЭX посредством `tex goof`, оттранслированный файл начинался бы с `**goof`, как если бы вы просто набрали `tex` а затем `goof` как отклик на подсказку `**`.)

В оттранслированном файле будет также содержаться помощь, и наконец он благополучно завершится следующим:

```
[1]
Output written on goof.dvi (1 page, 548 bytes).
```

4.3. Вы получите такое сообщение об ошибке:

```
! Undefined control sequence.
1.4 Here is a word in {\ititalics
                                \} and here is one
?
```

Теперь Т_ЭX игнорирует `\ititalics`, так что следует набрать `i\it italics`.

4.4. После появления [1], означающей, что 1-я страница готова к печати, появится сообщение

```
(\end occurred inside a group at level 1)
```

Взглянув на распечатку или экранный вывод, вы легко поймете, где скобки оказались несбалансированными, так как весь текст после *“italics”* также окажется курсивным.

Если статья достаточно длинная, вы вряд ли захотите тратить время на ее распечатку или вывод на экран и попытаетесь обнаружить ее с терминала. Для этого следует вставить `\enddocument` где-то посередине статьи и снова прогнать ее через Т_ЭХ. Если появится такое же сообщение об ошибке, то несбалансированные скобки находятся в первой половине этой статьи, в противном случае — во второй. В последнем случае перед `\enddocument` поставьте %, чтобы “закомментировать” эту команду, но иметь возможность найти ее в случае необходимости, и поместите еще одну команду `\enddocument` примерно на четверть от конца файла. Если же искомое событие произошло в первой половине, “закомментируйте” команду `\enddocument` и еще одну такую же команду поместите на расстоянии четверти от начала файла. Продолжая этот процесс, вы сможете выделить достаточно малую область, которую можно будет без особых усилий внимательно изучить.

4.5. Символ ‘!’ не является узаконенным ответом на сообщение об ошибке, поэтому Т_ЭХ перечислит допустимые ответы:

```
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.
```

Относительно клавиш `<return>`, I, H и X было сказано ранее. Нажатие E в ответ на сообщение об ошибке вернет вас в редактор в то место, где была обнаружена эта ошибка. Про другие опции рассказано в части 3.

4.6. Когда Т_ЭХ доберется до конца вашего файла, он выдаст запрос *. В отличие от **, которая запрашивает имя файла при первом обращении к Т_ЭХ’у без указания имени файла, * просто запрашивает дальнейший ввод. В этом месте не следует набирать i, наказывая Т_ЭХ’у что-нибудь вставить, поскольку Т_ЭХ уже миновал стадию исследования ввода. Если вы все же нажмете клавишу i, то не получите запрос `\insert>` — Т_ЭХ подумает просто, что i — это дальнейший текст и снова откликнется на это запросом *. Правильным будет набрать просто `\enddocument`.

Однако запрос * можно получить, даже если ваш файл снабжен в конце необходимой командой `\enddocument`. Некоторые ошибки могут привести Т_ЭХ в странное состояние, при котором он пожирает вашу команду `\enddocument` и затем продолжает пожирать все, что вы набираете после запроса *, вновь и вновь посылая вам очередной запрос *. В этом месте может возникнуть искушение прибегнуть к встроенному в операционную систему прерыванию, но это

в известной мере шаг отчаяния, потому что вы, вероятно, не получите файла транскрипции, позволяющего исследовать ситуацию. Вместо этого можете попробовать откликнуться на такой запрос * какой-нибудь неопределенной управляющей последовательностью вроде `\foo`. Т_ЕX, по-видимому, выдаст сообщение об ошибке, заявив, что `\foo` не определена, после чего поставит обычный ? на следующей строке. Тут вы можете набрать `X` или `x` и изящно удалиться.

4.7. Вы получите сообщение об ошибке

```
! I can't find file 'goofy.tex'
<*> goofy
```

Please type another input file name:

(Не могу найти файл 'goofy.tex'. Введите, пожалуйста, другое имя:)

Здесь следует набрать правильное имя файла `goof`.

4.8. Т_ЕX выдаст сообщение об ошибке

```
! I can't find file 'mystyle.sty'.
1.2 \documentstyle{mystyle}
```

Please type another input file name:

Обратите внимание, что команда АМ_S-Т_ЕX'a `\documentstyle{mystyle}` велит Т_ЕX'у прочесть файл `mystyle.sty`, а не `mystyle.tex`, так как стилевые файлы имеют расширение `sty`. Здесь правильным решением будет набрать `amsrpt.sty`. Если вы просто наберете `amsrpt`, Т_ЕX станет искать файл `amsrpt.tex`, и снова объяснит вам, что не может его найти.

5.1. ZZ Zzzz zz.

5.2. Он заменил `\P\P\ 1 and 2` на `\P\P\~1 and ~2`, забыв удалить `\` после `\P\P`. В результате образовалась управляющая последовательность `\~`, о которой вы узнаете из следующей главы, приведшая к тому, что `~` оказалась над следующим символом.

5.3.

```
Weird fruit (mangos, papayas, etc\.) are
avoided by farmers, fastidious eaters, et al.
```

Если бы вместо `'etc\.'` вы набрали `'etc.'`, то пробел после правой скобки получился бы больше — дополнительный пробел в конце предложения “перескочил” бы через правую скобку, заботясь о ситуации, когда все предложение заключено в скобки. После `'al'` символ `'\.'` не используется, поскольку эта аббревиатура оказалась в конце предложения (равно как и после `'et'`, так как это и вовсе не аббревиатура!).

6.1. Это правило не работает, когда % входит в управляющую последовательность `\%`. (Т_ЕX читает в строке слева направо, и как только он встречает литеру `\`, то тут же прочитывает вперед, пока не окажется в конце управляющей последовательности; так что `\%` исчезнет из строки, прежде чем Т_ЕX соберется проигнорировать все, стоящее за знаком %.)

6.2.

My \#1 solace is M\&M's, though any candy is dandy;
 I agree 100\% with O." Nash\dag\ that liquor
 is quicker, but a fifth of J\&B \@^\\$13.95 \{price as of
 this writing\} is beyond my means.

6.3. В слове h\`a\vc ek ошибка состоит в том, что нет пробела после \v, и поэтому T_EX воспримет это как управляющее слово \vc. В h\` a\v с ek нужный пробел после \v имеется (а также пробел после \`, который допустим), но зато есть неправильный пробел после с; такой набор даст в результате 'háċ ek'.

6.4.

below\`ed prot\`eg\`e; r\`ole co\`ordinator;
 soufl\`es, cr\`epes, p\`at\`es, etc.

6.5. Надо набрать na\`i ve (или na{\`i}ve или na{\`i}ve).

6.6. Ernesto Ces\`aro, P\`al Erd\H os, Serge\u\i\ Yur\`ev, Eduard \v Cech,
 \b T\`abit ibn Qorra, Mu\d hammad ibn M\`us\`a al-Khw\`ariz%,
 m\`i. Обратите внимание на _ после \i в Serge\u\i.

6.7. \AE sop's \OE uvres en fran\c cais. Можно также набирать {\AE}sop's
 {\OE}uvres en fran\c{c}ais; это выглядит несколько симпатичнее в файле, но затрудняет набор.

6.8. \O ystein Ore, Anders Jonas \AA ngstr\`om, Stanis\l aw \`Swie%
 rczkowski

6.9.

?\`What did you say, Se\`nor?
 I said, ``!\`Ma\`nana is good enough for me!''

6.10.

{\sl Commentarii Academi\ae\ Scientiarum Imperialis
 Petropolitan\ae\} is now {\sl Akademi\`t\i a
 Nauk RF, Doklady.}

6.11. Вы получите: Да ... ,волноваться надо самую малость.

6.12.

Теперь вы способны напечатать в \tex'e 95\% из обычно встречающихся текстов. Но достижение б\`ольших успехов в \tex ническом наборе требует больш\`их познаний: ведь вы владеете далеко не всеми возможностями \tex'a, поскольку все еще не умеете правильно набирать \dots, ну, скажем, то, что встретилось в этом абзаце.

7.1. Вы получите "Если формула $y = x - 1$ справедлива ... ", поскольку не было пробела перед знаком \$. Пробелы до и после формул A_MS-T_EX автоматически не вставляет, так как вам они не всегда бывают нужны, как можно убедиться из следующего упражнения.

7.2.

Удаление одного элемента из $\$n\$$ -ки приводит к $\$(n-1)\$$ -ке.

Хотя \LaTeX обычно не делает пробелов до и после формул, иногда наборщики любят, чтобы формулы немного больше отделялись от текста. Они не напечатают $z = x + y$, но вместо этого напечатают $z = x + y$, добавив чуть больше воздуха вокруг формулы. Если вам случится воспользоваться стилем, в котором заложена такая опция, то \LaTeX будет автоматически добавлять дополнительные пробелы. К сожалению, у \LaTeX 'а нет способа распознать случай, когда этот дополнительный пробел не требуется. Здесь вы должны подсказать \LaTeX 'у, что нужно полное слияние, введя “ $\backslash\snug$ ”:

Удаление одного элемента из $\$n\backslash\snug\$$ -ки приводит к $\$(n-1)\backslash\snug\$$ -ке.

7.3.

Рассмотрим граф $\$f-g+h\$$ в плоскости $\$x\$--\$y\$$.

Заметим, что в этом случае нельзя набирать в плоскости $\$x--y\$$, поскольку это приведет к ‘в плоскости $x - -y$ ’.

7.4. $\$1+2\{3+4[5+6(7+x)]\}\$$

7.5. Вы получите $1 + 23 + 4[5 + 6(7 + x)]$; фигурные скобки $\{$ и $\}$ используются просто для группировки, при которой ничего интересного не происходит.

7.6. Вы получите формулу $x+y$ без пробелов вокруг знака $+$. В математическом режиме использование фигурных скобок вокруг символа приводит к тому, что он воспринимается как обычный символ, даже если это обычно бинарный оператор или иной символ, предполагающий пробелы вокруг него. Так что не злоупотребляйте фигурными скобками без особой на то необходимости.

7.7. Это одна из ситуаций, упоминавшихся в ответе к упр. 7.6: если вы наберете $\$1\{,000z\$$, то получите $1,000z$ без всяких пробелов вокруг запятой.

7.8. Прежде всего, вам не захочется заменять *все подряд* x , поскольку некоторые из них входят в text . Так что ваш редактор лучше настроить на опцию “query replace” (замена с подтверждением), которая будет показывать вам каждый x и спрашивать, хотите ли вы произвести замену. (К счастью, x вне математических формул не общеупотребителен. Если бы вам нужно было заменить все ϵ в формулах на $\backslash\epsilonpsilon$, то даже такой способ не сработал бы.) Но здесь имеется подвох: обязательно меняйте x в формулах на $\backslash\kappaappa_{\square}$, а не просто на $\backslash\kappaappa$, так чтобы ввод типа xu не привел к $\backslash\kappaappa\text{ray}$!

7.9. Обязательно меняйте $\backslash\kappaappa$ на $\square\kappa$, чтобы ввод типа $\backslash\alpha\backslash\kappaappa$ не превратился в $\backslash\alpha\text{pk}$.

7.10.

- (1) $\$X\setminus(A\cup B) = (X \setminus A)\cap(X \setminus B)\$$
- (2) $\$x\notin A\not\subset B\$$
- (3) $\$(X\times Y)\times Z\subseteq X\times(Y\times Z)\$$
- (4) $\$\omega\wedge(\eta\wedge\lambda)=$
 $\quad(\omega\wedge\eta)\wedge\lambda\$$
- (5) $\$V\oplus\Lambda(V)\$$
- (6) $\$\|a(x+y)\|\leq\|a\|\cdot(\|x\|+\|y\|)\$$
- (7) $\$2\cdot\aleph=\aleph\$$
- (8) $\$2\cdot\omega\neq\omega\$$
- (9) $\$\nabla R(X,Y)\$$
- (10) $\$(100\text{pm.}001)\div 5\$$
- (11) $\$\forall x\in A\$$
- (12) $\$f+g:A\to B\$$
- (13) $\$x\mapsto\alpha+x\$$
- (14) $\$f(x)\in o(x) \text{ \& } g(x)\in o(x)$
 $\quad\text{\implies } f\circ g(x)\in o(x)\$$

Обратите внимание, что в (1) `\setminus` применяется для того, чтобы полученный символ отличался от `\backslash`. В некоторых шрифтах в обоих случаях используется один и тот же символ, однако пробелы будут разными — `X\backslash Y` дает $X\backslash Y$, тогда как `X\setminus Y` дает $X \setminus Y$. (Вы, разумеется, можете проконсультироваться у автора, какой именно символ нужен; `\setminus` более употребителен.)

В (2) следует обратить внимание на то, что $\not\subset$ набирается как `\not\subset` для получения “отрицания” \subset , но \notin набирается как `\notin`.

В (6) обратите внимание на то, что `|` дает $|$, в то время как `\|` дает двойные вертикальные линии $\|$. Если на вашей клавиатуре знака `|` нет, можете вместо него использовать `\vert`; в этом случае для двойных вертикальных линий используйте `\Vert` (нельзя применять `\|`, поскольку \TeX воспримет это как управляющую последовательность `\|`, за которой следуют буквы `vert`).

В (9) `\nabla` представляет собой “перевернутую” Δ , однако остальные греческие буквы аналогичных вариантов не имеют.

В (10) обратите внимание на то, что `\pm` (\pm) представляет собой единый символ — вы не получите его посредством `\+-`, что даст просто $+-$; существует также символ `\mp` (\mp). Точно так же \gg в (11) — это особый символ, который не получится при помощи `\>>`; последнее даст $>>$.

Формулу $f: A \rightarrow B$ часто произносят как “ f переводит A в B ”. Для формулы $f * g: A \rightarrow B$ в (12) мы используем `\;`, чтобы получить справа от двоеточия пробел (бывают также ситуации, когда в роли разделителя выступает двоеточие `:`). Правая стрелка \rightarrow называется `\rightarrow` (аналогично `\leftarrow` для \leftarrow), но `\to` распознается как синоним. Некоторые авторы скрупулезно различают стрелки \rightarrow и \mapsto (`\mapsto`); последняя использовалась в (13).

Знак \implies в формуле (14) можно получить при помощи `\Longrightarrow`, что длиннее чем `\rightarrow`; как вы увидите в приложении F, буква L в начале

этого имени отличается от первой буквы в `\longrightarrow`; последнее представляет собой длинную единичную правую стрелку. Но в настоящей формуле `&` и `\implies` заменяют логические “и” и “влечет”; `\and` дает даже большие пробелы вокруг `&`, чем обычно имеют бинарные операции, и аналогично `\implies` приводит к дополнительным пробелам вокруг `\implies`.

Наиболее интересные моменты в формуле (14) — это `o` и `O`, которые появляются в виде `o(x)` и `O(x)`. Заметьте, что здесь были набраны буквы `o` и `O`, а не цифра `0`. В качестве общего правила: Буквы `o` и `O`, а не цифра `0` пишутся тогда, когда следующим символом является левая скобка `(`. Единственным исключением может быть формула типа `20(x+y)`, где `0` есть часть более длинного числа `20`.

Другие подобного рода подробности вы найдете в части 3.

7.11. Сначала возникнет сообщение об ошибке

```
! Missing $ inserted.
```

```
...
\par
1.6
```

которое очень похоже на то, что было раньше, за исключением того, что теперь ошибка обнаруживается, когда `TEX` достигает строки `6`, пустой строки, которая вызывает тот же эффект, как если бы `\par` было набрано.

Следующее сообщение

```
! Missing $ inserted.
```

```
...
1.7 This paragraph has the formula  $3\alpha$ 
+\beta
```

также напоминает одно из полученных ранее. Заметим, однако, что `TEX` не вставляет `$` перед `3`, поскольку появление `3` вне математической формулы вполне возможно.

Следующее сообщение об ошибке

```
! Missing $ inserted.
```

```
...
\par
1.8
```

возникает когда `TEX` достигает конца второго абзаца: поскольку `TEX` вставил `$` перед `\alpha`, он, очевидно, должен вставить еще один до окончания абзаца!

Далее возникает аналогичная пара сообщений об ошибке, причем первой в паре будет

...

```
1.9 This is accurate within \pm
      .0003 percent.
```

Опять же, Т_ЭX понимает, что \$ нужно вставить перед \pm, поскольку \pm математический символ, который допустим лишь в математическом режиме.

(Тем не менее, вы можете поэкспериментировать с файлом, заканчивающимся

```
This is accurate within \pm.00003 percent.
\enddocument
```

Вы увидите, что Т_ЭX еще больше запутается, поскольку не обнаружит, что нужен закрывающий \$, пока не поймет, что означает \enddocument.)

7.12. ¡Вы получаете следующее:

The symbols ¡ and ¿ and — give < and > and | in math mode.

— вне математического режима можно использовать <, > and |, но вы, разумеется, не получите того, что ожидали! Одни и те же вещи Т_ЭX интерпретирует совершенно по-разному в математическом и текстовом режимах.

8.1.

```
There exist such division algebras
only for $n=1$, `2, 4 or `8.
```

8.2.

```
We have $f(x)=A$, $B$ or`$C$ for`$x=0$, `1, \dots,`$n$.
```

8.3.

```
For all $a$ and $b$ we have $a<b$, $a=b$, or`$a>b$.
We say that `$<$ is a {\it partial ordering}.
```

Этот пример иллюстрирует еще один важный момент: заметим, что первая точка набирается вне знаков \$. Это существенно. Если бы мы набрали \$a>b.\$, то эта точка была бы частью формулы, и тогда правила Т_ЭX'а относительно пробелов после точек в тексте были бы неприменимы, и мы бы получили обычный межсловный интервал между формулой 'a > b.' и словом 'We'.

8.4. Если вы наберете

```
If $a, b , c>0$, then $f(a,b,c)>0$.
```

то получите

If $a, b, c > 0$, then $f(a, b, c) > 0$.

Это не так уж плохо, но если набрать

If $a, b, c > 0$, then $f(a, b, c) > 0$.

то с пробелами дело будет обстоять несколько лучше. Это довольно специфический случай: фраза " $a, b, c > 0$ " фактически представляет собой аббревиатуру от " $a > 0, b > 0, c > 0$ ", и тут довольно трудно решить, следует трактовать ее как одну формулу или как три. Заметим, однако, что во всех случаях $c > 0$ остается в одной математической формуле, так что пробелы вокруг $>$ будут правильными.

8.5.

We have shown that $f(1, \dots, n) \leq f(0, \dots, 0) + f(1, \dots, 1) + \dots + f(n, \dots, n)$ for $n \geq 1$.

8.6.

После невообразимых усилий, предпринятых в предыдущей главе, мы, наконец, успешно доказали фундаментальный результат

$$1+1=2,$$

а теперь попробуем доказать, что

$$2+2=4.$$

В качестве первого шага в этом направлении докажем дистрибутивный закон умножения.

Заметим, что \TeX не начинает нового абзаца после выключной формулы, если вы особо это не укажете, оставив пустую строку или набрав `\par`.

8.7. Ну и дела! Как я могу знать — многострочные формулы не описываются до гл. 15. Верно! Мы только хотим отметить тот факт, что \TeX определенно не стал бы рвать формулу для вас. Если вы наберете

$$1+2+3+4+5+6+7+8+9+10+11+12+13+14+15+16+17+18+19+20=190$$

\TeX постарается изобразить это в виде однострочной формулы, и вы получите сообщение `Overfull box`. Это хороший пример для пробы. Он также позволяет понять, что произойдет, если набрать

$$a+b=c$$

$$A+B=C$$

Вместо того чтобы получить

$$a + b = c$$

$$A + B = C$$

у вас будет два отдельно выключенных уравнения $a + b = C$ и $A + B = C$. Между ними окажется слишком большое пространство (а именно, пространство после первой выключной формулы плюс пространство перед второй выключной формулой), и каждое из уравнений будет выровнено по центру отдельно. Поскольку буквы в этих выражениях имеют разную ширину, знаки $=$ не окажутся точно друг под другом.

9.1. Ну конечно, есть! Если в первом случае вы получите $x + {}_2F_3$, то во втором $x + {}_2F_3$, где 2 будет выступать в качестве индекса у знака + (когда бинарный оператор типа + имеет нижний индекс, вся снабженная индексом комбинация трактуется как бинарный оператор, который требует дополнительного пробела после +₂ в этой формуле). Это полезно знать для ситуаций, когда бывает нужно что-нибудь вроде +₁, +₂, ..., но это, вероятно, не то, что хотелось в нашем случае.

9.2.

If the $n-1$ numbers $x_1, \dots, x_{\{\alpha-1\}}, x_{\{\alpha+1\}}, \dots, x_n$ are all $\neq x_{\alpha \pm 1}$, then $f(x_1, \dots, x_n) > 0$.

(Обратите внимание на пробелы между \dots, (или \dots,) и следующей математической формулой.)

9.3. Первым сообщением об ошибке будет

! Missing \$ inserted.

....

1.4 This paragraph has the formula x^2 .

?

В этом случае Т_ЭX вставил знак \$ непосредственно перед ^, поскольку ^ может появляться только в математическом режиме. На букве x сообщение об ошибке не появилось, поскольку x может, разумеется, появиться и вне математического режима. Следовательно, Т_ЭX отреагирует так, как если бы мы набрали x^2 .

Нажав (carriage-return), снова получим сообщение об ошибке

! Extra }, or forgotten \$.

1.6 This paragraph has the formula x^{10}

\$

?

Это аналогично сообщению ! Extra }, которое мы уже встречали. Однако поскольку сейчас Т_ЭX находится в математическом режиме, то он лишь считает скобки внутри знаков \$; принимается во внимание то обстоятельство, что лишняя } может возникнуть из-за того, что мы просто забыли поставить закрывающий знак \$ — возможно, мы на самом деле хотели набрать x^{10} (если бы мы так набрали, то Т_ЭX с легкостью проскочил бы через формулу x^{10} , создав выражение x^{10} , но потом, встретив }, выдал бы сообщение об ошибке.

Снова нажав (carriage-return), получим следующее сообщение об ошибке:

! Missing } inserted.

....

1.7 and the formula x^{10}

?

Теперь ТрХ решил, что } лишняя, потому что достиг второго знака \$, свидетельствующего о завершении формулы, но фигурные скобки в ней оказались не сбалансированными. После очередного нажатия (carriage-return) ТрХ благополучно достигнет конца файла.

9.4.

В некоммутативной группе имеем

$$((ab)^{-1}=b^{-1}a^{-1})$$

и

$$((ab)^{-2}=[(ab)^{-1}]^2=[b^{-1}a^{-1}]^2$$

$$=b^{-1}a^{-1}b^{-1}a^{-1}),$$

$$((a^m)^{-2}=a^{-2m}).$$

но

$$((a^m)^{-2}=a^{-2m}).$$

9.5.

$$(2^{x_1+\dots+x_{n+1}}=2^{x_{n+1}} \cdot 2^{x_1+\dots+x_n})$$

$$(x^x)^x.$$

9.6.

Suppose that $(x^x)^x =$

$$(x^x)^x. \text{ Prove that } x=1 \text{ or } 2.$$

9.7. Первое дает x^{y^z} , а второе — x^{y^z} . В первом случае z есть просто верхний индекс формулы x^y , поэтому имеет размер s-size и располагается на другой высоте. Математиков никогда не устроит первый вариант, так что нечего о нем и думать.

9.8. Вы получите примерно такое сообщение об ошибке (о двойном верхнем индексе):

! Double superscript.

$$a^{b^c}$$

?

Если нажать (carriage-return), ТрХ живо проскочит файл и вы получите a^{bc} : ТрХ как бы вставит пустую группу {} перед вторым верхним индексом, что превратит ваш набор в $a^{b\{ }\ ^c$.

9.9.

Пусть не существует такого λ , что

$$x_i = \lambda y_i, \quad i=1, 2.$$

Тогда уравнение $(\lambda y_1 - x_1)(\lambda y_2 - x_2) = 0$, т.е., уравнение

$$(\lambda^2(y_1^2 + y_2^2) - 2\lambda(x_1 y_1 + x_2 y_2) + (x_1^2 + y_1^2)) = 0$$

$$(\lambda^2(y_1^2 + y_2^2) - 2\lambda(x_1 y_1 + x_2 y_2) + (x_1^2 + y_1^2)) = 0$$

$$\lambda$$

не имеет решения λ .

9.10. Вы получите:

В формуле “ $x + y$ ” ...

Правильно было бы набрать

В формуле ‘“ $x+y$ ”’ ...

и тогда ’’ дал бы двойную кавычку. (Тем не менее, ‘ неожиданно дает левую кавычку в математическом режиме; на самом деле даже ‘‘ дает двойную левую кавычку. Но это чистая случайность — в один прекрасный день кто-нибудь придумает что-нибудь для ‘ в математическом режиме, аналогично тому, что было сделано для ’.)

9.11. Т_ЭX просто оттранслировал бы это в $g^{\prime 2}$, что недопустимо.

9.12.

$$\begin{aligned} \text{\$} f^{-(n+2)} &= [f^{-(n+1)}]’ = [f^{-(n)}\{\}’]’ \\ &= [f^{-(n)}]’\text{\$} \end{aligned}$$

9.13. Можно было бы ожидать, что Т_ЭX сделает с A_{\notin} нечто странное, как было в случае A_{\neq} . Но теперь мы на самом деле получаем сообщение об ошибке

! Missing { inserted.

. . .
. . .

... This file contains only the formula A_{\notin} \$.
?

значительно менее очевидное, чем мы встречали до сих пор. Последние две строки свидетельствуют о том, что Т_ЭX споткнулся на \notin , а из двух верхних видно, что он попытался исправиться, поставив { перед \notin (что было бы превосходно в случае A_{\neq}). Точная причина этого покрыта мраком — \notin такая же аббревиатура, как и \neq , но, очевидно, более сложная — и опять самый лучший вариант — это попытаться снова нажать \langle carriage-return \rangle и уповать на лучшее! В самом деле, поступив так, получим другое сообщение об ошибке:

! Missing } inserted.

. . .

... This file contains only the formula A_{\notin}
?

На этот раз \TeX пошел несколько дальше, дошел до знака $\$$, закрывающего математическую формулу, и решил, что нужно вставить } (весьма разумное решение, так как перед этим он вставил {). Если снова нажать (carriage-return), то мы благополучно окажемся в конце файла.

Разумеется, такое затруднение можно было бы обойти, если бы вы знали, что вам следует заключить `\notin` в фигурные скобки. Не беспокойтесь об этом. Вероятность того, что вы в действительности столкнетесь с \notin (или с иным символом, вызывающим аналогичные затруднения) в нижнем индексе, весьма мала. Но если это все же случайно произойдет, вы, по крайней мере, будете знать, что пара фигурных скобок может все уладить.

10.1. Наиболее надежный способ такой:

```
$$\frac{x{1+{\dspace\frac{x2}}}}{}
```

и

```
$$\frac{{\dspace\frac{x2}+1}2}{}
```

с аккуратно заключенным в фигурные скобки выражением `\dspace\frac{x2`, чтобы оградить `\dspace` от влияния на что-нибудь еще. Но в обоих случаях эти скобки можно было бы опустить, набрав просто

```
$$\frac{x{1+\dspace\frac{x2}}{}}
```

и

```
\frac{\dspace\frac{x2+1}2}
```

В первой формуле влияние `\dspace` распространяется только до следующей } (и к тому же она завершает формулу); во второй формуле фигурные скобки необязательны потому, что `\dspace\frac{x2+1` выглядит точно так же, как и `{\dspace\frac{x2}+1`.

10.2.

```
{\sl H"older's Inequality\}: Let  $\forall p, q \in \mathbb{N}$ 
with  $\frac{1}{p} + \frac{1}{q} = 1$ .
If  $f \in L^p(\mu)$  and  $g \in L^q(\mu)$ ,
then  $fg \in L^1(\mu)$  and
 $\|fg\|_1 \leq \|f\|_p \|g\|_q$ .
```

10.3.

```
$$\frac{x{1+\dfrac{x2}}{}}
```

и

```
$$\frac{\dfrac{x2+1}2}{}
```

10.4.

```
$$
fg={\tsize\frac12}[(f+g)^2-f^2-g^2]
$$
```

Как и в упражнении 10.1, фигурные скобки вокруг `\tsize\frac12` вовсе не обязательны, поскольку `\tsize` не может нигде впоследствии изменить размер.

10.5.

$$f = \frac{1}{2}[(f+g)^2 - f^2 - g^2]$$

10.6.

$$\binom{n}{k/2}$$

или

$$\binom{n}{\frac{1}{2}k}$$

and

$$\frac{\binom{n}{k}}{2}$$

10.7. $\frac{\binom{n}{k}}{2}$

10.8. Первое сообщение — самое сложное из до сих пор виденных:

! Missing } inserted.

...
...

$\frac{1}{2} \rightarrow \{ \frac{1}{2}$
}

1.4 ... paragraph has $\frac{1}{2}a+b$ and $\frac{1}{2}a+b$
\$.

Исследуя, как обычно, две последние строки, видим, что \TeX не замечает пропуска } в первой формуле; ошибка была обнаружена только когда \TeX достиг } во второй формуле! Причина кроется в том, что { в первой формуле понуждает \TeX начать чтение второго аргумента $\frac{1}{2}$, и когда \TeX просматривает аргумент в управляющей последовательности, который начинается с {, он просто заглатывает буквы, пока не встретит закрывающую }; таким образом, \TeX думает, что вся строка

$a+b$ and $\frac{1}{2}a+b$

и является вторым аргументом для $\frac{1}{2}$! Затем \TeX транслирует $\frac{1}{2}$ в свои собственные термины (вот что закодировано в последних двух строках этого сообщения об ошибке). Наконец, когда \TeX попытается подставить ' $a+b$ ' и $\frac{1}{2}a+b$ ', он обнаружит первый \$ и поймет, что здесь что-то не так. Поэтому он вставит } справа перед знаком \$, который вернет сбалансированность.

Второе сообщение об ошибке

! Extra }, or forgotten \$.

$\frac{1}{2} \rightarrow \{ \frac{1}{2}$

1.4 ...paragraph has $\frac{1}{2}a+b$ and $\frac{1}{2}a+b$
\$.

аналогично тем, которые мы до сих пор получали, но снова за исключением того, что Т_ЭХ сначала транслирует `\frac` прежде чем обнаруживает неполадки. В этом случае Т_ЭХ просто опускает `}`, и получается $\frac{1}{a} + b$.

Последнее сообщение об ошибке

```
Runaway argument?
{a+b$.
! Paragraph ended before \frac was complete.
<to be read again>
\par
1.7
```

появилось потому, что Т_ЭХ опять просматривал второй аргумент `\frac`, который начинается с `{a+b`. На этот раз, однако, Т_ЭХ знает, что здесь что-то не так, поскольку он попадает на пустую строку `7`, обозначаемую `\par`, а аргумент для `\frac` не должен содержать `\par`. Если вы запросите помощь, Т_ЭХ сообщит

```
I suspect you've forgotten a '}', causing me to apply this
control sequence to too much text. How can we recover?
My plan is to forget the whole thing and hope for the best.
```

Т_ЭХ говорит это на полном серьезе! Если нажмете (carriage-return) Т_ЭХ просто проигнорирует все после `\frac`, в том числе и закрывающий `'$.'` в этом абзаце, так что вы получите сообщение

```
! Missing $ inserted.
<inserted text>
$
<to be read again>
\par
1.7
```

потому что Т_ЭХ'у нужно сбалансировать `$` перед `\frac`. Т_ЭХ затем завершит обработку файла, но если вы выведете его на печать, то увидите, что последняя формула просто исчезнет.

Однако, как и в ответе к упражнению 7.11, возникают дополнительные осложнения, если `\enddocument` встретится прежде, чем закончится абзац, содержащий оборванную формулу `$\frac{1}{a}=b$`. Вы получите сообщение

```
Runaway argument?
{a=b$.
! Forbidden control sequence found while scanning use of \frac.
<inserted text>
\par
<to be read again>
\enddocument
\enddocument
```

?

Снова \TeX изучает второй аргумент `\frac`, но теперь это управляющая последовательность `\enddocument`, а не конец абзаца, что заставляет \TeX встревожиться, что что-то неладно. `\enddocument` представляет собой специально набранную управляющую последовательность, которой запрещено появляться внутри аргумента другой управляющей последовательности. То же относится и к таким вещам, как `\proclaim` и `\demo` (исключая лишь, что `\proclaim` может появиться внутри `\abstract{...}`).

11.1.

- (1)
$$f \cdot g^{(n+1)}(a) = \sum_{k=0}^{n+1} \binom{n+1}{k} f^{(k)}(a) g^{(n+1-k)}(a)$$
- (2)
$$0 = (1+(-1))^n = \sum_{j=0}^n (-1)^j \binom{n}{j}$$
- (3)
$$\sum_{i=1}^n x_i^2 \cdot \sum_{i=1}^n y_i^2 = \sum_{i \neq j} x_i^2 y_j^2 + \sum_{i=j} x_i^2 y_j^2$$
- (4)
$$\sum_{i=1}^p \sum_{j=1}^q \sum_{k=1}^r a_{ij} b_{jk} c_{ki}$$
- (5)
$$\Sigma^2: [X, S_0(\infty)] \to [\Sigma^2 X, S_0(\infty)]$$
- (6)
$$\bigcup_{n=1}^m (A_n \cup B_n)$$
- (7)
$$X \setminus \bigcup_{i \in I} A_i = \bigcap_{i \in I} (X \setminus A_i)$$

Обратите внимание, что в формуле (5) стоит греческая Σ , а не \sum . Заметьте, что \TeX совершенно верно обрабатывает выражение $(1+(-1))$ в формуле (2).

11.2.

```


$$\sum_{\substack{1 \leq i \leq p \\ 1 \leq j \leq q \\ 1 \leq k \leq r}} a_{ij} b_{jk} c_{ki}$$


```

11.3. `\Sb` весьма сложно устроена и в данной ситуации \TeX окажется в полной растерянности; продолжая нажимать `(carriage-return)`, вы получите целый ряд сообщений об ошибках, завершающихся приглашением `*`, после того, как \TeX прочтет `\enddocument`, — ситуация, уже обсуждавшаяся в упр. 4.6.

11.4. Если вы наберете

```


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


```

то получите

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

Я почувствовал, что \bigcup слишком велик для этой ситуации, хотя его размер t -size ($y \bigcup$ нет меньшего размера s -size). С другой стороны, обычный \cup размера s -size слишком мал! Так что на самом деле я набрал

```


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


```

От вас, разумеется, не требуется предвидеть заранее такое исправление.

Заметим также, что Π — это обычная греческая буква Π . Ее не следует путать с \prod , еще одним большим оператором, который вы найдете в приложении F.

11.5.

```


$$\iint \lim_M d\omega = \iint \partial M \omega$$


```

12.1.

- (1) $y \, dx - x \, dy$
- (2) $x \, dy / dx$
- (3) $\int_1^x \frac{dt}{t}$
- (4) $\int \, dy = \int \frac{dy}{dx} \, dx$

12.2. Вы можете набрать

```


$$\frac{dz}{dx} = \frac{dz}{dy} \frac{dy}{dx}$$


```

используя \backslash , чтобы искусственно удлинить числитель и знаменатель.

12.3.

- (1) $1 \, \text{ml} \text{ equals } 1.00028 \, \text{cc}$
- (2) $g = 9.8 \, \text{m/sec}^2$
- (3) $0 \, \text{circ} \, C \text{ equals } 32 \, \text{circ} \, F$

[В (2) и (3) тонкая шпация могла быть помещена непосредственно перед знаками $\text{\$}$, а не сразу за ними.]

12.4.

```


$$\int_0^{\infty} \frac{t^{-ib}}{t^2 + b^2} e^{-iat} \, dt = e^{-ab} E_1(ab), \quad a, b > 0.$$


```

(Тем не менее в упр. 8.4 мы рекомендовали, чтобы во фразе типа ‘ $a, b > 0$ ’ в тексте набиралось $\text{\$a\$, \$b>0\$}$ для получения большего пробела после запятой. Но если выражение $a, b > 0$ входит в выключную формулу, то лучше об этом не беспокоиться вовсе.)

13.1.

(1)

Сумма векторов

\overrightarrow{OA} и \overrightarrow{OB} определяется как
 $\overrightarrow{OA} + \overrightarrow{OB} = \overrightarrow{OP}$,
 где $OAPB$ есть параллелограмм.

(2)

Для любого треугольника ABC справедливо неравенство
 $\overline{AB} + \overline{BC} > \overline{AC}$.

(3)

\sl Теорема Пифагора: Если $\angle C$ треугольника
 $\triangle ABC$ прямой, то
 $\overline{AB}^2 = \overline{AC}^2 + \overline{BC}^2$.

Для того чтобы получить общеупотребительный в геометрии символ \angle , мы использовали здесь `\angle`; другие варианты перечислены в приложении G. Вместо того чтобы применить `\Delta` для получения Δ , мы взяли `\triangle` и получили Δ , который в этом контексте смотрится лучше. (Имеются также `\bigtriangleup` и `\bigtriangledown`, которые дают Δ и ∇ , представляющие собой бинарные операторы. Не переживайте особенно, если вы упустите из виду тот или иной из этих замечательных символов — математики непрерывно приспособливают старые знаки к новым условиям. Вам за ними все равно никогда не угнаться — просто пусть они яснее выражают свои намерения.)

13.2.

$$B_{i,j} \longleftarrow A_{i+j, \overline{i+j}}$$

13.3.

Корни *\sl квадратного трехчлена* $ax^2+bx+c=0$ задаются формулой

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}.$$

13.4. В каждом из этих случаев \TeX сначала выдаст сообщение об ошибке, в котором говорится о том, что вставлена `{`; \TeX знает, что это необходимо, так как `\overline` и `\sqrt` суть команды для обработки символов — сами они не могут служить индексами. В сообщении по поводу второй формулы содержатся несколько дополнительных строк, которые мы, как обычно, благоразумно игнорируем, потому что `\sqrt` был в сущности получен (каким-то мистическим образом) из более элементарной управляющей последовательности `\radical`. В конце каждой формулы \TeX выдаст сообщение об ошибке, гласящее о том, что вставлена `}`.

13.5.

(1)

Для $F=0$ получается ‘вырожденная гипербола’

\$\$

$$x + \frac{B\sqrt{2A}}{y + \frac{D\sqrt{2C}}{A}} = \pm \sqrt{\frac{-C}{A}}$$

\$\$

(2)

F_n – е число Фибоначчи F_n задается формулой

\$\$

$$F_n = \frac{\left(\frac{1 + \sqrt{5}}{2}\right)^n - \left(\frac{1 - \sqrt{5}}{2}\right)^n}{\sqrt{5}}$$

\$\$

13.6.

Для $0 < x \leq 1$ имеем

$$f(x) = \frac{1}{\left[\frac{1}{x}\right]}$$

13.7.

\$\$

$$\int_0^1 |f|_p = \int_0^1 |f|^p = \frac{1}{p+1}$$

\$\$

13.8.

$$\left| \frac{dx}{|x|} \right| = \frac{1}{|x|} |dx|$$

13.9.

$$\int_{-\infty}^{\infty} \left[\int_{-\infty}^{\infty} f(x) dx \right]^2 dx$$

13.10.

Связка ∇ сравнима с метрикой тогда и только тогда, когда

$$\frac{d}{dt} \langle V, W \rangle = \langle \frac{dV}{dt}, W \rangle + \langle V, \frac{dW}{dt} \rangle$$

Обратите внимание, что для получения меньших $\langle \rangle$ мы использовали $\langle \rangle$ и $\langle \rangle$, а не $\langle \rangle$ и $\langle \rangle$, но для большего размера можно использовать $\langle \rangle$ и $\langle \rangle$.

13.11.

Мы вывели формулу корней квадратного трехчлена,
 ‘‘дополняя до полного квадрата’’:

$$\begin{aligned} \text{\$}\text{\$}ax^2+bx+c=a\text{\left}(x^2+\text{\frac}{a}bx + \text{\frac}{a}ca \text{\right)} \\ =a\text{\left}[\text{\left}(x+\text{\frac}{2a}b\text{\right)}^2+\text{\left}(\text{\frac}{a}ca \\ -\text{\frac}{b^2}{4a}\text{\right)}\text{\right}].\text{\$}\text{\$} \end{aligned}$$

13.12.

$$\text{\$}\text{\$}\text{\left}(\text{\frac}{1}{\sqrt{A}}+\text{\sum}\text{\limits}_{i=1}^n a_i\text{\right)} \\ \text{\left}(\text{\frac}{1}{\sqrt{B}}+\text{\sum}\text{\limits}_{i=1}^n b_i\text{\right)}\text{\$}\text{\$}$$

Обратите внимание, что первый `\tsize` не влияет на `\frac{1}{\sqrt{B}}`, стоящий за первым `\right)`, так что заключать это в фигурные скобки нет необходимости.

13.13. Вы получите сообщение об ошибке

```
! Missing delimiter (. inserted).
<to be read again>
```

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

Поскольку `,` — недопустимый ограничитель, `TeX` вставит для вас какой-нибудь иной ограничитель; наиболее для него очевидный — это `.`, поскольку `TeX` не может знать, что у вас на уме.

13.14. Вы получите сообщение об ошибке

```
! Extra \right.
\frac{dx^2}{dx}\right|
      _{x=a}=2a\right\}
```

Если вы попросите помощи, то обнаружите, что `TeX` просто опустит `\right`, поскольку для этого ограничителя нет парного `\left`.

13.15. Если вы наберете

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

то получите

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

Это один из тех случаев, обсуждаемых в части 3, когда вы можете пожелать подправить пробел после того, как просмотрите распечатку. Эта формула в тексте была набрана так:

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

13.16.

$$x^2 \left/ \frac{c+1}{d} \right.$$

Заметим, что эта формула выглядит не так красиво, как предыдущая, несмотря на то, что \TeX следовал тем же правилам расстановки пробелов. Но здесь одна из тех ситуаций, когда нужно кое-что подправить: удалите тонкую шпацию после x^2 , в результате чего формула превратится в

$$x^2 / \frac{c+1}{d}$$

На самом деле и здесь, и в формуле на с. 87 можно было бы обойтись без конструкции $\left. \dots \right.$, потому что имеются ограничители \biggl и \biggr (см. с. 88), которые дают правильный размер. Вообще-то у \TeX 'а есть еще и \bigg для получения ограничителей этого размера, которые действуют как ординарные символы (см. часть 3), и в подобных случаях следует предпочесть их.

13.17.

- (1)
$$\biggl(\sum_{i=1}^n x_{iy_i} \biggr)^2 = \sum_{i=1}^n (x_{iy_i})^2 + \sum_{i \neq j} x_{iy_ix_jy_j}$$
- (2)
$$\pi(n) = \sum_{m=2}^n \left\lfloor \frac{n}{m} \right\rfloor \biggl(\sum_{k=1}^{m-1} \left\lfloor \frac{m}{k} \right\rfloor \left\lceil \frac{m}{k} \right\rceil \biggr) \right\rfloor$$
- (3)
$$m^* \left(A \cap \bigcup_{i=1}^n E_i \right) = \sum_{i=1}^n m^* (A \cap E_i)$$

Обратите внимание на тонкую шпацию после \biggl (в (2)) и после $\biggl[$ (в (3)). Тонкие шпации часто бывают полезны, когда ограничитель \bigg соседствует с оператором, пределы которого выходят за габариты ограничителя.

13.18.

- (1)
$$\left(\sqrt{\frac{AB}{A+B}} + \sum_{i=1}^n N a_i \right)^2$$
- (2)
$$f \left(\sum_{i=1}^n p_{ix_i} \right) \leq \sum_{i=1}^n p_{if}(x_i)$$

В первой формуле ограничители \bigg оказались бы недостаточно большими для фрагмента формулы $\sqrt{\frac{A}{B}}$. Во второй формуле — это, скорее, дело вкуса: так как в выражениях типа $f(x)$ скобки играют особую роль, кажется более естественным аккуратно заключить в скобки все выражение $\sum_{i=1}^n p_i x_i$, выступающее здесь в качестве аргумента.

14.1.

`$$g(x)=f(x+\text{constant})+f(x-\text{constant})$$`

14.2. Можно поменять шрифт внутри `\text`, но сам `\text` не работает как заменитель шрифта — это просто управляющая последовательность с одним аргументом, как было сказано чуть выше. Ввод

`$$y=f(x+\text{constant})$$`

произведет

$$y = f(x + constant)$$

потому что Т_ЕX подумает, что аргументом `\text` является просто `c`, и фигурные скобки просто формируют несущественную группу.

14.3.

`$$F_n=F_{n-1}+F_{n-2}\quad`
`\text{для {\sl любого\}} $n>1$.$$`

(или вы можете набрать

`... \text{for {\sl every\}} $n>1$.$$`

с точкой в конце всего предложения.

Вам следует действовать методично, чтобы получить знаки `$` и `}` в соответствующем порядке; вероятно, вам поможет, если вы оставите больше пространства перед `$$`:

`... \text{\for {\sl every\}} $n>1$.` `$$`

Вы, разумеется, уже в хорошей форме, если научились помещать знаки `$$` в отдельной строке:

`$$`
`... \text{\for {\sl every\}} $n>1$.`
`$$`

14.4.

`$1\,\text{ml}=1.000028\,\text{cc}$`
`$0^\text{circ}\,\text{C}=32^\text{circ}\,\text{F}$`

То, что формула целиком помещена между знаками `$`, позволяет получить правильные пробелы вокруг знаков `=`.

14.5. Прежде всего вы получите сообщение об ошибке

```
! Missing } inserted.
<inserted text>
}
<to be read again>
$
1.6 \Gamma(n)=(n-1)! \quad \{\text{text когда } n\$ \text{ целое}\}
?
```

TeX решил уже, что аргументом `\text` служит просто следующая буква `k` и что оставшийся ввод должен быть опять в математическом режиме. Так что когда он видит знак `$` от выражения `$n$`, он думает, что вы пытались завершить математический режим. Но тогда `{` перед `\text` кажется несбалансированной и TeX вставляет фигурную скобку, потому что полагает, что пропущена `}`. Если набрать `h`, TeX сам скажет вам об этом, правда, не столь пространно.

Если вы в этом месте нажмете `(carriage-return)`, то увидите, что как только TeX пошел по этому пути, ситуация еще больше запуталась. Коль скоро вы вошли в режим “выключных формул” посредством `$$`, то и выйти из него нужно тоже через `$$`. Но TeX увидел лишь один `$` перед `n`, поэтому создал другое сообщение об ошибке:

```
! Display math should end with $$ .
<to be read again>
n
1.6 \Gamma(n)=(n-1)! \quad \{\text{text когда } n\$ \text{ целое}\}
?
```

Снова нажимаете `h` и TeX излагает вам это на свой манер. Если вы опять нажмете `(carriage-return)`, TeX решит, что вы начали новую математическую формулу `$` целое, и пожалуется, когда встретит `}`:

```
! Extra }, or forgotten $.
1.6 ...n-1)! \quad \{\text{text когда } n\$ \text{ целое}\}
?
```

Если снова нажать `(carriage-return)`, то TeX либо удалит `}`, либо вставит `$`, чтобы закончить математическую формулу. Нажав `h`, вы увидите, что на самом деле TeX удаляет `}`. Это позволяет ему преодолеть формулу, но теперь он думает, что находится в математическом режиме. Если непосредственно за этой формулой следует `\enddocument`, TeX должен будет вставить еще один знак `$`, прежде чем закруглиться.

14.6.

Имеем

$$\frac{(n+1)^{p+1}}{p+1} = \sum_{k=1}^n k^p + \text{члены, содержащие } \sum_{k=1}^n k^r \text{ для } r < p.$$

По индукции выводим, что

$$\frac{\sum_{k=1}^n k^p}{n^{p+1}} = \frac{1}{p+1} + \text{члены с отрицательными степенями } n.$$

14.7. Первая формула была набрана как

$$X_n = X_k \quad \text{if and only if} \quad Y_n = Y_k \quad \text{and} \quad Z_n = Z_k$$

(Вокруг ‘and’ был использован только один `\quad`, потому что стоящая справа фраза ‘ $Y_n = Y_k$ and $Z_n = Z_k$ ’ представляет собой в сущности одно выражение, которое отделяется от левой части формулы посредством `\quad`.)

Вторая формула была набрана так:

$$Y_n = X_{n+1} \quad \text{and} \quad Z_n = X_{n-1} \quad \text{for all } n \geq 0.$$

(В этом случае дополнительное условие ‘for all $n \geq 0$ ’ относится ко всей формуле, поэтому оно отделено от левой части на расстояние `\quad`, тогда как отдельные члены левой части разделены пробелом `\quad`.)

14.8.

Каждый хотел бы знать, будет ли множество $\{p, p+2\}$ простых, бесконечным или нет!

Такой скользкий подход предполагает некоторую возню с пробелами и с тем, чтобы запретить нежелательные разрывы строки. Обратите внимание на тильду и отметьте, что после `простые` нет пробела.

14.9.

the n^{th} Fibonacci number F_n

14.10.

$$\sum_{\text{\$1\$ odd}} \binom{n}{1} = 2^{n-1}$$

14.11. $f^{\text{(iv)}}$, $f^{\text{(v)}}$, $f^{\text{(vi)}}$, и т. д.

14.12. Как и в упр. 13.4, \TeX сначала вставит скобку $\{$, а затем $\}$. Из первого сообщения об ошибке выясняется также, что `\text` — довольно сложный инструмент.

14.13.

```


$$\frac{\max_{1 < n < m} \log_2 P_n}{\lim_{x \rightarrow 0} \frac{\sin x}{x}}$$


```

14.14.

(1)

```


$$\lim_{x \rightarrow 0} \frac{\sin^2 ax}{\sin^2 bx} = \left(\frac{a}{b}\right)^2$$


```

(2)

```


$$\lim_{x \rightarrow \infty} x \sin \frac{1}{x} = \lim_{x \rightarrow \infty} \frac{\sin \frac{1}{x}}{\frac{1}{x}} = \lim_{x \rightarrow 0^+} \frac{\sin x}{x}$$


```

Обратите внимание, что для получения верхнего индекса $+$ используется `x\to0^+`; некоторые авторы предпочитают `\lim_{x \rightarrow 0+}`, что достигается посредством `x\to0+`.

(3)

```


$$\frac{1}{2} \cos x + \cos 2x + \dots + \cos nx = \frac{\sin(n + \frac{1}{2})x}{2 \sin \frac{x}{2}}$$


```

(4)

```


$$(\log \circ f)' = f'/f$$


```

(5)

```


$$\lim_{x \rightarrow 0^+} x (\log x)^n = 0$$


```

(6)

```


$$\lim_{h \rightarrow 0^+} \int_{-1}^{-1+h} \frac{1}{h^2+x^2} = \left. \lim_{h \rightarrow 0^+} \arctan \frac{x}{h} \right|_{-1}^{-1+h} = \pi$$


```


(7)

$$\begin{aligned} & \arctan\left(\frac{1}{2}\right) + \arctan\left(\frac{1}{3}\right) \\ &= \arctan\left(\frac{\frac{1}{2} + \frac{1}{3}}{1 - \frac{1}{2} \cdot \frac{1}{3}}\right) = \arctan\left(\frac{5}{1}\right) = \frac{\pi}{4} \end{aligned}$$

(8)

$$\lim_{n \rightarrow \infty} \sum_{n \text{ even}} \frac{2 - a_n^{-2}}{1 + a_n} = \lim_{m \rightarrow \infty} \frac{2 - m^{-2}}{1 + m}$$

14.15. Она была набрана следующим образом:

$$\begin{aligned} \Delta &= \min\left(\sin^2\left(\frac{1}{\min(1, \epsilon/10)}\right)^9, \min(1, \epsilon/10), \min(1, \epsilon/6)\right) \end{aligned}$$

(За последней запятой вставлены две тонкие шпации, потому что она завершает довольно сложное выражение)

14.16.

$$\int_{\pi/2 - \Delta}^{\pi/2 + \Delta} \frac{\sin x}{x} dx \approx \frac{2\Delta}{\pi/2 + \Delta}$$

и

$$\begin{aligned} & \lim_{x \rightarrow \infty} \frac{\int_x^{\infty} \frac{\log t}{2t} e^{-t^2} dt}{e^{-x^2}} = \lim_{x \rightarrow \infty} \frac{e^{-\left(x + \frac{\log x}{2}\right)^2}}{e^{-x^2}} \\ &= \frac{1}{2} \end{aligned}$$

14.17.

$$\frac{\max_{1 < n < m} \log_2 P_n}{\lim_{x \rightarrow 0} \frac{d}{dx} \sin x}$$

14.18. $\operatorname{arg}\sinh$

14.19. Наберите что-то вроде

$$\begin{aligned} & \$\operatornamename{\text{\sl SO}}(n)\$ && SO(n) \\ & \$\operatornamename{\text{\bf SO}}(n)\$ && SO(n) \end{aligned}$$

15.1. О набирает

$$$$x=y\tag{(3-2)},$$$$

Теперь у метки появились дополнительные скобки. Более того, запятая оказалась частью метки, а не самой формулы.

15.2. Если вы хотите получить A' с курсивным A , как в математическом режиме, наберите

$$$$ \dots\tag{\textit{A}'}$$$$

Фигурные скобки вокруг \textit{A}' помогают $\text{T}_{\text{E}}\text{X}$ 'у правильно определять, какой из знаков $\$$ представляет собой часть метки, а какой указывает конец формулы. (На самом деле в случае $\dots\tag{\textit{A}'}\dots$ получилось бы то же самое — формула завершится, только когда $\text{T}_{\text{E}}\text{X}$ наткнется на комбинацию $$$$ без пробела между двумя знаками $\$$ — но не стоит создавать себе дополнительные трудности.)

Если же нужна метка A' с прямым A , наберите

$$$$ \dots\tag{A\textit{'}}$$$$

заклучив в фигурные скобки только ту часть метки, которая должна быть представлена в математическом режиме. Можно сделать и так:

$$$$ \dots\tag{\textit{\text{A}'}}$$$$

⚠ Если у вас переизбыток меток со штрихами и верхними индексами, и не так уж много с дефисами и тире, то вы предпочтете, чтобы $\text{A}_{\text{M}}\text{S-}\text{T}_{\text{E}}\text{X}$ обрабатывал все эти метки как математические формулы, а не как текст. В части 3 объясняется, как это организовать.

15.3. Следует набрать

$$$$ \dots \tag{\$*}\$$$

чтобы получить $(*)$; если вы наберете просто $\tag{*}$, то получите метку $(*)$ с приподнятой звездочкой, взятой из обычного шрифта для текста. Так как $*$ — это бинарная операция в математическом режиме, вам следует, если используется более одной звездочки, заключать их в фигурные скобки, чтобы предотвратить появление дополнительного пробела, который $\text{T}_{\text{E}}\text{X}$ может вставить. (На самом деле $\dots\tag{\$**}\$$ случайно сработает правильно, но $\dots\tag{\$***}\$$ даст уже метку $(***)$. Вместо того чтобы понапрасну волноваться, набирайте просто $\dots\tag{\$*}{*}\$$ и $\dots\tag{\$*}{*}{*}\$$, чтобы себя обезопасить.)

15.4.

```


$$Q^l = Q_1 \sum_k (-1)^k (PQ_1 - I)^k$$


$$Q^r = \sum_k (-1)^k (Q_1 P - I)^k Q_1$$


```

15.5. Вы получите

$$(1_l) \quad Q^l = Q_1 \left\{ \sum_k (-1)^k (PQ_1 - I)^k \right\}$$

$$(1_r) \quad Q^r = \left\{ \sum_k (-1)^k (Q_1 P - I)^k \right\} Q_1$$

Поскольку вы пропустили $\&$, TeX просто вставляет пустую формулу за второй формулой, так что вся вторая формула будет воспринята как левая часть, а правая часть будет пустой.

15.6.

```


$$\alpha_4 = \sqrt{\frac{1}{2}}$$


$$\alpha_8 = \sqrt{\frac{1}{2} + \frac{1}{2} \sqrt{\frac{1}{2}}}$$


$$\alpha_{16} = \sqrt{\frac{1}{2} + \frac{1}{2} \sqrt{\frac{1}{2} + \frac{1}{2} \sqrt{\frac{1}{2}}}}$$


```

15.7.

```


$$k_1, k_2 = \sqrt{H^2 - K} \quad \text{where} \quad K = \frac{(eg - f^2)(EG - F^2)}{Eg - 2Ff + Ge} \cdot \frac{1}{2(EG - F^2)}$$


```

15.8.

```


$$K = \frac{(eg - f^2)(EG - F^2)}{Eg - 2Ff + Ge} \cdot \frac{1}{2(EG - F^2)}$$


```

15.9. Ситуация аналогична рассмотренной в упр. 11.3.

15.10.

```


$$g = \det(g_{ij})$$

entry of the inverse matrix
of  $(g_{ij})$ 

```

15.11.

Имеем $(a+bi)^2 = \alpha + \beta i$ тогда и только тогда, когда

```


$$a^2 - b^2 = \alpha, \quad 2ab = \beta,$$


```

Решением последнего служат

```


$$a = \sqrt{\frac{\alpha + \sqrt{\alpha^2 + \beta^2}}{2}}, \quad b = \frac{\beta}{2\sqrt{\alpha + \sqrt{\alpha^2 + \beta^2}}}$$


$$a = -\sqrt{\frac{\alpha + \sqrt{\alpha^2 + \beta^2}}{2}}, \quad b = \frac{-\beta}{2\sqrt{\alpha + \sqrt{\alpha^2 + \beta^2}}}$$


```

16.1. Вы можете представлять ее в виде трех выравниваемых формул, причем левые части второй и третьей формул — пустые. Таким образом, вы могли бы набрать

```


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


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


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


```

с пустыми формулами перед вторым и третьим $=$ (вы могли даже набрать $\{ \}$, если пустота вызывает у вас чувство неудовлетворения.)

16.2. Вы могли использовать

```


$$(a+b)(a+b) = a^2 + 2ab + b^2$$


$$(a+b)(a-b) = (a+b)a - (a+b)b$$


$$= a^2 + ab - ab - b^2$$


$$= a^2 - b^2$$


```

\split устроен так, что знаки = внутри \split ... \endsplit, будут автоматически выровнены со знаками = из \align.

16.3. В этой формуле вторая строка представляет собой продолжение члена $(a_{11} + b_{11} + c_{11})$, так что вместо того, чтобы вторую строку отодвинуть от знака = на два квадрата, следует отодвинуть ее на два квадрата от левой скобки. Вот как выглядел набор:

```


$$\Delta = [a+b+c]^n (a_{11}+b_{11}+c_{11}) \quad \& \quad a_{12}+b_{12}+c_{12} \quad \& \quad a_{22}+b_{22}+c_{22}$$


```

16.4.

```


$$f^{(k)}(x) = e^{-1/x^2} \begin{cases} \frac{a_i x^i \sin \frac{1}{x}}{\cos \frac{1}{x}} \\ \frac{b_i x^i \cos \frac{1}{x}}{\sin \frac{1}{x}} \end{cases}$$


```

(Эта выключная формула смотрелась бы лучше, если бы между строками была заложена дополнительная отбивка; подробности см. в части 3.)

17.1.

```


$$\begin{pmatrix} a & b \\ c & d \end{pmatrix} \cdot \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} = \begin{pmatrix} a \cdot 0 + b \cdot 1 & a \cdot 1 + b \cdot 0 \\ a \cdot 1 + d \cdot 1 & c \cdot 1 + d \cdot 0 \end{pmatrix} = \begin{pmatrix} b & a \\ a+d & c \end{pmatrix}$$


```

(Вероятно, здесь было бы лучше заложить небольшую дополнительную отбивку между строками; такие усовершенствования обсуждаются в части 3.)

17.2. Можно набирать

```


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


```

снабжая каждый элемент второго столбца пробелом в один квадрат слева. (Вскоре мы изучим другой способ изменять расстояние между столбцами.)

17.3. Трюк состоит в том, чтобы поместить отточия в свои собственные строки и столбцы:

```


$$\begin{matrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{matrix}$$


```

17.4.

```


$$\begin{matrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{matrix}$$


$$\begin{matrix} \cos\phi & \sin\phi \\ -\sin\phi & \cos\phi \end{matrix}$$


$$\begin{matrix} \cos\rho & \sin\rho \\ -\sin\rho & \cos\rho \end{matrix},$$


$$\rho = \theta + \phi.$$


```

17.5. Мы можем представить его в виде матрицы с двумя столбцами:

```


$$\begin{matrix} 3.14159 \\ 2.71828 \\ 1.61808 \\ .57701 \end{matrix}$$


```

Но это кажется излишне замысловатым. На самом деле все цифры (в большинстве шрифтов) имеют одинаковую ширину, и именно по этой причине столбцы цифр хорошо выстраиваются. Если бы у нас была управляющая последовательность `\dwidth` для введения пробела именно такой ширины, то можно было бы обойтись только одним столбцом:

```


$$\begin{matrix} 3.14159 \\ 2.71828 \\ 1.61808 \\ \dwidth.5770 \end{matrix}$$


```

В части 3 объясняется, как может быть определена `\dwidth`.

17.6.

```


$$\det \begin{vmatrix} c_0 & c_1 & c_2 & \dots & c_n \\ c_1 & c_2 & c_3 & \dots & c_{n+1} \\ c_2 & c_3 & c_4 & \dots & c_{n+2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ c_n & c_{n+1} & c_{n+2} & \dots & c_{2n} \end{vmatrix} > 0.$$


```

Обратите внимание, что `&&` просто обозначает пустую формулу в соответствующей позиции (можете набирать `&{}&`, если вам от этого легче).

17.7. Первым попопзновением будет сделать так:

```


$$\omega = \begin{pmatrix} 0 & \omega_{12} & 0 & \dots & 0 \\ -\omega_{12} & 0 & \omega_{23} & 0 & \dots & 0 \\ 0 & -\omega_{23} & 0 & \omega_{34} & 0 & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots & & \\ 0 & \dots & 0 & \dots & \omega_{n-1,n} \\ 0 & \dots & 0 & -\omega_{n-1,n} & \dots & 0 \end{pmatrix}$$


```

А на самом деле упр. 17.7 было набрано следующим образом:

```


$$\omega = \begin{pmatrix} \quad & \quad & \quad & \quad & \quad & \quad \\ \quad & \quad & \quad & \quad & \quad & \quad \\ \quad & \quad & \quad & \quad & \quad & \quad \\ \dots & & & & & \\ \quad & \quad & \quad & \quad & \quad & \quad \end{pmatrix}$$


```

чтобы формула смотрелась лучше.

17.8.

```


$$\begin{aligned} f^{(k)}(x) &= ax^{-m-k} \sin \frac{1}{x} \\ &+ \sum_{l=k+1}^{\infty} \frac{1}{x^{2k-1}} \left( a_l x^{m-1} \sin \frac{1}{x} \right. \\ &\quad \left. + b_l x^{m-1} \cos \frac{1}{x} \right) \\ &\begin{cases} x^{-m-2k} \sin \frac{1}{x}, & \text{если } k \text{ четное} \\ x^{-m-2k} \cos \frac{1}{x}, & \text{если } k \text{ нечетное} \end{cases} \end{aligned}$$


```

(Мы воспользовались `\dfrac`, потому что строки `\cases` обычно бывают размера `t-size`.)

18.1.

$$\sqrt{ab + \sqrt{\gamma^2}} \leq x + \frac{ab}{\gamma}$$

18.2. $\mathbb{2}^{\{ab\}}$ и $\Gamma_{\{ab\}}$. Заметьте, что вокруг $\backslash ab$ необходимы фигурные скобки (ср. со с. 69): если просто набрать $\mathbb{2}^{\backslash ab}$, Т_ЭX оттранслирует это в $\mathbb{2}^{\backslash \alpha^2 + \backslash \beta^2}$ и пожалуется на двойной верхний индекс, а выражение $\Gamma_{\backslash ab}$ оттранслирует в

$$\Gamma_{\alpha^2 + \beta^2}$$

Если бы мы ввели определение

$$\backslash \text{define} \backslash ab \{ \{ \alpha^2 + \beta^2 \} \}$$

с двумя парами фигурных скобок, то $\mathbb{2}^{\backslash ab}$ было бы оттранслировано в

$$\mathbb{2}^{\{ \alpha^2 + \beta^2 \}}$$

что позволило бы избежать ряда проблем. Такого рода дополнительные скобки — неплохая идея, но будьте бдительны, дабы не добавить их в неподходящем месте. Например, вы можете попробовать дать такое определение

$$\backslash \text{define} \backslash Ne \{ \{ ne \} \}$$

чтобы $\$A_Ne\$$ работало правильно (ср. со с. 69). К сожалению, $\$x_Ne Y\$$ дал бы в этом случае неправильный пробел (ср. с упр. 7.6).

18.3. Теперь Т_ЭX заменит $\backslash ab$ на $\sqcup \backslash \alpha^2 \sqcup \backslash \beta^2 \sqcup$, но это совершенно несущественно, поскольку $\backslash ab$ будет использоваться только в математическом режиме, где пробелы никакой роли не играют.

18.4.

The $\backslash \text{deRham} \backslash$ cohomology ring $\mathbb{H}^*(G)$
had, of course, already essentially been
computed by $\backslash \text{Cartan}$.

(Обратите внимание, что совсем не думать нельзя — следует помнить о $\backslash \sqcup$ после $\backslash \text{deRham}$.)

18.5. Если ввести $\backslash \text{define} \backslash \text{должны} \{ \{ \backslash \text{bf} \text{ должны} \} \}$, то Т_ЭX заменит $\backslash \text{должны}$ на $\backslash \text{bf} \text{ должны}$, так что ‘должны’ и все, что за ним стоит станет полужирным! Правильный ответ такой:

$$\backslash \text{define} \backslash \text{должны} \{ \{ \backslash \text{bf} \text{ должны} \} \}$$

с дополнительными фигурными скобками, так что $\backslash \text{должны}$ заменится на $\{ \backslash \text{bf} \text{ должны} \}$. Тогда ввод вроде

Вы $\backslash \text{должны} \backslash$ помнить о фигурных скобках!

будет оттранслирован в

Вы `\bf \должны\` помнить о фигурных скобках!

где только ‘должны’ станет полужирным. Вам, однако, все еще нужен `_` после `\должны` — Т_ЭX, как только прочтет `\должны`, проигнорирует пробел после него, так что этот пробел был проигнорирован прежде чем Т_ЭX оттранслировал `\должны` в `\bf \должны`¹.

18.6.

`$$\a~\a+\a~{\a+\a}+\a~{\a~\a}+\sqrt{\a^2+\a^3}$$`

18.7. Встретив `\ab`, Т_ЭX заменил бы его на `\a^2+\beta^2`, а затем вместо `\a` подставил бы `\alpha`. Так что `\ab` фактически было бы заменено на `\alpha^2+\beta^2`. Таким образом, сухой остаток такой же, как и при

`\define\ab{\alpha^2+\beta^2}`

Аббревиатуры типа `\a` для `\alpha` так удобны потому, что вы можете использовать их повсеместно. Приятно знать, что вы можете определить новую управляющую последовательность вроде `\ab` в терминах `\a`, не возвращаясь к старому имени `\alpha`.

18.8. Можно было бы набирать `\a_`, чтобы получать в тексте пробелы после α , но беда в том, что нет удобного способа набирать $\alpha + 1$. Если вы наберете `$$\a+1$`, то первый знак `$` говорит Т_ЭX’у, что он должен войти в математический режим, но тот знак `$`, с которым он столкнется после раскрытия `\a`, даст команду выйти из математического режима. Тогда непонятно, как работать с `\alpha` вне математического режима!

Если на самом деле вам нужен способ набора одной и той же управляющей последовательности `\a` в текстовом и математическом режимах, вы можете определить `\define\a{\text{\$alpha$}}`, поскольку `\text` может использоваться в обоих случаях и с ним происходят все необходимые изменения размеров для верхних и нижних индексов. (Дополнительная пара фигурных скобок обеспечивает вам возможность набирать `$$2^{\a$}`.)

18.9. Если попытаться ввести

`\define\c{\gamma}`

это приведет к сообщению об ошибке, потому что `\c` уже определялось — это управляющая последовательность для “cedilla” как в слове ‘*façade*’. Так что следует предпринять что-то вроде

`\predefine\cedilla{\c}`

`\redefine\c{\gamma}`

Таким образом, введение `$$\c$` для γ логически обосновано, причем можно набирать `fa\cedilla cade`, когда нужен правильный *façade*.

¹К сожалению, подобные упражнения с русским языком практически бессодержательны из-за многообразия форм, которые могут принимать русские слова. Т_ЭX не способен понять, например, что ‘должны’ и ‘должен’ суть формы одного слова (ср. с ответом к упр. 3.3), так что если вы будете следовать данному совету без должной доли сомнения, вам придется написать с десяток различных форм слова. В оригинале, разумеется, речь шла о слове ‘must’. — Прим. ред.

18.10. Получится vicious circle — порочный круг: управляющая последовательность `\vicious` заменяется на `\circle`, которая в свою очередь заменяется на `\vicious`, а та в свою очередь — на `\circle`, ... Чтобы TeX вышел из порочного круга, нужно удалить причину его возникновения.

18.11. Одно из средств такое:

```

$$
\define\1{\(\dfrac{1+\sqrt{5}2\)}
\define\2{\(\dfrac{1-\sqrt{5}2\)}
\define\5{\sqrt{5}
\align
a_n &=a_{n-1}+a_{n-2}\
&=\frac{1^{n-2} - 2^{n-2} + 1^{n-1} - 2^{n-1}}{5}\
&=\frac{1^{n-2}\(1+\dfrac{1+\sqrt{5}2\)}
- 2^{n-2}\(1+\dfrac{1-\sqrt{5}2\)}{5}\
&=\frac{1^{n-2}1^2 - 2^{n-2}2^2}{5}\
&=\frac{1^n - 2^n}{5}.\endalign
$$

```

Такие новые управляющие последовательности, как `\1` и `\2` экономят львиную долю набора, и еще немного помогает `\5`. После того, как выключная формула набрана, последовательности `\1`, `\2` и `\5` опять неопределены и при каком-нибудь непредвиденном случае могут быть определены снова посредством `\define`.

На самом деле автор предпочитает набирать

```

$$
\redefine\+{\(\dfrac{1+\sqrt{5}2\)}
\redefine\-\{\(\dfrac{1-\sqrt{5}2\)}
\define\5{\sqrt{5}

```

так как названия `\+` и `\-` лучше запоминаются. `\redefine` нужно было использовать потому, что `\+` и `\-` уже были определены (`\-` — “факультативный перенос” из гл. 3, а `\+` средство TeX’a, которое мы не обсуждали в этом учебнике, поскольку в математических текстах оно редко бывает нужно). Так как `\redefine` находится внутри знаков `$`, старые определения восстановятся, как только выключная формула закончится.

18.12. `$$\deriv{x^2}$$` и `$$\deriv{(x^2+x^3)}$$`. Если бы мы в первой формуле набрали просто `$$\deriv x^2$$`, то получили бы

$$\frac{dx^2}{dx},$$

а `$$\deriv (x^2+x^3)$$` дало бы нечто совсем непонятное, что вы даже не способны были придумать.

18.13. Наберите

```
\define\vector#1{(#1_1,\dots,#1_n)}
```

и затем используйте `$\vector x` для (x_1, \dots, x_n) и `$\vector y` для (y_1, \dots, y_n) , и т. д.

18.14. $\vec{\alpha}$ и $\vec{x'}$.

18.15. Получить (x'_1, \dots, x'_n) можно, набрав $\vec{\{x'\}}$; теперь аргументом является $\{x'\}$, и $\{x'\}_1$ дает x'_1 , и т. п. С другой стороны, при помощи $\vec{\text{vector}}$ нельзя получить формулу (x'_1, \dots, x'_n) — придется набирать ее полностью.

18.16. Наученные горьким опытом упражнения 18.2, определим

```
\define\power#1{2^{#1}}
```

Тогда $\text{\power } x$ производит 2^x , а $\text{\power } \alpha$ производит 2^α . Более того, $\text{\power } \{x+y\}$ дает 2^{x+y} , как и должно быть. Заметим, было бы неверно определять

```
\define\power#1{2^#1}
```

Фигурные скобки вокруг $x+y$ в вводе $\text{\power } \{x+y\}$ просто говорят Т_EX'у, что аргументом \power служит вся формула $x+y$; так что при данном выше неправильном определении $\text{\define } \text{\power}$ будет заменять $\text{\power } \{x+y\}$ на 2^{x+y} .

18.17. Поскольку большие дроби имеют один и тот же знаменатель, можно поступить так:

```
$$
\redefine\+{\(\dfrac{1+\sqrt{5}2\)}
\redefine\-\{\(\dfrac{1-\sqrt{5}2\)}
\define\fracc#1{\frac{#1}{\sqrt{5}}}
\align
a_n &=a_{n-1}+a_{n-2} \\
&=\fracc{\+^{n-2} - \-^{n-2} + \+^{n-1} - \-^{n-1}} \\
&=\fracc{\+^{n-2}\(1+\dfrac{1+\sqrt{5}2\)}
- \-^{n-2}\(1+\dfrac{1-\sqrt{5}2\)} \\
&=\fracc{\+^{n-2}\+^2 - \-^{n-2}\-^2} \\
&=\fracc{\+^n - \-^n} .\endalign
$$
```

18.18. Можно набирать $\text{\pd } f\{x_1\}$, $\text{\pd } g\{x_2\}$ и т. д., но если ввести определение

```
\define\pd#1#2{\dfrac{\partial#1}{\partial x_{#2}}}
```

то можно набирать просто $\text{\pd } f$, $\text{\pd } g$, и т. д. Заметьте (ср. с упр. 18.16), что в формулах типа $\frac{\partial f}{\partial x_{n+1}}$ вокруг того, что обозначено #2, нужны фигурные скобки.

18.19. Если так определить $\text{\define } \text{\sqrts}$, то с $\text{\sqrts } ab$ будет все в порядке, но если вы захотите иметь $\sqrt{a+b} + \sqrt{c+d}$, то нельзя набирать

```
\sqrts\{a+b}\{c+d}
```

Фигурные скобки говорят \TeX 'у, что аргументом #1 является $a+b$, а аргументом #2 является $c+d$, но \TeX тогда оттранслирует $\sqrt{a+b}\sqrt{c+d}$ в

$$\sqrt{a+b}\sqrt{c+d}$$

и мы получим $\sqrt{a+b} + \sqrt{c+d}$. Правильным будет определение

$$\define\sqrt{#1}\sqrt{#2}\sqrt{#1}+\sqrt{#2}$$

Общее правило таково: чтобы избежать таких неприятностей в этом определении, сами аргументы #1, #2, ..., должны быть заключены в фигурные скобки, если только у вас нет особых причин их не использовать.

На самом деле, даже если вы определите \sqrt{ab} таким образом, вы все равно не получите то, что хотите, потому что \sqrt{ab} даст $\sqrt{a}\sqrt{b}$ вместо $\sqrt{a+b}$. См. распорки в гл. 19.

18.20.

$$\define\vector{#1}\dots{#1}_{#2}$$

18.21. Ваш текстовый редактор, безусловно, хитроумен, но нет никакого простого способа заменить все эти $R_{i\{j\}}^k$ на $R^i_{\{j\}^k}$, а $R_{\alpha\{j\}}^{\beta}$ на $R^{\alpha}_{\{j\}^{\beta}}$, и т. д., и т. д., и т. д. К счастью, вы дали определение

$$\define\R{#1}\{#2\}^{\#3}\{#4\}$$

и повсюду использовали \R . Так что вам всего-то и нужно, что заменить это определение на

$$\define\R{#1}\{#2\}^{\#3}\{#4\}$$

и \TeX позаботится обо всем остальном сам.

18.22. В гл. 9 мы рекомендовали набор x_i^2 , с нижним индексом, набранным до верхнего, потому что это чрезвычайно просто меняется при помощи текстового редактора на x_i^2 . Но еще останется масса работы, так что если вы заранее знаете, что потребуется немало изменений, можно прибегнуть к

$$\define\sb{#1}^{\#2}$$

и затем набирать $x\sb{i}^2$, $y\sb{j}^3$, ..., В работе того автора, который предпочитает x_i^2 , y_j^3 , ..., вы просто замените это на

$$\define\sb{#1}^{\#2}$$

Таким образом, обдумав все заранее, можно при помощи \define заставить \TeX работать в качестве текстового редактора.

18.23. Наберите просто

```
\define\vector#1#2{(#2_1,\dots,#2_{#1})}
```

Хотя после `\define\vector` элементы #1 и #2 должны появляться именно в таком порядке, в самом определении они могут располагаться в любом порядке.

18.24.

```
\define\vector#1,#2.{{#1_1,\dots,#1_{#2}}}
```

с запятой после #1 и точкой после #2.

18.25. Для первых двух выражений можно набрать `$$\powers3,4.$` и `$$\powers 10.03, 1,034.$` (пробел между числами во втором случае необязателен, но это облегчает чтение). В третьем случае нельзя набирать `$$\powers 1,034, 10.03.$`, потому что `TeX` может подумать, что первым аргументом является 1 (все, что стоит до первой запятой), а вторым является 034,10 (все, что находится между первой запятой и первой точкой), так что у вас получится “ $2^1 + 3^{034,10}03$ ”. Вам следовало набрать `“$$\powers {1,034},{10.03}.$”`. Теперь фигурные скобки “прячут” запятую и точку — `TeX`’у не придет в голову, что первый аргумент {1, или что второй аргумент 034},{10, поскольку фигурные скобки при каждом аргументе всегда должны быть сбалансированными.

18.26.

```
The \deRham/ cohomology ring  $H^*(G)$ 
had, of course, already essentially been computed
by \Cartan/.
```

Теперь пробел после `\deRham/` превращается в пробел при печати, а пробел после `\Cartan/` нам не нужен. (Вы можете также набирать `\deRham /` и `\Cartan /`, так как пробелы после управляющих слов `\deRham` and `\Cartan` все еще игнорируются, но вы вряд ли захотите так усложнять свою жизнь.)

18.27. Это вполне приемлемое определение, но вы не определили нового управляющего слова ‘`\AmS-TeX`’, так как в управляющем слове не может стоять буква — как часть его имени. Вместо этого вы определили управляющее слово `\AmS`, и сообщили `TeX`’у, что за ним всегда будет идти последовательность `-TeX`. Если вы набираете `\AmS-TeX`, то `TeX` даст вам фирменный знак `AmS-TeX`, и можно также набрать `\AmS -TeX`, так как пробел после управляющего слова `\AmS`, как обычно, игнорируется. С другой стороны, пробелы после `-TeX` не игнорируются, так что это определение обладает тем же преимуществом, что и `\deRham/` и `\Cartan/` — вам не приходится беспокоиться относительно `_`, чтобы получить пробел после `\AmS-TeX`.

18.28. Первое определение полностью эквивалентно

```
\define\ab{\alpha^2+\beta^2}
```

— пробелы после управляющих слов `\define` и `\ab` всегда игнорируются.

Но после управляющего символа `\1` пробелы не игнорируются, так что второе определение говорит Т_ЭX'у, что за `\1` всегда должен следовать пробел. Если бы вы набрали

```
\1+\gamma^2
```

то получили бы сообщение об ошибке, что использование `\1` не совпадает с определением:

```
! Use of \1 doesn't match its definition.
```

(В вашем определении после `\1` стоит два пробела, но это не существенно, поскольку Т_ЭX всегда воспринимает последовательность пробелов как один пробел.)

18.29. Если за этим случайно окажется какая-нибудь группа

```
\define\alpha . . . { _ _ _ }
```

Т_ЭX предположит, что

- (1) вы пытались определить управляющую последовательность `\a`,
- (2) за `\a` всегда должно идти `\alpha . . .`,
- (3) `\a\alpha . . .` должно меняться на `_ _ _`.

Если такой группы не окажется, Т_ЭX пожалуется, что он добрался до конца файла, так и оказавшись не в состоянии завершить определение.

19.1.

```
$$
\frac {b^{p+1}-a^{p+1}}{\underbrace{1+\dots+1}_{\text{\$p\$ times}}} =
\frac {b^{p+1}-a^{p+1}}{p+1}.
$$
```

19.2. Вот один из вариантов:

```
$$
\underbrace{f\biggl(\frac{in}{biggr})+\dots+
f\biggl(\frac{in}{biggr})}_{\text{\$n\$ times}}=
f\biggl(\underbrace{\frac{in}{biggr}+\dots+
\frac{in}{biggr}}_{\text{\$n\$ times}}\biggr)
=f(1)=c.
$$
```

в котором используется особый размер `\bigg` для круглых скобок. (Тонкие шпации были добавлены после первой распечатки, вид которой нас не устроил.)

Если у вас аналогичная формула, и вы не уверены в ее высоте, т. е. нужно воспользоваться конструкцией `\left...\right`, тогда возникает более сложная

ситуация, поскольку вам не хочется заключать в круглые скобки целиком всю формулу

$$\underbrace{\frac{1}{n} + \dots + \frac{1}{n}}_{n \text{ times}}$$

Лучше всего было бы применить сначала к этой формуле `\botsmash`, а затем использовать `\vphantom{\frac{in}}`, благодаря чему Т_ЕX подумал бы, что высота формулы такая, как у $\frac{1}{n}$.

```
f\left(\,
  \botsmash{ \underbrace{\frac{in+\dots+\frac{in}}{n}}
    {\text{\$n\$ times}} } \vphantom{\frac{in}}{\,}, \right)
```

19.3.

```
$$
\overset{\frown}{\to{BC}}=\theta, \quad \quad \quad
\overline{AB}=\sqrt{2+2\cos\theta} \quad \quad \quad
\text{(by the law of cosines).}
$$
```

19.4.

```
$$
\align
G(z)&=e^{-\ln G(z)}=\exp\left(\sum_{k\ge 1}\frac{S_kz^k}{k}\right)=\prod_{k\ge 1}e^{-S_kz^k/k}\backslash\backslash
&=\left(1+S_1z+\frac{S_1^2z^2}{2!}+\dots\right)\left(1+\frac{S_2z^2}{2}+\frac{S_2^2z^4}{2^2\cdot 2!}+\dots\right)\dotsm\backslash\backslash
&=\sum_{m\ge 0}\left(\sum_{\substack{k_1,k_2,\dots,k_m\ge 0\backslash\backslash k_1+2k_2+\dots+mk_m=m}}\end{sub}
\frac{S_1^{k_1}}{1^{k_1}k_1!}\backslash\backslash
\frac{S_2^{k_2}}{2^{k_2}k_2!}\dotsm
\frac{S_m^{k_m}}{m^{k_m}k_m!}\right)z^m
\endalign
$$
```

(Эта формула взята из *The Art of Computer Programming*, Vol. 1, Дональда Кнута, и использована в качестве примера в руководстве для пользователей по математическому набору с помощью программы TROFF, которое написано Brian W. Kernighan и Lorinda L. Cherry.)

19.5. Ее можно набрать так:

```
$$
\alignat 2
m'&=m_1+2n_1&&=3m+4n, \backslash\backslash
n'&=m_1+\hphantom{2n_1}&&=2m+3n.
\endalignat
$$
```

Третий столбец предварен знаками `&&`, так что выравнивание делается по левому краю.

19.6.

```


$$\begin{aligned} f(x) &= \sum_{n=1}^{\infty} a_n x^{n-1} \\ & \quad \&\&= 1+x+2x^2+3x^3+\dotsb, \\ xf(x) &= \sum_{n=1}^{\infty} a_n x^n \\ & \quad \&\&= \phantom{1+}x+\phantom{2}x^2 \\ & \quad \quad +2x^3+\dotsb, \\ x^2f(x) &= \sum_{n=1}^{\infty} a_n x^{n+1} \\ & \quad \&\&= \phantom{1+x}x^2+\phantom{2}x^3+\dotsb. \end{aligned}$$


```

Обратите внимание, что мы набирали `\hphantom{1+{}}` с `{}` в конце, чтобы обеспечить дополнительный пробел после знака `+`. Заметьте также, что `AMS-TEX` автоматически вставляет тонкие шпации между `\dotsb` и запятой или точкой, поскольку он считает, что вы включаете это выражение как часть формулы только потому, что набираете выключную формулу.

Приложение С. Библиография

Если вы, находясь в стиле `amsppt`, уже подошли к стадии набора библиографии, прежде всего наберите `\Refs`. Это приведет к заголовку `REFERENCES`, и затем можно приступать к набору отдельных библиографических названий. На самом деле конкретные слова в заголовке рассматриваются как “frill” (ср. с автоматическое форматирование, отмена в гл. 19), так что вы имеете право подставить свой заголовок, набрав

```
\Refs\nofrills{(Ваш заголовок)}
```

После того как последняя библиографическая ссылка набрана, библиографический раздел следует завершить командой `\endRefs`.

Типичная библиографическая ссылка оформляется так:

```
\ref \no 9 \by S. S. Chern \pages 947--955  
\paper Integral formulas for hypersurfaces in Euclidean  
space and their applications to uniqueness theorems  
\yr1959 \vol 8  
\jour J. Math. Mech.\endref
```

В стиле `amsppt` это даст

9. S. S. Chern, *Integral formulas for hypersurfaces in Euclidean space and their applications to uniqueness theorems*, J. Math. Mech. 8 (1959), 947–955.

Обратите внимание, что нет необходимости набирать отдельные элементы ссылки в том порядке, в котором это должно быть напечатано, равно как не нужно заботиться о шрифтах и пунктуации или о таких деталях, как скобки, в которые заключается год — все это берет на себя стилевой файл. Однако если есть такая необходимость, можно отменить знаки препинания, автоматически проставляемые стилевым файлом. Средство `\nofrills` используется либо непосредственно перед, либо сразу после любой из команд, встречающихся внутри `\ref... \endref`. Если `\nofrills` набрано перед командой, то знак пунктуации, который должен был бы предшествовать этой команде, будет удален, а если `\nofrills` набрано после команды, то будет отменен знак пунктуации, который автоматически проставляется после команды.

Заметьте также, что в ‘`Math.`’ и ‘`Mech.`’ не нужно использовать ‘`\.`’; после команды `\Refs` все пробелы считаются межсловными, даже после точек. (Если нужен правильный пробел после завершенной фразы, например, если название состоит из двух предложений, вы можете воспользоваться конструкцией ‘`\@.`’, которая обычно используется для получения точки в конце предложения, заканчивающегося одной прописной буквой.)

Для ссылок, которые помечаются иначе, чем просто число `\no`, можно воспользоваться `\key`: например `\key C1` даст в начале ссылки ‘`[C1]`’. Обратите внимание, что квадратные скобки не набираются — это автоматически проставляет стилевой файл. И `\key` и `\no` могут быть опущены.

В стиле `amsrpt` по умолчанию установлена втяжка на двузначное число. Она может быть уменьшена или увеличена в зависимости от наиболее широкого числа или ключа, используемого в качестве пометки библиографической ссылки. Например,

```
\widestnumber\{no{999}
\widestnumber\{key{GHMaR}
```

увеличит втяжку до ширины соответственно трехзначного числа или ключа [GHMaR]. Разумеется, оформительские детали, зависящие от того или иного журнального стиля (как то: точка после номера, квадратные скобки, в которые заключен ключ, плюс обычные пробелы), автоматически принимаются во внимание. Можно также дать команду `\widestnumber\{no{9}`, которая уменьшит втяжку с ширины двузначной цифры до однозначной, если в вашей библиографии менее десяти работ.

Нет особо оговоренного обязательного объема информации, который вы обязаны включать в каждую конструкцию `\ref... \endref` — стилевой файл должен поступать разумно с той информацией, которую вы задаете. Например, если вы не заполнили `\vol`, то он просто не будет напечатан, но если вы оставили пустым `\jour`, то `\vol` и `\yr` будут проигнорированны, даже если вы их заполнили. Что именно будет напечатано, также зависит от контекста, в котором это используется. Например, если вы воспользовались `\paper`, то в `\page` или `\pages` будут напечатаны только номера страниц. Но если вы применили `\book` или `\inbook`, то в `\page` или `\pages` будет еще напечатано “р.” или “pp.” перед номерами страниц.

Кроме перечисленных, внутри `\ref... \endref` могут быть `\toappear`, который дает текст ‘(to appear)’, и `issue`, который может быть нужен для особых выпусков, автоматически дающий перед выпуском номер “no.” Если данная ссылка представляет собой перевод, можно использовать `\lang` для указания языка оригинала. Эта информация будет напечатана после завершающего знака пунктуации и автоматически взята в круглые скобки. Информация о переводе вводится посредством `\transl`, который должен быть набран непосредственно перед относящимся к нему.

Если имеется более одной работы одного и того же автора, то первая ссылка набирается, как обычно, посредством `\by`, а в последующих все это заменяется на `\bysame`. В стиле `amsrpt` все ссылки с `\bysame` вместо фамилии автора снабжаются горизонтальной чертой длины 3 em как это показано в следующем примере.

```
\ref \no7
\by F. T. Farrell and L. E. Jones
\paper Rigidity and other topological aspects of
compact non-positively curved manifolds
\jour Bull. Amer. Math. Soc.
\toappear \endref
\ref \no8 \by I. A. Sokolov
\jour Vests\B\i\ Akad. Navuk BSSR Ser. F\B\i z.-Mat.
Navuk
```

```

\vol 1969 \issue 5 \pages 64--71
\lang Russian \endref
\ref
\no9 \by K. M. Rasulov
\jour Dokl. Akad. Nauk SSSR
\vol 252 \yr 1980 \pages 1059--1063
\transl\nofrills English transl. in \jour Soviet Math.
Dokl. \vol 21 \yr 1980 \endref
\ref
\no10 \bysame \jour Dokl. Akad. Nauk SSSR
\vol 270 \yr 1983 \pages 1061--1065
\transl\nofrills English transl. in \jour Soviet Math.
Dokl. \vol 27 \yr 1983 \endref

```

Это даст на печати следующее:

7. F. T. Farrell and L. E. Jones, *Rigidity and other topological aspects of compact non-positively curved manifolds*, Bull. Amer. Math. Soc. (to appear).
8. I. A. Sokolov, *Vestsi Akad. Navuk BSSR Ser. Fiz.-Mat. Navuk* 1969 no. 5, 64-71. (Russian)
9. K. M. Rasulov, *Dokl. Akad. Nauk SSSR* 252 (1980), 1059-1063; English transl. in *Soviet Math. Dokl.* 21 (1980).
10. _____, *Dokl. Akad. Nauk SSSR* 270 (1983), 1061-1065; English transl. in *Soviet Math. Dokl.* 27 (1983).

Для ссылок на книги имеются такие средства: `\book`, `\ed` или `\eds`, `\publ` и `\publaddr`. По аналогии с `\page` и `\pages` введены два варианта задания фамилий редакторов: `\ed` после фамилии редактора автоматически поставит "ed.", а `\eds` поставит "eds." за фамилиями редакторов. Оба автоматически заключают фамилии редакторов в круглые скобки. Например, в стиле `amsprt` ввод

```

\ref\no 20\by P. Chadwick
\book Thermoelasticity, the dynamical theory, {\rm Progress
in Solid Mechanics, Vol. I}
\eds J. N. Sneddon and R. Hill
\publ North-Holland Publishing Company
\yr 1960 \page 265 \endref

```

даст

20. P. Chadwick, *Thermoelasticity, the dynamical theory*, Progress in Solid Mechanics, Vol. I (J. N. Sneddon and R. Hill, eds.), North-Holland Publishing Company, 1960, p. 265.

Обратите внимание, что в данном случае `\page` дает 'р.' перед номером страницы.

Иногда статья помещается в книге, а не в журнале. В этом случае для названия статьи используется `\rarep`, а для названия книги — `\inbook` (Эта книга).

Ниже приведен пример статьи из книги. Ввод

```
\ref\no 9\by S. Kripke\paper\nofrills Semantical
analysis of intuitionistic logic \rm I;
\inbook in Formal Systems and Recursive Functions
\eds J. Crossely and M. A. E. Dummett
\publaddr North-Holland\yr 1965\pages 92--130\endref
```

даст

9. S. Kripke, *Semantical analysis of intuitionistic logic I*; in *Formal Systems and Recursive Functions* (J. Crossely and M. A. E. Dummett, eds.), North-Holland, 1965, pp. 92-130.

Иногда в библиографических стилях требуется, чтобы название книги или статьи было заключено в кавычки, а не выделено курсивом. В стиле `amsprt` вы можете воспользоваться после `\Refs` управляющими последовательностями `\bookinquotes` и (или) `\paperinquotes` и получите название вашей книги и (или) статьи в кавычках, набранное обычным шрифтом.

Для двух ссылок, объединенных в одну, можно воспользоваться конструкцией `\moreref`. Перед текстом, выдаваемым `\moreref`, будет автоматически проставляться точка с запятой. Например, в стиле `amsprt` ввод

```
\ref\no4 \by L. Auslander \paper On the
Euler characteristic of
compact locally affine spaces \jour Comment.
Math. Helv. \vol 35 \yr 1961
\pages 25--27 \moreref \paper II\jour Bull. Amer. Math. Soc.
\vol 67\yr 1961\pages 405--406 \endref
```

даст

4. L. Auslander, *On the Euler characteristic of compact locally affine spaces*, *Comment. Math. Helv.* **35** (1961), 25-27; *II*, *Bull. Amer. Math. Soc.* **67** (1961), 405-406.

Помимо всех перечисленных управляющих последовательностей для указания вспомогательной информации существует средство `\paperinfo`. Эта информация будет напечатана сразу после названия статьи (для указания, например, что статья представляет собой препринт на mimeографе). Точно так же `\bookinfo` можно использовать для указания дополнительной информации к `\book`. Средство `\finalinfo` поместит вспомогательную информацию в самом конце. После `\paperinfo`, `\bookinfo` и `\finalinfo` знаки препинания проставляются автоматически, но посредством `\nofrills` сразу после этих команд заданные знаки пунктуации можно отменить, например, `\finalinfo\nofrills`. Следующий

ВВОД

```

\ref \key C \by H. Cartan
\paper Operations dans les construction acycliques
\inbook Seminaire H. Cartan 1954-55
\bookinfo Expos{\ ' e} 6 \publ ENS \publaddr Paris
\finalinfo reprinted by W. A. Benjamin, New York (1967)
\endref
\ref\key D
\by L. N. Deshpande
\book Approximate solution of eigenvalue problems
\bookinfo Ph.D. Thesis \publ Indian Institute of Technology
\publaddr Bombay \yr1987
\endref
\ref\key K \by G. Knieper \paper On the second
derivative of metric entropy at locally symmetric spaces
\paperinfo in preparation \endref

```

ДАСТ

- [C] H. Cartan, *Operations dans les construction acycliques*, Seminaire H. Cartan 1954-55, Exposé 6, ENS, Paris, reprinted by W. A. Benjamin, New York (1967).
- [D] L. N. Deshpande, *Approximate solution of eigenvalue problems*, Ph.D. Thesis, Indian Institute of Technology, Bombay, 1987.
- [K] G. Knieper, *On the second derivative of metric entropy at locally symmetric spaces*, in preparation.

Приложение D. Сравнение с 'plain' TeX'ом

TeX обычно выступает с обилием управляющих последовательностей, предоставляемых стилем 'plain', и AMS-TeX основывается на этих конструкциях. В целом, AMS-TeX состоит из plain, снабженного множеством дополнительных управляющих последовательностей, так что большинство конструкций из plain'a, если вы с ними знакомы, могут быть доступны непосредственно из AMS-TeX'a. Например, в AMS-TeX'e можно также использовать `\settabs` и `\+`, которые описываются в *The TeXbook*, несмотря на то, что нигде в этом учебнике `\settabs` не упоминается. Точно так же можно использовать конструкции `\narrower` и `\item`, приводимые в *The TeXbook* (только учтите, что функционально `\item` абсолютно отличны от тех, что встречаются в конструкции AMS-TeX'a `\roster`).

Однако некоторые средства plain не могут быть использованы с AMS-TeX'ом, и, напротив, многие управляющие последовательности AMS-TeX'a перекрывают или даже конфликтуют с управляющими последовательностями с тем же самым именем в plain'e. Если вы намереваетесь использовать plain и AMS-TeX вместе, или переключаться с одного на другой, вам может оказаться полезной следующая информация.

Прежде всего, plain использует `\.` для получения точки над буквой, тогда как AMS-TeX применяет `\.` для точки после аббревиатуры (которая не стоит в конце предложения), а для точки над буквой ставит `\D`, продолжая пользоваться plain-средством `\d`, чтобы поставить точку под буквой. Несмотря на полное несоответствие plain'у, такое использование `\.` настолько удобно, что именно его и следует придерживаться (во всяком случае `\.` для точки над буквой не представляется логичным!). Поскольку plain использует `\b` для черты под буквой, кажется естественным использовать по аналогии `\B` для черты над буквой. В plain'e черта над буквой обозначается `\=`. В AMS-TeX'e эта управляющая последовательность не определена, так что доступна для использования в иных целях. (Если вы уже нашли, для чего применить `\=`, то можете дать для нее `\define` сами; в `\redefine` нет никакой необходимости.)

Столь явные конфликты с plain'ом бывают редко, по немало случаев, когда AMS-TeX использует plain-конструкции в расширенном варианте. Например, в plain'e следует заботиться о том, чтобы ни с одной стороны не остался пробел, тогда как AMS-TeX их просто игнорирует. В plain'e использование управляющих последовательностей `\{` и `\}`, а также `\,` и `\!` допускается только в математическом режиме, а в тексте вместо `\,` и `\!` применяются соответственно `\thinspace` и `\negthinspace`. Но AMS-TeX все шесть этих последовательностей использует как в математическом, так и в текстовом режиме.

Аналогично, AMS-TeX допускает использование как в математическом режиме, так и вне его, таких средств как `\medspace`, `\negmedspace`, `\thickspace` и `\negthickspace`, и в качестве синонима для `\medspace` применяет также `\;`. В plain'e толстая шпация `\;` может быть использована только в математическом режиме. С другой стороны, у plain'a есть краткое название `\>` для AMS-TeX'ового `\medspace` (хотя `\>` можно применять только в математическом режиме), тогда как AMS-TeX фактически делает `\>` неопределенным, оставляя его доступным для будущих разработок или для ваших личных целей. Потеря

краткого названия для шпации “среднего размера”, которая проставляется вокруг бинарных операторов, не должна вас сильно огорчать, поскольку обычно Т_ЕХ сам вставляет такие шпации — явное задание “средней” шпации бывает нужно крайне редко (во всей книге *The Т_ЕХbook* это ни разу не понадобилось).

Есть одно незначительное обстоятельство, которое вы можете никогда не заметить: акцент `\t` работает в АМ_S-Т_ЕХ'е и plain'е несколько по-разному. В АМ_S-Т_ЕХ'е у управляющей последовательности `\t` два аргумента, и перед вторым вы можете даже оставить пробел, поскольку Т_ЕХ всегда игнорирует пробелы, когда ищет аргументы управляющих последовательностей, но в plain'е никаких пробелов быть не должно.

И в АМ_S-Т_ЕХ'е, и в plain'е неукоснительно требуется использовать иные названия для акцентов в математическом режиме (что позволяет Т_ЕХ'у обнаруживать пропущенные знаки \$). Но plain позволяет использовать инструкцию замены шрифта типа `\bf` и т. п. в математическом режиме, тогда как АМ_S-Т_ЕХ ведет себя в этом вопросе весьма последовательно и требует также, чтобы вы неукоснительно придерживались в математическом режиме иных названий типа `\bold` и т. п.; более того, `\bold` и т. п. суть управляющие последовательности с одним аргументом, а не указание изменить шрифт. Для изменения шрифта в математических формулах можно использовать только такую конструкцию:

```
$$ \text{\bf...} $$
```

Здесь, разумеется, все пробелы, которые вы оставите в . . . , появятся и в распечатке: этим качеством не наделено средство замены шрифта из plain'a.

Далее, у АМ_S-Т_ЕХ'a имеется `\cal`, управляющая последовательность с одним аргументом, для получения “рукописных” букв, тогда как plain достигает этого посредством инструкции замены шрифта `\cal`. В АМ_S-Т_ЕХ'е управляющая последовательность `\cal` становится неопределенной. Ту же судьбу ожидает замена шрифта на “математический курсив” — `\mit` из plain'a, который используется для того, чтобы получить другое написание прописных греческих букв; в АМ_S-Т_ЕХ'е для них просто есть иное название `\varGamma`, и т. п. И наконец, управляющая последовательность `\oldstyle`, которую plain использует для изменения написания цифр: 1, 2, 3, . . . , вероятно станет неопределенной, ее роль в АМ_S-Т_ЕХ'е присвоила себе конструкция `\oldnos`.

Мы уже упоминали, что вам следует быть внимательными (см. сс. 69 и 175) в ситуациях, когда за `^` или `_` стоит управляющая последовательность, а не группа символов; эту последовательность следует взять в фигурные скобки, если только в ее определении эти дополнительные фигурные скобки не были заранее введены. В plain Т_ЕХ'е аналогичная опасность вас подстерегает с управляющими последовательностями после `\sqrt`, `\underline` и `\overline`, но АМ_S-Т_ЕХ предусмотрительно их сторонится. Следовательно, конструкции, в которые входят `\sqrt`, `\underline` и `\overline`, отлично работающие с АМ_S-Т_ЕХ'ом, могут выдать головоломную распечатку или сообщение об ошибке в plain Т_ЕХ'е.

Еще одно разногласие возникает между plain Т_ЕХ'ом и АМ_S-Т_ЕХ'ом, когда дело касается верхних индексов. Кроме обычного приема, когда индекс ' ' в математическом режиме транслируется в `^{\prime\prime}`, в plain Т_ЕХ'е можно

даже набирать следующее:

$$f''^2$$

что дает f''^2 , вместо того чтобы набирать

$$f^{\prime\prime\prime 2}$$
 или $f^{\prime\prime\prime 2}$

Но А_MS-Т_EX не должен пользоваться таким приемом, поскольку нет четко очерченных правил для получения f''^2 (что тоже встречается, хотя и значительно реже).

Имеется еще ряд незначительных отличий в представлении отдельных математических конструкций в plain'е и в А_MS-Т_EX'е. Например, `\ldots` и `\cdots` в А_MS-Т_EX'е автоматически оставляют дополнительную тонкую шпацию в конце формулы или после правого ограничителя, подобно общей опции `\dots` из А_MS-Т_EX. Но такого автоматического соглашения в plain'е нет. Точно так же `\rmod` в А_MS-Т_EX'е применяет разные пробелы в тексте и выключных формулах. Кроме того, большие операторы типа `\sum`, названия операторов типа `\max`, и ряд других конструкций в А_MS-Т_EX'е и в plain'е, в сущности, определяются совершенно по-разному, так что в разных форматах они могут делать разные вещи; но такие различия не должны привести к конфликтным ситуациям.

Противоречия между А_MS-Т_EX'ом и plain'ом наиболее ярко выражены в управляющих последовательностях `\matrix` и `\cases`. Правильный "синтаксис" для этих команд в этих двух ситуациях совершенно отличается один от другого. Другие управляющие последовательности А_MS-Т_EX'а, вроде `\align`, естественным образом поддерживают синтаксис управляющих последовательностей А_MS-Т_EX'а `\matrix` и `\cases`, тогда как соответствующие управляющие последовательности plain'а типа `\eqalign` и `\displaylines` в основном следуют синтаксису plain'а. Но никаких дальнейших конфликтов не возникает, поскольку ни одна из подобных управляющих последовательностей не имеет одиноких названий в plain'е и в А_MS-Т_EX'е.

И, наконец, последнее противоречие несколько иного плана между plain'ом и А_MS-Т_EX'ом. Управляющие последовательности `\proclaim` и `\footnote` используются и в plain'е, и в А_MS-Т_EX'е, и соответствующий им синтаксис в обоих случаях различный. Например, в plain'е управляющая последовательность `\footnote` имеет два аргумента, первый представляет собой знак сноски, а второй — ее текст, тогда как в А_MS-Т_EX'е знак сноски определяется стилем. Фактически, `\footnote` и `\proclaim` определяются не самим А_MS-Т_EX'ом, а конкретным стилевым файлом, который он использует, так что все стилевые файлы, написанные для А_MS-Т_EX'а будут использовать правильный синтаксис. Если А_MS-Т_EX используется без указания конкретного стиля, т. е. без всякого `\documentstyle{...}`, то естественно иметь в таком случае стиль, который обычно работает с plain'ом. Здесь, однако, А_MS-Т_EX, столкнувшись с `\proclaim` и `\footnote`, заявит, что они *неопределены* и вам придется вместо них использовать `\plainproclaim` и `\plainfootnote` с их совершенно иным синтаксисом.

Приложение Е. Отсутствующие клавиши

Если на вашей клавиатуре отсутствуют некоторые клавиши, или их почему-либо неудобно нажимать, можете воспользоваться следующими управляющими последовательностями:

<i>Использование</i>	<i>Для</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>	

Если для какой-то из клавиш, например для ', вы используете соответствующую управляющую последовательность, то вам потребуются и другие команды для комбинаций с \':

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

Обратите внимание, что краткие названия `\acute`, ..., `\tilde` уже заняты для математического режима.

Управляющие последовательности

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

можно принять для получения фигурных скобок. На иных национальных клавиатурах, в которых используется расширенная таблица кодов ASCII, сами фигурные скобки могут быть заменены буквами с акцентами. В подобных ситуациях для группирования можно воспользоваться клавишами < и >, а вместо них ввести символы вроде `\less` и `\greater`.

Приложение F. Специальные знаки

Математические знаки + - = < > { / () [] и * имеются на клавиатуре. Однако вместо |, [,], \{, \} и * можно использовать \vert, \lbrack, \rbrack, \lbrace, \rbrace и \ast.

• Строчные греческие буквы

α	<code>\alpha</code>	β	<code>\beta</code>	γ	<code>\gamma</code>	δ	<code>\delta</code>
ϵ	<code>\epsilon</code>	ε	<code>\varepsilon</code>	ζ	<code>\zeta</code>	η	<code>\eta</code>
θ	<code>\theta</code>	ϑ	<code>\vartheta</code>	ι	<code>\iota</code>	κ	<code>\kappa</code>
λ	<code>\lambda</code>	μ	<code>\mu</code>	ν	<code>\nu</code>	ξ	<code>\xi</code>
π	<code>\pi</code>	ϖ	<code>\varpi</code>	ρ	<code>\rho</code>	ϱ	<code>\varrho</code>
σ	<code>\sigma</code>	ς	<code>\varsigma</code>	τ	<code>\tau</code>	υ	<code>\upsilon</code>
ϕ	<code>\phi</code>	φ	<code>\varphi</code>	χ	<code>\chi</code>	ψ	<code>\psi</code>
ω	<code>\omega</code>						

• Прописные греческие буквы

Прописные греческие буквы имеются в трех начертаниях — в прямом, наклонном и полужирном:

Γ	<code>\Gamma</code>	Δ	<code>\Delta</code>	Θ	<code>\Theta</code>	Λ	<code>\Lambda</code>
Ξ	<code>\Xi</code>	Π	<code>\Pi</code>	Σ	<code>\Sigma</code>	Υ	<code>\Upsilon</code>
Φ	<code>\Phi</code>	Ψ	<code>\Psi</code>	Ω	<code>\Omega</code>		
Γ	<code>\varGamma</code>	Δ	<code>\varDelta</code>	Θ	<code>\varTheta</code>	Λ	<code>\varLambda</code>
Ξ	<code>\varXi</code>	Π	<code>\varPi</code>	Σ	<code>\varSigma</code>	Υ	<code>\varUpsilon</code>
Φ	<code>\varPhi</code>	Ψ	<code>\varPsi</code>	Ω	<code>\varOmega</code>		
Γ	<code>\bold\Gamma</code>	Δ	<code>\bold\Delta</code>	Θ	<code>\bold\Theta</code>	Λ	<code>\bold\Lambda</code>
Ξ	<code>\bold\Xi</code>	Π	<code>\bold\Pi</code>	Σ	<code>\bold\Sigma</code>	Υ	<code>\bold\Upsilon</code>
Φ	<code>\bold\Phi</code>	Ψ	<code>\bold\Psi</code>	Ω	<code>\bold\Omega</code>		

• “Рукописные” прописные буквы

Прописные буквы A, \dots, Z получаются посредством `\Cal A, \dots, \Cal Z`.

• Бинарные операторы

\pm	<code>\pm</code>	\cap	<code>\cap</code>	\vee	<code>\vee, \lor</code>
\mp	<code>\mp</code>	\cup	<code>\cup</code>	\wedge	<code>\wedge, \land</code>
\setminus	<code>\setminus</code>	\uplus	<code>\uplus</code>	\oplus	<code>\oplus</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>
\ast	<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>\&</code>

Некоторые математики вместо \wedge используют оператор $\&$, который получается посредством `\and`. Обратите внимание, что когда \dagger и \ddagger выполняют роль бинарных операторов, используются `\dagger` и `\ddagger`.

• Бинарные отношения

\leq	<code>\leq</code> , <code>\le</code>	\geq	<code>\geq</code> , <code>\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 , \owns	<code>\ni</code> , <code>\owns</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</code> , <code>\ne</code>	\notin	<code>\notin</code>		

`\mid` и `\parallel` суть те же самые литеры, что и `|` и `\|`, но трактуемые как бинарные отношения, поэтому здесь с обеих сторон заложены дополнительные пробелы.

Можно получить отрицание многих из этих отношений, поместив перед ними `\not`. Например, `\not\subset` дает $\not\subset$. А вот `\ne` и `\neq` просто аббревиатуры для `\not=`. Но такое расположение далеко не идеально и, в частности, следует для \notin всегда использовать `\notin`, а не `\not\in`.

• Мозаика одиночных символов

\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>	∇	<code>\nabla</code>	\neg	<code>\neg</code> , <code>\not</code>
\jmath	<code>\jmath</code>	\surd	<code>\surd</code>	\flat	<code>\flat</code>
ℓ	<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>	\parallel	<code>\ </code> , <code>\Vert</code>	\clubsuit	<code>\clubsuit</code>
\Im	<code>\Im</code>	\angle	<code>\angle</code>	\diamond	<code>\diamondsuit</code>
∂	<code>\partial</code>	\triangle	<code>\triangle</code>	\heartsuit	<code>\heartsuit</code>
∞	<code>\infty</code>	\backslash	<code>\backslash</code>	\spadesuit	<code>\spadesuit</code>
\int	<code>\smallint</code>	\dagger	<code>\dagger</code>	\ddagger	<code>\ddagger</code>
\P	<code>\P</code>	\S	<code>\S</code>		

Символы `\imath` и `\jmath` используются для получения варианта с акцентом: `\hat{\imath}` дает \hat{i} . Символ `\backslash` (а не `\setminus`) следует использовать для двойных множеств ($G \setminus H$) и для указания, что p делится на n ($p \setminus n$). Обычно `\prime` используется для верхних и нижних индексов. Символ `\angle` построен из кусочков и не делается меньше для индексов. Довольно редко используются `\smallint` и `\surd`. Такие значки, как `\dagger`, `\ddagger`, `\P` и `\S` находят

применение в качестве особых эффектов; в индексах правильно меняют свой размер.

• Стрелки

\leftarrow	<code>\leftarrow, \gets</code>	\longleftarrow	<code>\longleftarrow</code>
\Lleftarrow	<code>\Lleftarrow</code>	\Longleftarrow	<code>\Longleftarrow</code>
\rightarrow	<code>\rightarrow, \to</code>	\longrightarrow	<code>\longrightarrow</code>
\Rrightarrow	<code>\Rrightarrow</code>	\Longrightarrow	<code>\Longrightarrow</code>
\leftrightarrow	<code>\leftrightarrow</code>	\longleftrightarrow	<code>\longleftrightarrow</code>
\Leftrightarrow	<code>\Leftrightarrow</code>	\Longleftrightarrow	<code>\Longleftrightarrow</code>
\uparrow	<code>\uparrow</code>	\Uparrow	<code>\Uparrow</code>
\downarrow	<code>\downarrow</code>	\Downarrow	<code>\Downarrow</code>
\updownarrow	<code>\updownarrow</code>	\Updownarrow	<code>\Updownarrow</code>
\nearrow	<code>\nearrow</code>	\searrow	<code>\searrow</code>
\swarrow	<code>\swarrow</code>	\nwarrow	<code>\nwarrow</code>
\mapsto	<code>\mapsto</code>	\longmapsto	<code>\longmapsto</code>
\hookrightarrow	<code>\hookrightarrow</code>	\hookrightarrow	<code>\hookrightarrow</code>
\leftharpoonup	<code>\leftharpoonup</code>	\leftharpoonup	<code>\leftharpoonup</code>
\rightharpoonup	<code>\rightharpoonup</code>	\leftharpoonup	<code>\leftharpoonup</code>
\rightleftharpoons	<code>\rightleftharpoons</code>	\rightharpoonup	<code>\rightharpoonup</code>

Вертикальные стрелки представляют собой “ограничители”, такие же, как перечисленные ниже; они меняют размер, когда используются после `\left` и `\right`. Команда `\iff` дает стрелку наподобие `\Longleftrightarrow`, с тем различием, что здесь пробелы с обеих сторон чуть побольше. $\mathcal{A}\mathcal{M}\mathcal{S}\text{-}\mathcal{T}\mathcal{E}\mathcal{X}$ также предоставляет `\implies` и `\impliedby`, которые в точности совпадают соответственно с `\Longrightarrow` и `\Longleftarrow`, но опять же с большими пробелами по обе стороны.

• Большие операторы типа Σ

Все большие операторы бывают двух размеров, причем больший соответствует случаю `\dsize`.

Σ	<code>\sum</code>	\cap	<code>\bigcap</code>	\odot	<code>\bigodot</code>
\prod	<code>\prod</code>	\cup	<code>\bigcup</code>	\otimes	<code>\bigotimes</code>
\coprod	<code>\coprod</code>	\sqcup	<code>\bigsqcup</code>	\oplus	<code>\bigoplus</code>
\vee	<code>\bigvee</code>	\uplus	<code>\biguplus</code>	\wedge	<code>\bigwedge</code>

• Большие операторы типа \int

\int	<code>\int</code>	\oint	<code>\oint</code>
\iint	<code>\iint</code>	\iiint	<code>\iiint</code>
\iiint	<code>\iiint</code>	$\int \dots \int$	<code>\int \dots \int</code>
\iiint	<code>\iiint</code>	$\int \dots \int$	<code>\int \dots \int</code>

• **Ограничители**

Следующие символы относятся к “ограничителям”.

([\lbrack	{	\{, \lbrace	
)]	\rbrack	}	\}, \rbrace	
[\lfloor	[\lceil	<	\langle
]	\rfloor]	\rceil	>	\rangle
	\vert		\ , \Vert	/	/
\	\backslash				

Все направленные вверх и вниз стрелки также могут выступать в качестве ограничителей. Более того, < и > можно использовать вместо \langle и \rangle после \left и \right. И, разумеется, для “пустого” ограничителя после \left и \right можно использовать также ‘.’.

Приложение G. Дополнительные шрифты

Многие годы математики принуждают полиграфистов создавать груды спецзнаков, еще невообразимее представленных в приложении F. Американское математическое общество разработало семейства новых шрифтов, расширив семейство стандартных шрифтов Computer Modern набором спецзнаков.

Были созданы два новых семейства: `msam` (содержащее шрифты размером от 5 до 10 пунктов — `msam5`, ..., `msxm10`) и `msbm` (содержащее шрифты размером от 5 до 10 пунктов — `msbm5`, ..., `msbm10`). Загрузка таких шрифтов в `AMS-TeX`'е осуществляется в случае, если они нужны, посредством команд `\loadmsam` и `\loadmsbm`. В стиле `\amspp` эта загрузка выполняется автоматически. (Если вы пользуетесь стилем `\amspp` или применяете такую загрузку, когда сами шрифты отсутствуют, то можете получить странное сообщение о том, что шрифтов нет.)

Но если все необходимые шрифты имеются, вы можете воспользоваться всем многообразием символов, содержащихся в этих шрифтах. Все они вместе с соответствующими управляющими последовательностями перечислены ниже. Если вы обращаетесь к `AMS-TeX`'у не в стиле `\amspp`, то необходимо предпринять соответствующие действия, чтобы эти названия стали доступными; если же вы работаете в стиле `\amspp`, то эти имена уже определены.

Если, работая вне стиля `\amspp`, вы пожелаете воспользоваться лишь отдельными из этих символов, то у вас есть две возможности. Можете набрать `\UseAMSsymbols`, чтобы сразу определить все эти новые имена. В результате к внутренней таблице `TeX`'а добавится 200 новых управляющих последовательностей. Так что если вы ограничены в пространстве, или вам нужна лишь небольшая часть новых символов, можете применить команду `\newsymbol`, как только такая необходимость возникнет.

• Буквы в качестве символов

Кроме спецзнаков, математики часто используют в качестве символов буквы особого начертания. Специально для Американского математического общества были созданы особые дополнительные буквенные шрифты, которые также могут использоваться в `AMS-TeX`'е. В семейство эйлеровых шрифтов включены три основных стиля; каждый стиль подразделяется на две насыщенности: “средний” и “полужирный”. Доступ к ним возможен посредством соответствующих команд, как показано ниже:

шрифт	команда	
<code>eufm</code>	<code>\loadeufm</code>	Готический средний (загружается с <code>\amspp</code>)
<code>eufb</code>	<code>\loadeufb</code>	Готический полужирный
<code>eusm</code>	<code>\loadeusm</code>	Рукописный средний
<code>eusb</code>	<code>\loadeusb</code>	Рукописный полужирный
<code>eurm</code>	<code>\loadeurm</code>	Курсивный средний
<code>eurb</code>	<code>\loadeurb</code>	Курсивный полужирный

• Спецзнаки и ажурные буквы

Некоторые символы из семейства `msam` можно специфицировать управляющими последовательностями, которые будут определяться, как только в файле появится команда `\loadmsam`.

Прежде всего, четыре символа, которые обычно используются вне математического режима:

✓	<code>\checkmark</code>	Ⓜ	<code>\circledR</code>
✠	<code>\maltese</code>	℥	<code>\yen</code>

Они, равно как и ¶, §, † и ‡, могут также использоваться и в математическом режиме и будут правильно менять свои размеры в индексах.

Затем имеются четыре символа типа “ограничителей” (хотя для них нет больших версий, получаемых посредством `\left` и `\right`), поэтому их следует использовать в математическом режиме:

┌	<code>\ulcorner</code>	┐	<code>\urcorner</code>
└	<code>\llcorner</code>	┘	<code>\lrcorner</code>

И, наконец, из символов этого семейства составлены две пунктирные стрелки:

`-->` `\dashrightarrow`, `\dasharrow` `←--` `\dashleftarrow`

Ажурные буквы

A, ..., Z

находятся в семействе `msbm`. Как только выполнена команда `\loadbmsbm`, можно в математическом режиме набирать `\Bbb A`, ..., `\Bbb Z`. Семейство `msbm` содержит расширенную версию `\widehat` и `\widetilde`, как описано в гл. 19 в разделе “Акценты в математическом режиме”.

• Команда `\newsymbol`

Все другие символы из семейства шрифтов `msam` и `msbm` именуется своими управляющими последовательностями, так что как только шрифты загружены, этими символами можно пользоваться. Этого можно достичь, загрузив их все разом; если набрать команду `\UseAMSsymbols`, символы будут загружаться в файл `AMSSYM.TEX`. Эта команда включена в стиль `\amsptt`, так что названия будут приписаны автоматически, и появятся более 200 новых управляющих последовательностей. Если вы ограничены в пространстве, или вам нужно лишь несколько новых символов, можете опустить команду `\UseAMSsymbols` и вместо этого назначить только те имена, которые вам нужны, используя `\newsymbol` для создания управляющей последовательности, которая и произведет этот символ. Эта управляющая последовательность может получить либо одно из перечисленных ниже “стандартных” имен, либо то, которое вы выберете для нее сами.

В приведенном ниже списке каждый символ представлен собственно символом, четырехлитерным кодом “ID” и “стандартным названием” символа. Например, символ $\not\leq$ представлен так:

$\not\leq$ 230A `\nleqslant`

Управляющую последовательность с тем же именем можно извлечь и из файла `AMSSYM.TEX`:

\newsymbol\neqslant 230A

Ту же самую команду может набрать и пользователь, не применяющий стиль `amsprt` и выбравший вариант, при котором не надо загружать все шрифты посредством `\UseAMSsymbol`, после чего команда `\neqslant` выдаст символ \neq (в математическом режиме), и будет действовать в точности как “бинарное отношение”.

Несколько символов в этих шрифтах заменяют символы, которые определены в `AMS-TeX` в виде комбинации символов, доступных из семейства шрифтов `Computer Modern`. Чтобы для замены существующего определения использовать `\newsymbol`, имя нужно сначала сделать “неопределенным”. Команды, к которым это относится, и символы, для которых это должно быть сделано, приведены ниже. Переопределение выполняется автоматически в стиле `\amsprt`, либо посредством `\UseAMSsymbols`.

Первая литера кода “ID” указывает, к какому семейству принадлежит этот символ: символу из семейства `msam` соответствует 1 в первом разряде, а символ из семейства `msbm` имеет первой литерой 2.

Все перечисленные символы должны использоваться в математическом режиме.

• Строчные греческие буквы

\mathbb{F} 207A `\digamma` \mathbb{K} 207B `\varkappa`

• Буквы еврейского алфавита

\beth 2069 `\beth` \gimel 206A `\gimel`
 \daleth 206B `\daleth`

• Мозаика символов

\hbar 207E <code>\hbar (U)</code>	\backprime 1038 <code>\backprime</code>
\hslash 207D <code>\hslash</code>	\varnothing 203F <code>\varnothing</code>
\triangle 134D <code>\vartriangle</code>	\blacktriangle 104E <code>\blacktriangle</code>
∇ 104F <code>\triangledown</code>	\blacktriangledown 1048 <code>\blacktriangledown</code>
\square 1003 <code>\square</code>	\blacksquare 1004 <code>\blacksquare</code>
\lozenge 1006 <code>\lozenge</code>	\blacklozenge 1007 <code>\blacklozenge</code>
\circledS 1073 <code>\circledS</code>	\bigstar 1046 <code>\bigstar</code>
\angle 105C <code>\angle (U)</code>	\sphericalangle 105E <code>\sphericalangle</code>
\measuredangle 105D <code>\measuredangle</code>	
\nexists 2040 <code>\nexists</code>	\complement 107B <code>\complement</code>
\mho 2066 <code>\mho</code>	\eth 2067 <code>\eth</code>
\Finv 2060 <code>\Finv</code>	\diagup 231E <code>\diagup</code>
\Game 2061 <code>\Game</code>	\diagdown 231F <code>\diagdown</code>
\Bbbk 207C <code>\Bbbk</code>	

Символы \angle и \hbar из этих новых шрифтов заменяют символы, показанные на с. 249 (убедитесь, что вы используете стиль `\amsrpt` или что строки

```
\undefine\angle
\newsymbol\angle 105C
\undefine\hbar
\newsymbol\hbar 207E
```

в вашем файле присутствуют). Такой новый символ `\angle` будет правильно менять размер в верхних и нижних индексах.

• Бинарные операторы

$\dot{+}$ 1275 <code>\dotplus</code>	\times 226E <code>\ltimes</code>
\setminus 2272 <code>\smallsetminus</code>	\times 226F <code>\rtimes</code>
\Cap 1265 <code>\Cap, \doublecap</code>	λ 1268 <code>\leftthreetimes</code>
\Cup 1264 <code>\Cup, \doublecup</code>	λ 1269 <code>\rightthreetimes</code>
$\bar{\wedge}$ 125A <code>\barwedge</code>	λ 1266 <code>\curlywedge</code>
\veebar 1259 <code>\veebar</code>	Υ 1267 <code>\curlyvee</code>
$\bar{\wedge}$ 125B <code>\doublebarwedge</code>	
\boxminus 120C <code>\boxminus</code>	\ominus 127F <code>\circleddash</code>
\boxtimes 1202 <code>\boxtimes</code>	\otimes 127E <code>\circledast</code>
\boxdot 1200 <code>\boxdot</code>	\odot 127D <code>\circledcirc</code>
\boxplus 1201 <code>\boxplus</code>	\cdot 1205 <code>\centerdot</code>
\div 223E <code>\divideontimes</code>	\intercal 127C <code>\intercal</code>

• Бинарные отношения

\leqq 1335 <code>\leqq</code>	\geqq 133D <code>\geqq</code>
\leqslant 1336 <code>\leqslant</code>	\geqslant 133E <code>\geqslant</code>
\leslantless 1330 <code>\leslantless</code>	\gtrsim 1331 <code>\eqslantgtr</code>
\lessssim 132E <code>\lessssim</code>	\gtrsim 1326 <code>\gtrsim</code>
\lessapprox 132F <code>\lessapprox</code>	\gtrapprox 1327 <code>\gtrapprox</code>
\approx 2375 <code>\approxeq</code>	
\lessdot 236C <code>\lessdot</code>	\gtrdot 236D <code>\gtrdot</code>
\lll, \lllless 136E <code>\lll, \lllless</code>	\ggg, \gggtr 136F <code>\ggg, \gggtr</code>
\lessgtr 1337 <code>\lessgtr</code>	\gtrless 133F <code>\gtrless</code>
\lesseqgtr 1351 <code>\lesseqgtr</code>	\gtreqless 1352 <code>\gtreqless</code>
\lesseqqgtr 1353 <code>\lesseqqgtr</code>	\gtreqqless 1354 <code>\gtreqqless</code>
\doteqdot, \Doteq 132B <code>\doteqdot, \Doteq</code>	\eqcirc 1350 <code>\eqcirc</code>
\risingdotseq 133A <code>\risingdotseq</code>	\circeq 1324 <code>\circeq</code>
\fallingdotseq 133B <code>\fallingdotseq</code>	\triangleq 132C <code>\triangleq</code>
\backsim 1376 <code>\backsim</code>	\thicksim 2373 <code>\thicksim</code>

⋈	1377	\backsimeq	≈	2374	\thickapprox
⊂	136A	\subseteq	⊃	136B	\supseteq
⊆	1362	\Subset	⊇	1363	\Supset
⊄	1340	\sqsubset	⊅	1341	\sqsupset
⋈	1334	\preccurlyeq	⋉	133C	\succcurlyeq
⋈	1332	\curlyeqprec	⋊	1333	\curlyeqsucc
⋈	132D	\precsim	⋋	1325	\succsim
⋈	2377	\precapprox	⋌	2376	\succapprox
△	1343	\vartriangleleft	▷	1342	\vartriangleright
△	1345	\trianglelefteq	▷	1344	\trianglerighteq
⊖	130F	\vDash	⊖	130D	\Vdash
⊖	130E	\Vvdash			
☺	1360	\smallsmile	∣	2370	\shortmid
☹	1361	\smallfrown	∥	2371	\shortparallel
⊖	136C	\bumpeq	∕	1347	\between
⊖	136D	\Bumpeq	⊖	1374	\pitchfork
α	135F	\varpropto	∩	237F	\backepsilon
◀	134A	\blacktriangleleft	▶	1349	\blacktriangleright
∴	1329	\therefore	∴	132A	\because

• Отношения отрицания

≠	2304	\nless	≠	2305	\ngtr
≠	2302	\nleq	≠	2303	\ngeq
≠	230A	\nleqslant	≠	230B	\ngeqslant
≠	2314	\nleqq	≠	2315	\ngeqq
≠	230C	\lneq	≠	230D	\gneq
≠	2308	\lneqq	≠	2309	\gneqq
≠	2300	\lvertneqq	≠	2301	\gvertneqq
≠	2312	\lnsim	≠	2313	\gnsim
≠	231A	\lnapprox	≠	231B	\gnapprox
≠	2306	\nprec	≠	2307	\nsucc
≠	230E	\npreceq	≠	230F	\nsucceq
≠	2316	\precneqq	≠	2317	\succneqq
≠	2310	\precnsim	≠	2311	\succnsim
≠	2318	\precnapprox	≠	2319	\succnapprox
≠	231C	\nsim	≠	231D	\ncong
⋈	232E	\nshortmid	≠	232F	\nshortparallel
⋈	232D	\nmid	≠	232C	\nparallel

⋈	2330	<code>\nvdash</code>	⋈	2332	<code>\nvDash</code>
⋈	2331	<code>\nVdash</code>	⋈	2333	<code>\nVDash</code>
↯	2336	<code>\ntriangleleft</code>	↯	2337	<code>\ntriangleright</code>
↯	2335	<code>\ntrianglelefteq</code>	↯	2334	<code>\ntrianglerighteq</code>
⊆	232A	<code>\nsubseteq</code>	⊆	232B	<code>\nsubseteq</code>
⊆	2322	<code>\nsubseteqq</code>	⊆	2323	<code>\nsubseteqq</code>
⊂	2328	<code>\subsetneq</code>	⊂	2329	<code>\subsetneq</code>
⊂	2320	<code>\varsubsetneq</code>	⊂	2321	<code>\varsubsetneq</code>
⊂	2324	<code>\subsetneqq</code>	⊂	2325	<code>\subseteqq</code>
⊂	2326	<code>\varsubsetneqq</code>	⊂	2327	<code>\varsubseteqq</code>

• Стрелки

⇐	1312	<code>\leftleftarrows</code>	⇒	1313	<code>\rightrightarrows</code>
⇐	131C	<code>\leftrightharpoons</code>	⇐	131D	<code>\rightleftarrows</code>
⇐	1357	<code>\Lleftarrow</code>	⇒	1356	<code>\Rrightarrow</code>
↵	1311	<code>\twoheadleftarrow</code>	→	1310	<code>\twoheadrightarrow</code>
↵	131B	<code>\leftarrowtail</code>	↵	131A	<code>\rightarrowtail</code>
↶	1322	<code>\looparrowleft</code>	↷	1323	<code>\looparrowright</code>
⇐	130B	<code>\leftrightharpoons</code>	⇐	130A	<code>\rightleftharpoons (U)</code>
↶	2378	<code>\curvearrowleft</code>	↷	2379	<code>\curvearrowright</code>
↶	1309	<code>\circlearrowleft</code>	↶	1308	<code>\circlearrowright</code>
↵	131E	<code>\Lsh</code>	↵	131F	<code>\Rsh</code>
⇐	1314	<code>\upuparrows</code>	⇐	1315	<code>\downdownarrows</code>
↵	1318	<code>\upharpoonleft</code>	↵	1316	<code>\upharpoonright, \restriction</code>
↵	1319	<code>\downharpoonleft</code>	↵	1317	<code>\downharpoonright</code>
↵	1328	<code>\multimap</code>	↵	1320	<code>\rightsquigarrow</code>
↵	1321	<code>\leftrightsquigarrow</code>			

Символ \rightleftharpoons из этих новых шрифтов заменяет символ со с. 250 (убедитесь, что вы используете стиль `\amsrpt` или что строки

```
\undefine\rightleftharpoons
\newsymbol\rightleftharpoons 130A
```

в вашем файле присутствуют). Такой новый символ `\rightleftharpoons` будет правильно менять размер в верхних и нижних индексах (хотя маловероятно, что он вам может там понадобиться), тогда как старый — нет.

• Отрицания стрелок

↵	2338	<code>\leftarrow</code>	↵	2339	<code>\rightarrow</code>
↵	233A	<code>\Leftarrow</code>	↵	233B	<code>\Rightarrow</code>
↵	233D	<code>\leftrightarrow</code>	↵	233C	<code>\Lefttrightarrow</code>

Приложение Н. TUG — группа пользователей ТҒХ'а

Группа пользователей ТҒХ'а — ТҒХ Users Group (TUG) — была организована для обмена информацией между пользователями пакета ТҒХ относительно возникающих проблем и способов их решения. Начиная с 1980 года эта организация публикует свой журнал *TUGboat*, который в настоящее время выходит 4 раза в год. Сеть “координаторов-установочников” TUG'а помогает приспособить ТҒХ к работе на различных машинах; периодически организуются краткие курсы обучения работе с пакетом ТҒХ; не реже одного раза в год проводятся общие конференции членов TUG'а. [В России функционирует Ассоциация *CyrTUG* пользователей кириллического ТҒХ'а, которая ставит перед собой аналогичные задачи. См. рекламу в конце данного издания. — *Перев.*]

Вопросы о вступлении в TUG и о том, как подписаться на *TUGboat*, направляйте по адресу

ТҒХ Users Group
P.O. Box 9506
Providence, RI 02940 USA.

Приложение I. На помощь!

В этом учебнике были указаны самые разнообразныe ловушки, которых вам следует избегать, и неверные шаги, которых не следует делать. Но, невзирая на все эти предостережения, то или иное сообщение об ошибке может загнать вас в тупик, либо окажется, что вы напрасно тратите силы, пытаясь получить при помощи AMS-TeX 'а особые эффекты. Безрезультатно загубив изрядное количество времени и усилий, еще раз загляните в приложение Н. Возможно, кто-нибудь из TUG'a уже сталкивался с подобной проблемой либо способен объяснить, в чем ваша ошибка.

Разумеется, причиной того или иного таинственного поведения AMS-TeX 'а может оказаться ошибка в программе, а не ваше непонимание. Если вы уверены, что нашли изъян в AMS-TeX 'е, попробуйте связаться с кем-нибудь из Американского математического общества.

Но ни в коем случае, милый читатель, не пытайтесь наладить контакт с автором этого учебника. Отсутствие у Н эрудиции в полной мере выразилось в стиле данной книги, и было представлено столь очевидно, что любые попытки осмысленного диалога с Н будут бесплодными, тем более что О — не более чем псевдоним, взятый группой изгоев, не склонных к воздержанию и достигших на почве компьютерной подготовки изданий последней стадии фанатизма.

Предметный указатель

Некоторые из управляющих слов, приведенных ниже, снабжены примечанием [`\redefine` запрещено], так как они используются внутри *AMS-TeX*'а и придание им новых значений может вызвать неразбериху. Кроме того, большинство управляющих *символов* также не должны переопределяться. Вы можете свободно развлекаться с `\` и `\<`, `\>`, `\+` и `\=`, так же как и `\[`, `\]`, `\(`, `\)`, `\?` и `\0`, ..., `\9` — и этого достаточно!

- аббревиатуры
 - пробелы после 41
- абзац 20, 22
 - в математическом режиме 56
 - формулах 135
 - сдвиг влево 123
- абсолютное значение 50
- автоматическое форматирование
 - отмена 124–125
- автор 180, 181
- ажурный шрифт, или “жирный мелом на доске” (blackboard bold) 178, 253
- акцент
 - акут 44
 - венгерский умлаут 44
 - гачек 44
 - гравис 44
 - краткость 44
 - тильда 44
 - точка “над” 44
 - “под” 44
 - умлаут 44
 - цедилла 44
 - циркумфлекс, или “крышка” 44
 - черта “над” 44
 - “под” 44
- акценты 44, 47
 - в виде верхнего индекса 126
 - математическом режиме 53, 125–128
 - различных шрифтах 46
 - двойные 127–128
 - широкие варианты 126
- аннотация
 - в `amsppt` 183
- апостроф 15
- перенос с 198
- аргумент
 - в виде литеры
 - для `\footnote` 161, 162
 - `\footnotemark` 161, 162
 - `\footnotetext` 161, 162
 - `\item` 167
 - `\tag` 154
 - `\define` 119–121
 - пробел после 120
 - управляющего символа 45
 - управляющей
 - последовательности 29, 118, 119, 211, 234
 - пробелы перед 70, 120, 245
 - факультативный
 - для `\footnote` 163, 164
 - `\footnotemark` 163, 164
 - `\footnotetext` 163, 164
 - `\item` 167
- библиография 190, 239–243
- бинарное отношение, *см.*
 - отношения
- бинарный оператор, *см.* операторы
- биномиальные коэффициенты 73
- большие операторы 75, 129–131, 250
 - с многострочными пределами 76
- больший пункт 141
- буквы в математическом режиме 16, 49, 89–96
 - еврейского алфавита 254
- буферный пробел 129, 130
- бэкслэш 9, 21, 44
- вертикальный пробел 131–132
 - в `\align` 131, 189

- \aligned 131, 189
- \cases 132
- \gather 131
- \matrix 131
- \Sb 131
- между строками выключной формулы 132
- верхние индексы 63-69
 - возможная западня 69
 - расположение на разных расстояниях 67
 - с \overbrace или \underbrace 140
 - \overset и \underset 147
- верхний индекс слева 167
- возврат каретки (carriage-return) 16, 19, 22, 24, 25
 - в выключных формулах 60
 - математическом режиме 50
- восклицательный знак
 - в математических формулах 138
- вставки 132-134
- выключение
 - в математическом режиме 60, 97
 - разрыв страниц в 158
 - сообщение об ошибке 220
 - дробей 70
 - с более чем одной системой выравниваемых формул 99
 - выравниваемыми формулами 98
- выключные формулы
 - абзац после 206
 - вертикальный пробел внутри 131-132
 - несколько формул, выключенных по центру 101
 - предотвращение разрыва страниц после 158
 - сноска в 161, 162
 - текст внутри 123, 124
- выходные данные 181, 182
- глаголы 50
- глобальная команда 129, 131, 154, 175
- горизонтальные
 - пробелы 138
 - скобки 139, 140
- готический шрифт (German Fraktur) 178, 193, 252
- греческие буквы
 - прописные 248
 - другое написание 53
 - наклонные 248
 - полужирные 177, 248
 - строчные 248, 254
 - другое написание 53
 - полужирные 177
- группа 26, 28
 - пустая 27
 - \define внутри 116
 - \end внутри 199
- группировка элементов 26, 27, 51
- двоеточие
 - в математическом режиме
 - пробелы вокруг 92, 169, 203
- двойные
 - акценты 127, 128
 - вертикальные черты 84, 107, 203
 - кавычки 16, 20, 34, 44, 195, 209
 - комножества 249
 - левые кавычки 209
 - могильные крестики 43, 47, 253
 - сноски 161
 - стрелки вверх 155
 - вверх-вниз 155
 - вниз 155
- декларирование, см. также \proclaim 140
- дефис, см. также короткое тире (en-dash) и длинное тире (em-dash) 20, 21
 - в начале строки 159
 - разрыв строки после 42
- дифференциалы 79
- длина страницы
 - в amsppt 179
- длина строки
 - в amsppt 179
- длинное тире 20, 24, 142, 194
 - в конце входной строки 25

- предотвращение разрыва строки
 - после 42
- доказательства 187
- дополнительное условие 80
 - в выравниваемых формулах 136, 137
- допустимое отклонение для
 - растянутых строк 34
- дроби 70–71, 140–141
 - в верхних индексах 72
 - внутри дробей 71
 - непрерывные 151
 - с более толстой чертой 74
 - через косую черту 72
- дужка 45
- дюйм 141
- единицы измерения 141, 142
 - - принтера 141
 - принятые в континентальной Европе 141
- единичная кавычка 16, 20, 23, 24
- жирный шрифт для бедных *см.* ажурный шрифт
- заголовки 185
- запятые 58
 - в математическом режиме 58
 - пробел после 52, 205–206, 214
 - точки перед 170
- звездочка 16, 52
 - в математическом режиме 52
- знак абзаца ¶, *см. также* \P 21, 43
 - бесконечности (∞) 52
 - интеграла, *см.* \int
 - “минус” 16, 20, 21, 49
 - параграфа 43
 - суммы, *см.* \sum
- знаменатель 70, 92
- имя файла
 - расширение 30, 32
- исправление ошибки 38
- кавычки 16, 20, 189
 - единичные 3
 - знаки препинания с 194
 - внутри кавычек 48
 - слева 15, 16
- квадрат, *см. также* \quad 80
- квадратный корень 82, 138
- ключевые слова
 - в `amsrpt` 183
- Кнут, Дональд 237
- колонтитулы 142–143, 179
 - в `amsrpt` 143, 179
 - монографиях 143, 192
 - укороченная форма 143, 192
- команда глобальная 129, 130, 154, 175
- комментарий, *см. также* % 24, 143, 199
- коммутативная диаграмма 144–147
- конец предложения 41
 - прописные буквы 48
- корни 82, 140, 138, 147
 - подправить расположение 147
- короткое тире 20, 24, 194
 - в \tag 97
 - предотвращение разрыва строки после 42
- косая черта
 - в математических формулах 49, 50
 - обратная, *см. также* бэкслэш 155
 - разрыв строки после 159
- крышка, *см.* акцент циркумфлекс
- курсив 26, 28, 29
 - в математическом режиме 89
- левый край
 - абзацы 123
 - вхождение в содержание 184, 185
 - заголовков 185, 186
 - столбцы в \matrix 107, 108, 148
 - числители 152
- лигатуры
- лист бумаги
 - положение страницы на 158

- макетирование книги, см. также
 - монография 190–192
- малая матрица 147
- массивы, см. `\matrix` 106
- математические акценты, широкие варианты, см. акценты
- формулы 49, 50
 - в тексте 49–57
 - выключные 56
 - многострочные 98–105
 - однострочные 59
 - пробелы в 79, 80
 - вокруг 202
 - текст в 89–96
- математический режим 49–57
 - акценты 125
 - широкие варианты 126
 - выключение 60, 61
 - пустая строка в 204
 - сообщение об ошибке 37, 40, 56, 164
 - шрифты в 156–159, 175–178
 - `\define` в 110
 - `\par` в 55
- матрица 106, см. `\matrix`
- межсловный пробел
 - обычный 42, 48, 130, 131, 205, 239
- метки, расположенные слева 97, 98, 153, 154
- миллиметр 141
- многострочные выключные формулы 150
 - вертикальные пробелы в 135
- пределы
 - под знаком суммы 76
- многоточие (`\dots`)
 - в конце формулы 172
 - `\matrix` 148–150
 - пробелы 149–150
 - горизонтальная строка точек 107
 - между знаками интеграла 170
 - операторами и отношениями 170
 - умножаемыми символами 171
 - на строке 170
 - перед запятой или точкой с запятой 170
 - по вертикали 107
 - горизонтали 107
 - диагонали 107
 - посередине 170
- мозаика символов 249, 253, 254
- монографии 239
 - колонтитулы 192
- над, один символ над другим 165
- название 180
- наклонные буквы
 - в математических формулах 176
- неопределенная управляющая последовательность 37
- непрерывные дроби 151, 152
- нижние индексы 63–69
 - возможная западня 69
 - расположение на разных расстояниях 67
 - с `\overbrace` или `\underbrace` 140
 - `\overset` и `\underset` 147
- номер уравнения, см. `\tag`
- нумерация страниц 156, 179
- формул 153, 154
 - по центру для формул, разбитых на несколько строк 154
 - расположение номеров 97, 98
 - трактовка номеров в качестве математических формул 154
- обелиск, см. могильный крестик
- обозначение штриха 68
- ограничители 84–155, 218, 251, 253
 - большие 88
 - в качестве отношений 128, 129
 - вокруг дробей 141
 - полужирные 177, 178
 - пробелы вокруг 84
 - пустые 86, 87
 - различного размера 128
- однострочная формула 61
- операторы 75, 95, 155, 248, 255
 - большие 75

- полужирные 177
- пределы на 75
- пробелы вокруг 138–139
- ответ на сообщение об ошибке, см.
- сообщение об ошибке
- отмена, автоматическое
- форматирование, см. `\nofrills`
- отношения
 - бинарные 50, 129, 177, 249, 255
 - в качестве ограничителей 129
 - отрицания 256–257
 - полужирные 177
 - пробел вокруг 139, 168
 - точки после 170
- отрицания 256, 257
- отрицание бинарного оператора 53
- стрелок 257
- отрицательная тонкая шпация 138
- пайка 141
- пакетный режим 31, 164
- переводчик
 - в `amspp` 183
- перенос 20
 - в математическом режиме 16, 21, 49, 50
 - `\tag` 97
 - возможный 35
 - разрыв строки после 42
 - запрещение переноса 42
- переносы 34, 155, 158
- переполнение 34, 56, 58, 137, 156, 157
 - `\hbox` 34, 56, 137, 157
 - `\vbox` 157
- печать фигурных скобок 51
- подписи 189
 - под рисунками и/или в таблицах 132, 133
 - ширина 133
- подчеркивание 63
- слов 29
- полужирный шрифт
 - ажурный 178, 252, 253
 - акценты 178
- буквы в математических формулах 173, 175–177, 244, 245
 - греческие прописные 177, 248
 - эйлеровы 255
- математические символы 176–178
- тип 26
- помощь в `TeX`'е 39
- поправка на курсив 28
- после `\endroster` 167
- преамбула 179–180, 191
- пределы
 - суммирования 75
 - многострочные 76
 - у больших операторов 75–78, 129, 130, 133
- предметная классификация в `amspp` 183
- предосторожности для `\define` 114, 116
- предотвращение разрыва страниц, см. разрыв страницы
- правый край
 - столбцы в `\matrix` 107, 108
 - числители 152
- пробелы
 - в выключных формулах 60, 61
 - математическом режиме 43, 44, 50
 - после запятой 52
 - `\operatorname` 96
- вертикальные 131–132
- во входном файле 23, 24
 - в начале строки 19
 - между словами 19
 - после знаков препинания 19
- вокруг операторов 50, 51, 67, 139
- отношений 50, 51, 139
- ~ 41
- горизонтальные 138–139
- для рисунка 132–134
- между кавычками 48
- над и под пределами 129–130
- перед аргументом управляющей последовательности 45, 70, 244, 245

- после аргумента управляющего символа 45
- точек 41
- управляющего символа 23, 24, 45
- - слова 21–24, 45
- проверка синтаксиса 156
- производные 118
- прописные буквы 15
 - ажурные 252, 253
 - в конце предложения 48
 - греческие 242
 - наклонные 242
 - полужирные 177
 - рукописные
- прямые буквы
 - в математическом режиме 89–96
 - нижних индексах 92
 - для операторов 93–96
- пункт 141
 - большой 141
 - Дидо 141
 - масштабный 141
 - принтера 33
- пунктуация
 - в математическом режиме 49
 - после прописной буквы
 - в конце предложения 48
 - с кавычками 194
- пустая группа (`{}`) 27, 64, 67
- строка 16, 20, 22, 24
 - в математическом режиме 135
 - `\foldedtext` 124
 - `\footnote` 183
 - перед `\block` 189
 - после `\endroster` 167
- размер
 - входных строк 19
 - для верхних и нижних индексов 62
 - выключных формул 70, 71
 - текста 66, 67
 - страницы 157
 - шрифта набора 158
- размерность нуль 142
- разрыв
 - слов 158
 - страниц 158
 - между строками в `\align` 134
 - строк 35, 58, 159
 - перед штрихом 159
 - после косой черты 159
 - формул 58, 103, 104, 160
 - страницы 158
 - предотвращение 158
 - принудительный после выключения 134
 - разрешение внутри `\align` 134
 - строк 41, 159
 - в выключных формулах 102–104
 - математическом режиме 58
 - сведениях об издании 181, 182
 - после дефисов и тире 42
 - предотвращение 159
 - в математическом режиме 58
 - принудительный 35
 - в математическом режиме 58
 - разрешение 159
 - в математическом режиме 58
 - расположение 63
 - `\dots` 171
 - индексов на разных расстояниях от буквы 67
 - номера страницы на листе бумаги 156
 - распорки 160
 - расширение
 - имени файла 30, 32
 - рисунки 132, 133
 - место для 132
 - надписи на 132, 133, 189
 - рукописные буквы 178, 245, 248
- сантиметр 141
- сведения об издании 180, 185, 190, 191
 - вспомогательный материал 160
- связка 50
- сигма, прописная буква 75
- символ *Лежандра* 74, 141
- символы 152, 153
 - новые 152, 153

- составные 165
- специальные 43, 44
- друг над другом 168
- с подписями 165
- нижние и верхние индексы 165
- скобки
 - большие 83
 - в математических формулах 51
 - "mod" 151
 - после точки 200
 - размер зависит от размера формулы 83
- сноски 161-164, 189
 - в выключных формулах 162
 - двойные 161
 - исчезновение 162
 - маркеры 162
 - несколько сносок в выключных формулах 162, 163
 - нумерация 161, 163
 - шрифты в 26
- содержание 184
 - выравнивание 184
 - в монографиях 192
 - для статей 183-185
- сообщение об ошибке 37, 39, 56, 164
- ответ на 164, 165
- Display math should end with `$$`. 220
- Double superscript 208
- Extra `\right.` 217
- Extra `}`, or forgotten `$` 207, 211, 220
- Missing `$` inserted. 56, 57, 165, 207, 212
- Missing `}` inserted. 211, 220
- Missing delimiter (`.`inserted) 217
- Overfull
 - `\hbox` 34, 56, 137, 157
 - `\vbox` 157
 - `box` 58, 156
- Paragraph ended before `\next` `@` was complete 182
- Runaway argument? 74, 212
- TeX capacity exceeded 114
- Undefined control sequence. 37, 38
- AMS-TeX 114, 183
- Too many `}`'s 39
- составные символы, см. также дробь 165
- специальные литеры в TeX'e 16
- специальный знак окончания доказательства, см. `\qed`
- список, см. `\roster`
- стилевой файл 179, 180
- стиль "препринт", см. также `amsppt` 179
- `amsppt` 32, 33, 126
- `plain` 246
- страница
 - высота 157
 - номер на 156
 - положение на листе бумаги 157
 - размер 157
 - ширина 157
- стрелки 81, 168, 203, 204, 250, 257
 - в качестве ограничителей 154, 251
 - коммутативных диаграммах 145, 146
 - вертикальные 144
 - горизонтальные 144
 - некоторой длины 146
 - укороченные 147
- выражения над и под 168
- над формулами 81
- отрицания 257
- под формулами 81
- строочные буквы 15, 16
 - греческие 53, 248, 253
- стяжка 168-169
- ссылка на конкретный пункт списка 167
- - номер формулы 153, 154
- ссылки библиографические 190, 239
- таблицы 189
- текст
 - внутри выключных формул 123, 124

- между выравниваемыми формулами 134, 123, 124
- размер 66
- трактовка номеров формул 154
- текущий шрифт 93, 96
- тензорный анализ 67
- теорема *Пифагора* 82
- теоремы, см. `\proclaim`
- теоретико-множественные обозначения 92, 169
- `TeX` 10, 23, 24
- `TeX`-обработка 31, 32
- тильда 41, 159, 160, 161
 - после аббревиатуры 42
- точки (dots) см. многоточие
- точки (periods)
 - в аббревиатурах 41
 - выключных формулах 61
 - за которыми следуют кавычки 41
 - - - скобки 41
 - после прописной буквы с акцентом 48
 - прописных букв
 - указывают на конец предложения 48
 - не являются концом предложения 48
- угловые скобки 74, 85
- управляющее слово 21, 23, 24
- управляющие последовательности 12, 16, 23, 24, 164
 - образование новых 110
 - определение 110, 114
 - внутри математической формулы 116
 - с аргументами 29
- символы 23, 24, 114, 195
- уровень полиграфического исполнения 175
- файл транскрипции 33, 39, 157
- файлы 155
 - ввода 164, 165
- факультативный аргумент
 - для `\footnote` 164
 - `\footnotemark` 164
 - `\footnotetext` 164
 - `\item` 181
- фигурные скобки 17, 84
 - в математическом режиме 51
 - `\define` 112
 - `\predefine` 113
 - вокруг операторов 202, 224
 - горизонтальные 163, 164
 - для группировки 26
 - дополнительные 78, 126, 230, 231
 - печатать 51, 247
 - размер соответствует формуле 84
- формулы 102–105
 - абзацы в 135
 - выключение 97
 - выравнивание 97, 98
 - более чем одно 99
 - - по одной позиции 135
 - несколько пар 136
 - номера (метки) 100
 - текст между строками 134
 - заключенные в рамочку 174, 175
 - несколько формул,
 - расположенных по центру 101
 - нумерация 101
 - номера (метки) 97
 - разбитые на несколько строк 102–103
 - разрыв 160
 - страницы после 158
 - точка в конце 170
 - `Overfull boxes` 137
- цитата 189
- цицера 141
- черный прямоугольник 34, 175
- числа
 - для нумерации страниц 157
 - - формул 98, 102–105
 - старинного написания 175
 - полужирные 178
- числители 66, 69–71
 - сдвиг влево или вправо 152
- число *Фибоначчи* 83, 92, 221

- Эйлера 74, 141
- шпация толстая, отрицательная 139
- тонкая, отрицательная, см. также _ 79, 94
 - после точек в конце формулы 172
 - - перед правым ограничителем 172
- шрифты
 - в математическом режиме 175–178
 - сносках 26
 - изменение 26, 89
 - в математическом режиме 244, 245
 - для специальных символов 252–257
 - текущие 92, 96
- электронный адрес 182
- язык иностранный 16, 44
- _ 16, 23, 25, 60
- _ 24, 162, 195
- # 16, 43, 118, 119
- \# (#) 43, 47
- \$, см. также \\$ 16, 23, 47, 49
- \\$ (\$) 23, 43, 195
- \$\$ 12, 16, 60
- % 16, 24–25, 199, 200
- \% 47
- & 16, 43, 98
 - в \align, 98
 - \alignat, 136
 - \CD, 144
 - \gather, 101
 - \matrix, 106, 149
- \& (&) 43, 47
- && 148
- ' 15, 17, 20, 247
 - в математическом режиме 72
- \' ('акцент) 44, 126, 247
- '' (") 195
- ' 15, 16, 20, 247
- \' ('акцент) 44, 247
- '' (") 195
- " (двойные кавычки) 16, 44, 162, 163, 167
- \" ("акцент) 44, 46
- (15, 50–51, 83, 176, 177, 251
-) 15, 50–51, 83, 176, 177, 251
- [15, 17, 28, 51, 83, 164, 167, 176, 177, 247, 251
- [1], 33
-] 15, 17, 28, 51, 83, 84, 164, 167, 176, 177, 247, 251
- { 16, 17, 26, 39, 40
- \{ ({} 16, 26, 43, 47, 51, 84, 177, 244, 247, 251
- } 16, 17, 26, 27, 38–40
- \} (}) 16, 26, 43, 47, 51, 84, 177, 244, 247, 251
- + 16, 141, 176
 - в единицах измерения 141
 - полужирный 177
- \+, 232, 244
- , 15, 20, 50, 176
 - в единицах измерения 141
 - \tag 97
 - полужирный 177
- \- 35, 158
- (-) 20, 24, 194
 - в \tag 97
- (—) 20, 24, 194
- * 16, 52, 67, 68, 176
 - в математике 52
 - \tag 224
 - полужирный 177
- * запрос 199
- * 160
- ** запрос 32, 198
- /, см. также косая черта 16, 176, 251
 - переменного размера 87
 - полужирная 177
- \| 28, 196
- | (вертикальная черта) 16, 17, 49, 84, 154, 176–178, 203, 249, 250
 - в сообщениях об ошибке 34
- \| (||) 84, 154, 178, 203, 244, 247, 251

- \ 16, 17, 21, 44
- \\
 - в \align 98
 - \author 181
 - \CD 144, 145
 - \cfrac 151
 - \format 108
 - \gather 101
 - \head 186
 - \matrix 106
 - \Sb 76
 - \specialhead 186
 - \title 180
- < 16, 85, 176, 177, 251
- = 16, 176, 177
- > 16, 85, 176, 177, 251
- \, 79, 93, 96, 138, 139, 168, 244
- \>, 244
 - в тексте 79
- ., см. также ограничители,
 - пустые 176, 177
 - в библиографии 239
- \. 41, 48, 200, 239, 244
- ; 176, 177
- \; 139, 244
- : 176, 177
- \: 203
- ? 176, 177
 - в сообщениях об ошибках 38
- ?' (¿) 46
- !, 176 177
 - в сообщениях об ошибках 37
- \! 138, 139
- !' (¡) 46
- _ 16, 17, 76
- _ 16, 43, 47, 63, 247
- ~ 16, 17, 44, 63
- \^ (^ акцент) 44, 46, 126, 130
- ~ 16, 17, 41, 44, 48
- \~ (~ акцент) 44, 247
- © 16, 43
- \© (@), 43
- ©" 48
- ©- 42
- ©-- 42
- ©--- 42
- ©| 146
- ©<<< 168, 144, 145
- ©= 146
- ©>>> 144, 145, 168
- ©, 48
 - пробелы в математическом режиме 139
- ©. 48, 145, 239
- ©; 48
- ©: 48
- ©? 48
- ©! 48
 - пробелы в математическом режиме 139
- ©AAA 144
- ©\vert 146
- ©VVV 144
- \aa (å) 46
- \AA (Å) 46
- \abstract 181, 183, 213
 - использование \nofrills 125
- \accentedsymbol 128
- \acute (' акцент в математическом режиме) 125
- \Acute 127
- \acuteaccent, 247
- \address 181, 182
- \adjustfootnotemark 163, 164
- \ae (æ) 46
- \AE (Æ) 46
- Æsop 46
- \affil 181, 182
- \after 148
- al-Khwârizmî, Muḥammad ibn Mûsâ 46
- \aleph (ℵ) 249
- \align, 98, 106, 134
 - \tag в 98
 - внутри \gather 137
 - пробелы между строками 134
 - разрыв страницы внутри 134
 - текст между строками 134
- \alignat 135–137
- \aligned 99, 135
- \alignedat 137
- \allowdisplaybreak 134–136
- \allowdisplaybreaks 134

- `\allowlinebreak` 159
- `\allowmathbreak` 58
- `\alpha` (α) 53, 112, 177, 248
- `\amalg` (\amalg) 248
- American Mathematical Society* 179, 188
- `amsppt` 31, 170, 179–193
- `amsppt.sty` 32, 33
- `AMSSYM.TEX` 253
- `\AmSTeX` 24, 122
- `AMS-TeX` 24, 179
 - сообщения об ошибках 111, 112, 114, 182, 187
- `amstex.tex` 33
- `\and` 204, 248
 - в `\sideset` 167
- `\angle` (\angle) 153, 215, 249, 255
- Ångström, Anders Jonas 46
- `\approx` (\approx) 249
- `\approxeq` (\approx) 255
- `\arccos` 95
- `\arcsin` 95
- `\arctan` 95
- `\arg` 95
- `arg sinh` 96
- `\arrowvert` 155
- `\Arrowvert` 155
- Art of Computer Programming, The* 237
- `\ast` ($*$) 248
- `\asymp` (\asymp) 249
- `\author`, 161, 162, 181–182

- `\b` ($_$ акцент) 44, 112, 244
- `\B` ($\bar{_}$ акцент) 44, 127, 244
- `\backepsilon` (ϵ) 256
- `\backprime` (\prime) 254
- `\backsim` (\sim) 255
- `\backsimeq` (\simeq) 256
- `\backslash` (\backslash) 154, 203, 249–251
- `badness` 197
- `\bar` ($\bar{_}$ акцент в математике) 125–127
- `\Bar` 127
- `\barwedge` ($\bar{\wedge}$) 255
- `\Bbb` (ажурный, или жирный мелом на доске) 178, 253
 - `\Bbbk` (\mathbb{k}) 255
 - `\because` (\because) 256
 - `\beta` (β) 53, 113, 248
 - `\beth` (\beth) 253
 - `\between` (\O) 256
 - `\bf` [`\redefine` запрещено] 26, 29, 73, 175
 - `\big` 128–129
 - `\Big` [`\redefine` запрещено] 129
 - `\bigcap` (\bigcap) 75, 250
 - `\bigcirc` (\bigcirc) 248
 - `\bigcup` (\bigcup) 75, 250
 - `\bigg` 128
 - `\Bigg` 129
 - `\biggl` 88, 128, 236
 - `\biggm` 128
 - `\biggr` 88, 128, 236
 - `\bigl` 87, 104, 128
 - `\bigm` 128
 - `\bigodot` (\odot) 250
 - `\bigoplus` (\oplus) 250
 - `\bigotimes` (\otimes) 250
 - `\bigpagebreak` 131
 - `\bigr` 87, 128
 - `\bigscup` (\sqcup) 250
 - `\bigstar` (\star) 255
 - `\bigtriangledown` (∇) 215, 248
 - `\bigtriangleup` (\triangle) 215, 248
 - `\biguplus` (\uplus) 250
 - `\bigvee` (\vee) 250
 - `\bigwedge` (\wedge) 250
 - `\binom` 73
 - `\BlackBoxes` 175
 - `\blacklozenge` (\blacklozenge) 254
 - `\blacksquare` (\blacksquare) 254
 - `\blacktriangle` (\blacktriangle) 254
 - `\blacktriangledown` (\blacktriangledown) 254
 - `\blacktriangleleft` (\blacktriangleleft) 256
 - `\blacktriangleright` (\blacktriangleright) 256
 - `\block` 189
 - `\bmatrix` 107
 - `\bmod` 151
 - `\bold` 175–178, 245
 - `\boldkey` 176–178
 - `\boldsymbol`, 176–178
 - `\book`, 240–243
 - `\bookinfo` 242

- `\bookinquotes` 242
- `\bot` (\perp) 249
- `\botaligned` 135
- `\botcaption` 133, 189
- `\botfoldedtext` 124
- `\botshave` 130
- `\botsmash` [`\redefine` запрещено]
168, 169, 237
- `\bowtie` (\bowtie) 249
- `box`, `Overfull` 137, 156, 157
- `box`, `Underfull` 157, 197
- `\boxdot` (\boxdot) 255
- `\boxed` 174
- `\boxminus` (\boxminus) 255
- `\boxplus` (\boxplus) 255
- `\boxtimes` (\boxtimes) 255
- `bp` 141
- `\bracevert` 155
- `\breve` (\breve акцент в математическом
режиме) 125
- `\Breve` 127
- `\buffer` 130
- `\bullet` (\bullet) 248
- `\bumpeq` (\bumpeq) 256
- `\Bumpeq` (\Bumpeq) 256
- `\by` 240
- `\bysame` 240

- `\c` (\grave{c} , акцент) 44
- `\c` в `\format` 108
- `\cal` 245
- `\Cal` 178, 245, 248
- `\cap` (\cap) 75, 248
- `\Cap` (\Cap) 255
- CAPS и SMALL CAPS шрифт 124,
195
- `\captionwidth` 133
- `\cases` 109, 132
отличия между `plain` и `amstex`
246
пробелы после строк 132
- `cc` 141
- `\CD` 144–147
- `\cdot` (\cdot) 248
- `\cdots` (\cdots) 170, 246
- Čech, Eduard 46
- `\centerdot` (\cdot) 255
- `\CenteredTagsOnSplits` 154
- Cesàro, Ernesto 46
- `\cfrac` 151
- `\ChangeBuffer` 131
- `\chapter` 191
- `\check` (\checkmark акцент в математическом
режиме) 125, 126
- `\Check` 127
- `\checkmark` (\checkmark) 253
- Chern, S. S. 232
- Cherry, Lorinda L., 237
- `\chi` (χ) 248
- `\circ` (\circ) 248
- `\circeq` ($\overset{\circ}{=}$) 255
- `\circlearrowleft` (\circlearrowleft) 257
- `\circlearrowright` (\circlearrowright) 257
- `\circledast` (\circledast) 255
- `\circledcirc` (\circledcirc) 255
- `\circleddash` (\circleddash) 255
- `\circledR` (\circledR) 253
- `\circledS` (\circledS) 255
- `\cite` 190
- `\clubsuit` (\clubsuit) 249
- `cm` 141
- `\comment` 143
- `\complement` (\complement) 255
- `\cong` (\cong) 249
- `\coprod` (\coprod) 250
- `\copyright` (\copyright) 43
- `\cos`, 92, 95
- `\cosh`, 95
- `\cot`, 95
- `\coth`, 95
- `\csc`, 95
- `\cup` (\cup) 75, 248
- `\Cup` (\cup) 255
- `\curlyeqprec` (\curlyeqprec) 256
- `\curlyeqsucc` (\curlyeqsucc) 256
- `\curlyvee` (\curlyvee) 255
- `\curlywedge` (\curlywedge) 255
- `\curvearrowleft` (\curvearrowleft) 257
- `\curvearrowright` (\curvearrowright) 257
- CyrTUG* 6, 258

- d-size 70–72, 91, 106
- `\d` (\d , точка “под”) 44, 244
- `\D` (\D , точка “над”) 44, 244

- `\dag` (†), 43, 249
- `\dagger` (†), 248
- `\daleth` (ℵ), 254
- `\dasharrow` 253
- `\dashleftarrow` 253
- `\dashrightarrow` 253
- `\dashv` (⊥) 249
- `\date` 181, 182
- `\dbinom` 73
- `dd` 141
- `\ddag` (‡) 43, 248
- `\ddagger` (‡) 248
- `\ddddot` (⋯ акцент в математическом режиме) 125
- `\dddots` (⋯ акцент в математическом режиме) 125
- `\ddot` (¨ акцент в математическом режиме) 125
- `\Ddot` 127
- `\ddots` (⋮) 107
- `de Rham` 111
- `\dedicatory` 181, 182
- `\define` 110–122
 - внутри группы 116
 - математической формулы 116
 - определение управляющей последовательности для использования после `\dots` 172
- `\definition` 187
- `\deg` 95
- `\delta` (δ) 53, 248
- `\Delta` (Δ) 53, 215, 248
- `\demo` 187, 213
- `\det` 95
- `\dfrac` 72
- `\diagdown` (↘) 255
- `\diagup` (↗) 255
- `\diamond` (◇) 248
- `\diamondsuit` (♠) 249
- `\digamma` (Γ) 254
- `\dim` 95
- `\displaybreak` 134, 135
- `\displaylines` 246
- `\div` (÷) 248
- `\divideontimes` (⊛) 255
- `\document` 31, 181
- `\documentstyle` 31, 32, 179, 181, 191
- `\dot` (· точка “над” в математическом режиме) 125
- `\Dot` 127
- `\doteq` (≐) 249
- `\Doteq` (≐) 255
- `\doteqdot` (≐) 255
- `\dotplus` (⊕) 255
- `\dots`, см. также многоточие 47, 59, 170–172, 246
- `\DOTSB` [`\redefine` запрещено], 172
- `\dotsb` [`\redefine` запрещено], 170, 171
- `\dotsc` [`\redefine` запрещено], 171
- `\DOTSI` [`\redefine` запрещено], 172
- `\dotsi` [`\redefine` запрещено], 172
- `\dotsm` [`\redefine` запрещено], 151
- `\dotso` [`\redefine` запрещено], 170, 171
- `\DOTSX` [`\redefine` запрещено], 172
- `\doublebarwedge` (⌘) 255
- `\doublecap` (⊓) 255
- `\doublecup` (⊔) 255
- `\downarrow` (↓) 154, 250
- `\Downarrow` (⇓) 154, 250
- `\downdownarrows` (⇓) 257
- `\downharpoonleft` (⤵) 257
- `\downharpoonright` (⤴) 257
- `\dsize` [`\redefine` запрещено] 71–73, 78, 86, 210
- `.dvi` 33
- `dvi` файл 33
- `È. Cartan` 111
- `\ed` 241
- `\eds` 241
- `\eightpoint` [`\redefine` запрещено] 158
- `\ell` (ℓ) 52, 178, 249
- `em` 142
- `\email` 181, 182
- `\emptyset` (∅) 53, 249
- `\end` 199
- ***[для всего, что начинается с `\end`, `\redefine` запрещено]

- `\endabstract` 181, 183
- `\endaddress` 181, 182
- `\endaffil` 181
- `\endalign` 98, 106
- `\endaligned` 99
- `\endauthor` 181
- `\endblock` 189
- `\endbmatrix` 106
- `\endcaption` 133, 134
- `\endcases` 109
- `\endCD` 144–147
- `\endcfraction` 151
- `\endcomment` 143
- `\enddate` 181, 182
- `\endeddicator` 182
- `\enddefinition` 187
- `\enddemo` 186, 187
- `\enddocument` 32, 199
- `\endemail` 181, 182
- `\endexample` 187
- `\endgather` 101
- `\endgathered` 101
- `\endgraf` 124, 135, 182
- `\endhead` 185
- `\endkeywords` 181–183
- `\endmatrix` 106, 107
- `\endmultiline` 105
- `\endpmatrix` 106
- `\endproclaim` 187
 - сообщение об ошибке
 - относительно пропуска 187
- `\endref` 240
- `\endRefs` 239
- `\endremark` 187
- `\endroster` 167, 189
- `\endSb` 76
- `\endsmallmatrix` 147
- `\endSp` 76
- `\endspecialhead` 185
- `\endsplit` 103, 105
- `\endsubhead` 185
- `\endsubjclass` 181, 183
- `\endsubsubhead` 185
- `\endthanks` 181, 182
- `\endttitle` 180–182
- `\endtoc` 183, 192
- `\endtopmatter` 181
- `\endtranslator` 181, 183
- `\endVmatrix` 106
- `\endvmatrix` 106
- ***[для *всего*, что начинается с `\end`, `\redefine` запрещено]
- `\epsilon` (ϵ), 47, 270
 - другое написание 53
- `\equalign` 246
- `\eqcirc` (\equiv) 256
- `\eqslantgtr` (\gtrsim) 255
- `\eqslantless` (\lesssim) 255
- `\equiv` (\equiv) 53, 249
- Erdős, Pál 46
- `\errorstopmode` 164
- `\eta` (η) 248
- `\eth` (\eth) 255
- `ex` 142
- `\example` 187
- `\exists` (\exists) 249
- `\exp` 95
- `\fallingdotseq` (\fallingdotseq) 255
- `\finalinfo` 243
- `\Finv` (\Finv) 255
- `\flat` (b) 249
- `\flushpar` 123
- `\foldedtext` 123–124
- `\foldedwidth` 123
- `\footnote` 161–163, 173, 189
 - аргумент в виде литеры для 163
 - пробелы в 161, 162
 - факультативный для 163
 - отличие между `plain` и `\amstex` 246
- `\footnotemark` [`\redefine` запрещено] 161–163
 - аргумент в виде литеры для 164
 - факультативный для 163
- `\footnotetext` [`\redefine` запрещено] 161–163
 - аргумент в виде литеры для 163
 - факультативный для 163
- `\forall` (\forall) 249
- `\format` [`\redefine` запрещено] 108, 147–150
- `\frac` 70, 81, 237

- $\frac{\text{withdelims}}$ 141
 $\frac{\text{frak}}$ 178
 \frown (˘) 249
 \galley s 156
 \Game (⊖) 255
 γ (γ) 53, 248
 Γ (Γ) 53, 177, 248
 \gather 101, 137
 \align в 138
 \gathered 101
 \gcd 95
 \ge (\ge) 53, 249
 \geq (\ge) 53, 249
 \geqq (\ge) 255
 \geqslant (\ge) 255
 \gets (\leftarrow) 250
 \gg (\gg) 249
 \ggg (\ggg) 255
 \gggtr (\ggg) 255
 \gimel (\beth) 254
 \gnapprox (\gtrapprox) 256
 \gneq (\gtr) 256
 \gneqq (\gtr) 256
 \gnsim (\gtrsim) 256
 \goth 178
 $\grave{}$ (˘ акцент в математическом режиме) 125
 \Grave 127
 $\grave{\text{accent}}$ 247
 \greater 247
 \gtrapprox (\gtrapprox) 255
 \gtrdot (\gtr) 255
 \gtreqless (\gtr) 255
 \gtreqqlless (\gtr) 255
 \gtrless (\gtr) 255
 \gtrsim (\gtrsim) 255
 \gvertneqq (\gtr) 256

h, в ответ на сообщение об ошибке 38, 39, 74
H, в ответ на сообщение об ошибке 38, 39, 74
 $\mathring{}$ (ˆ акцент) 44
 haswidth 146
 $\hat{}$ (ˆ акцент в математическом режиме) 125–127
 \Hat 127, 128
 $\hat{\text{accent}}$ 247
 \hbar (\hbar) 153, 249, 254
 \hbox [\redefine запрещено]
 Overfull 34, 51, 137, 175
 Underfull 156, 159
 \hcorrection 157
 \hdots 107
 \hdotsfor 149
 \head 185
 в монографиях 192
 - содержания 184
 \headmark 143
 \heartsuit (\heartsuit) 249
Hölder's Inequality 72, 210
 \hom 95
 \hookleftarrow (\leftrightarrow) 250
 \hookrightarrow (\leftrightarrow) 250
 \hphantom 173
 \hslash (\hbar) 254
 \hyphenation 35, 158, 198

i (i) 46, 47, 126
i, в ответ на сообщение об ошибке 38
I, в ответ на сообщение об ошибке 38
 \idotsint ($\int \dots \int$) 77, 250
 \iff (\iff) 250
 \iiiint (\iiint) 77, 250
 \iiint (\iiint) 77, 250
 \iint (\iint) 77, 133, 139, 250
 \Im (\Im) 249
 \imath (\imath в математическом режиме) 126, 249
 \impliedby (\impliedby) 250
 \implies (\implies) 204, 250
 \in (\in) 52, 249
in 141
 \inbook 240, 241
 \inf 52, 95
 ∞ (∞) 52, 249
 \injl 95
 \innerhdotsfor 149
 \input 172

- `\int` (\int) 12, 77, 129, 250
- `\intercal` (\intercal) 255
- `\intertext` 124, 134–136
- `\iota` (ι) 248
- `\issue` 240
- `\it` [`\redefine` запрещено] 26, 29, 30, 73
- `\italic` 176
 - в верхних индексах 176
- `\item` 167, 189
 - с аргументом в виде литеры 167
 - факультативным аргументом 167
 - ссылка на 167
- `\item, plain TeX` 244
- `\j` (j) 46, 47
- `\jmath` (j in math) 126, 249
- `\jot` [`\redefine` запрещено] 132
- `\jour` 240
- Journal of the American Mathematical Society* 179, 183
- `\kappa` (κ) 53, 248
 - другое написание 53
- `\ker` 95
- Kernighan, Brian W. 237
- `\key` 239, 240
- `\keywords` 181, 183
- `\l` (l) 46
- `\l in \format` 108
- `\L` (L) 46
- `\lambda` (λ) 248
- `\Lambda` (Λ) 248
- `\land` (\wedge) 248
- `\lang` 240
- `\langle` (\langle) 85, 177, 216, 251
- `\lbrace` ($\{$) 51, 247, 248, 251
- `\lbrack` ($[$) 28, 51, 83, 247, 248, 251
- `\lceil` (\lceil) 84, 251
- `\lcfraction` 152
- `\ldots` (\dots) 170, 246
- `\le` (\leq), см. также `\leq` 53, 249
- `\left` [`\redefine` запрещено] 85, 129, 130, 154
- `\left(` 83
- `\left.` 86, 145
- `\left<` 85, 216
- `\left[` 83, 106
- `\leftarrow` (\leftarrow) 203, 250
- `\Leftarrow` (\Leftarrow) 250
- `\leftarrowtail` (\leftarrowtail) 257
- `\leftharpoondown` (\leftharpoondown) 250
- `\leftharpoonup` (\leftharpoonup) 250
- `\leftheadtext` 143, 192
- `\leftleftarrows` (\leftleftarrows) 257
- `\leftrightarrow` (\leftrightarrow) 250
- `\Leftrightarrow` (\Leftrightarrow) 250
- `\leftrightarrows` (\leftrightarrows) 257
- `\leftrightharpoons` (\leftrightharpoons) 257
- `\leftrightsquigarrow` (\leftrightsquigarrow) 257
- `\leftroot` 257
- `\leftthreetimes` (\leftthreetimes) 147
- `\left\|` 107
- `\left|` 106
- `\leq` (\leq) 53, 249
- `\leqq` (\leqq) 255
- `\leqslant` (\leqslant) 255
- `\less` 247
- `\lessapprox` (\lessapprox) 255
- `\lessdot` (\lessdot) 255
- `\lesseqgtr` (\lesseqgtr) 255
- `\lesseqqgtr` (\lesseqqgtr) 255
- `\lessgtr` (\lessgtr) 255
- `\lesssim` (\lesssim) 255
- `\lfloor` (\lfloor) 24, 251
- `\lg` 95
- `\lgroup` 155
- `\lim` 95
- `\liminf` 95
- `\limits` [`\redefine` запрещено] 77, 78, 95
- `\LimitsOnInts` 129
- `\LimitsOnNames` 129
- `\LimitsOnSums` 129
- `\limsup` 95
- `\linebreak` 35, 159, 182, 186
- `\ll` (\ll) 249
- `\llcorner` (\llcorner) 253
- `\Lleftarrow` (\Lleftarrow) 257
- `\lll` (\lll) 255
- `\lless` (\lless) 255

- `\lmoustache` 155
- `\ln` (ln) 95
- `\lnapprox` (\approx) 152, 256
- `\lneq` (\lesssim) 256
- `\lneqq` (\lesseqgtr) 256
- `\lnot` (\neg) 249
- `\lnsim` (\sim) 256
- `\loadbold` 176
- `\loadeufb` 252
- `\loadeufm` 152, 252
- `\loadeurb` 252
- `\loaeurm` 252
- `\loaeusb` 252
- `\loaeusm` 252
- `\loadmsam` 152, 252
- `\loadmsbm` 152, 178, 252
- `\log` 93-95
- log файл 33, 39
- `\longleftarrow` (\longleftarrow) 250
- `\Longleftarrow` (\Longleftarrow) 250
- `\longleftrightarrow` (\longleftrightarrow) 250
- `\Longleftrightarrow` (\Longleftrightarrow) 250
- `\longmapsto` (\longmapsto) 250
- `\longrightarrow` (\longrightarrow) 168, 204, 250
- `\Longrightarrow` (\Longrightarrow) 81, 203, 250
- `\looparrowleft` (\looparrowleft) 257
- `\looparrowright` (\looparrowright) 257
- `\lor` (\vee) 248
- `\lozenge` (\diamond) 254
- `\lq` [redefine запрещено] 17, 20, 45, 247
- `\lrcorner` (\lrcorner) 253
- `\Lsh` (\uparrow) 257
- .lst 33, 39
- `\ltimes` (\ltimes) 255
- `\lvertneqq` (\nlessgtr) 256
- `\maltese` (\blacktimes) 253
- `\mapsto` (\mapsto) 203, 250
- margin gap в `\multline` 150
- `\mathbreak` 58
- `\mathstrut` 160, 173
- `\matrix` [redefine запрещено] 106, 107, 131-132, 147-150
 - отличия между `plain` и `amstex` 245, 246
 - пробелы между столбцами 106
 - - строками 131-132
- `\max` 95
- `\measuredangle` (\sphericalangle) 255
- `\medpagebreak` 131
- `\medspace` 139, 244, 245
- `\mho` (\mho) 255
- `\mid` (\mid) 169, 249
- `\midinsert` 133
- `\min` 95
- `\minCDarrowwidth` 147
- `\mit` 245
- mm 141
- `\mod` 151
- `\models` (\models) 249
- `\Monograph` 179, 191
- `\moreref` 242
- `\mp` (\mp) 203, 248
- msam 152, 252
- msbm 126, 152, 252
- `\mu` (μ) 248
- `\multimap` (\multimap) 257
- `\multline` 104, 107, 150
 - в стиле `amsppt` 104
 - расстояния от полей 150
- `\multlinegap` 150
- `\MultLineGap` 150
- `\nabla` (∇) 203, 249
- `\narrower` 244
- `\natural` (\natural) 249
- `\ncong` (\ncong) 256
- `\ne` (\neq) 249
 - в качестве верхнего или нижнего индекса 69
- `\nearrow` (\nearrow) 250
- `\neg` (\neg) 249
- `\negmedspace` 139, 244
- `\negthickspace` 139
- `\negthinspace` 139
- `\neq` (\neq) 53, 249
- `\newline` 159
- `\newpage` 158
- `\newsymbol` 152-153, 253
- `\nexists` (\nexists) 255
- `\ngeq` (\ngtrless) 256
- `\ngeqq` (\ngtrlessgtr) 256

- `\ngeqslant` (\nlessdot) 256
- `\ngtr` (\gt) 256
- `\ni` (\ni) 249
- `\nleftarrow` (\leftarrow) 257
- `\nLeftarrow` (\Leftarrow) 257
- `\nleftrightarrow` (\leftrightarrow) 257
- `\nLeftrightarrow` (\Leftrightarrow) 257
- `\nleq` (\leq) 256
- `\nleqq` (\leqslant) 256
- `\nleqslant` (\leqdot) 152, 153, 253, 256
- `\nless` (\lessdot) 256
- `\nmid` (\nmid) 256
- `\no` 240
- `\NoBlackBoxes` 175
- `\nofrills` 124, 143, 182–183, 191, 230, 242
 - `c \title` 180
- `\nolimits` [`\redefine` запрещено] 78, 95
- `\NoLimitsOnInts` 129
- `\NoLimitsOnNames` 129
- `\NoLimitOnSums` 129
- `\nolinebreak` 159
- `\nomathbreak` 58
- `\nomultlinegap` 150
- `\nopagebreak` 158, 162
- `\nopagenumbers` 156
- `\NoPageNumbers` 156, 179
- `\NoRunningHeads` 142, 179
- `\not` [`\redefine` запрещено] 53
- `\notin` (\notin) 53, 203, 249
- `\nparallel` (\nparallel) 256
- `\nprec` (\nprec) 256
- `\npreceq` (\npreceq) 256
- `\nrightharpoonright` (\rightharpoonright) 257
- `\nRrightarrow` (\Rrightarrow) 257
- `\nshortmid` (\nshortmid) 256
- `\nshortparallel` (\nshortparallel) 256
- `\nsim` (\sim) 256
- `\nsubseteq` (\nsubseteq) 257
- `\nsubseteqq` (\nsubseteqq) 257
- `\nsucc` (\succ) 256
- `\nsucceq` (\nsucceq) 256
- `\nsupseteq` (\nsupseteq) 257
- `\nsupseteqq` (\nsupseteqq) 257
- `\ntriangleleft` (\triangleleft) 257
- `\ntrianglelefteq` (\trianglelefteq) 257
- `\ntriangleright` (\triangleright) 257
- `\ntrianglerighteq` (\trianglerighteq) 257
- `\nu` (ν) 53, 248
- `\nVdash` (\nVdash) 257
- `\nVDash` (\nVDash) 257
- `\nvdash` (\vdash) 257
- `\nvDash` (\nvDash) 257
- `\nwarrow` (\nwarrow) 250
- `o` 204
- `O` 204
- `\o` (\o) 46
- `\O` (\O) 46
- `\odot` (\odot) 248
- `\oe` (\oe) 46
- `\OE` (\OE) 46
- `\of` 147
- `\oint` (\oint) 77, 129, 250
- `\oldnos` 175, 178, 245
- `\oldstyle` 245
- `\omega` (ω) 177, 248
- `\Omega` (Ω) 177, 248
- `\ominus` (\ominus) 248
- `\openup` 185
- `\operatorname` 95, 117, 129, 155
- `\operatornamewithlimits` 95, 117, 129, 155
- `\oplus` (\oplus) 248
- `Ore`, `\Oystein` 46
- `\oslash` (\oslash) 248
- `\otimes` (\otimes) 248
- `\over` [`\redefine` запрещено] 74
- `\overarrow` 81, 126
- `\overbrace` [`\redefine` запрещено] 139, 166
 - формулы над 139
- `\overleftarrow` 81
- `\overleftrightarrow` 81
- `\overline` 81, 127, 215
- `\overrightarrow`, *см. также* `\overarrow` 81
- `\overset` 165, 166
- `\oversetbrace` 166
- `\owns` (\owns) 249
- `\P` (\P) 21–23, 27, 43, 249

- `\page` 240–241
- `\pagebreak` 158
- `\pageheight` 157, 179, 180
- `\pages` 240–241
- `\pagewidth` 157, 179, 180
- `\paper` 240–243
- `\paperinfo` 243
- `\paperinquotes` 242
- `\par` [`\redefine` запрещено] 22, 135, 195
 - в `\foldedtext` 124
 - в математическом режиме 56
- `\parallel` (\parallel) 249
- `\partial` (∂) 119, 242
- pc 141
- `\perp` (\perp) 249
- `\phantom` 169, 173
- `\phi` (ϕ) 53, 248
 - другое написание 53
- `\Phi` (Φ) 248
- `\pi` (π) 53, 248
 - другое написание 53
- `\Pi` (Π) 248
- `\pitchfork` (\pitchfork) 256
- plain Т_рX 244
- `\plainfootnote` 246
- `\plainproclaim` 246
- `\pm` (\pm) 203, 248
- `\pmatrix` 106–108
- `\pmb` 142
- `\pmod` 151, 246
- `\pod` 151
- `\Pr` 95
- `\preafill` 160
- `\preauthor` 160
- `\prec` (\prec) 249
- `\precapprox` (\approx) 256
- `\preccurlyeq` (\preccurlyeq) 256
- `\preceq` (\preceq) 249
- `\precnapprox` (\approx) 256
- `\precneqq` (\neq) 256
- `\precnsim` (\sim) 256
- `\precsim` (\sim) 256
- `\predate` 160
- `\predefine` 113
- `\prepaper` 160
- `\pretend` 146
- `\pretitle` 160
- `\prime` 68, 249
 - в нижних индексах 69
 - полужирный 177
- `\printoptions` 156
- `\proclaim` 93, 124, 125, 186, 213
 - отличие между `plain` и `amstex` 246
 - пробел над 131
- `\prod` (\prod) 250
- Programming, The Art of Computer* 237
- `\projlim` 95
- `\propto` (\propto) 249
- `\psi` (ψ) 248
- `\Psi` (Ψ) 248
- pt 34, 141
- `\publ` 241
- `\publaddr` 241
- Q, в ответ на сообщение об ошибке 164
- `\qed` 187
- Qorra, Tābit ibn 46
- `\quad` 80, 104, 136, 139
- `\quad` [`\redefine` запрещено] 80, 108, 139, 142, 148
- R, в ответ на сообщение об ошибке 164
- `\r` в `\format` 108
- `\radical` [`\redefine` запрещено] 215
- `\rangle` (\rangle) 85, 177, 216, 251
- `\rbrace` ($\}$) 51, 247, 248, 251
- `\rbrack` ($\}$) 28, 51, 83, 247, 248, 251
- `\rceil` (\lceil) 84, 251
- `\rcfrac` 152
- `\Re` (\Re) 249
- `\redefine` 113, 114
- `\ref` 240
- `\Refs` 239, 242
- `\remark` 187, 188
- `\ResetBuffer` 131
- `\restriction` (\restriction) 257
- `\rfloor` (\rfloor) 84, 251
- `\rgroup` 155

- `\rho` (ρ) 53, 248
 - другое написание 53
- `\right` [`\redefine` запрещено] 85, 129, 154
- `\right.` 86, 145
- `\right|` 106, 107
- `\right)` 83, 130
- `\right>` 85, 216
- `\right]` 83, 106
- `\rightarrow` (\rightarrow) 81, 203, 250
- `\Rightarrow` (\Rightarrow) 250
- `\rightarrowtail` (\rightarrowtail) 257
- `\rightharpoondown` (\rightharpoondown) 250
- `\rightharpoonup` (\rightharpoonup) 250
- `\rightheadtext` 143, 192
- `\rightleftarrows` (\rightleftarrows) 257
- `\rightleftharpoons` (\rightleftharpoons) 153, 250, 257
- `\rightrightarrow` (\rightrightarrow) 257
- `\rightsquigarrow` (\rightsquigarrow) 257
- `\rightthreetimes` (\rightthreetimes) 255
- `\right\|` 107
- `\risingdotseq` (\risingdotseq) 255
- `\rm` [`\redefine` запрещено] 26, 73
- `\rmoustache` 155
- `\roman` 176
- `\root` 147
- `\roster` 167, 189
 - помещение первого условия
 - внутри абзаца 167, 168
 - регулирование отступа в списке 189
- `\rq` [`\redefine` запрещено] 17, 20, 247
- `\Rrightarrow` (\Rrightarrow) 257
- `\Rsh` (\Rsh) 280
- `\rtimes` (\rtimes) 255
- Runaway argument? 74, 212
- `\runinitem` 167, 168
- `\Runinitem` 167, 168
- S, в ответ на сообщение об ошибке 164
- `\S` (§) 43, 249
- s-size 66, 70, 72
- `\sb` 64, 76, 247
- `\Sb` 76
- `\scrollmode` 164
- `\searrow` (\searrow) 250
- `\sec` 95
- `\setminus` (\setminus) 203, 248
- `\settabs` 244
- `\sharp` (\sharp) 249
- `\shave` [`\redefine` запрещено] 130
- `\shortmid` (\shortmid) 256
- `\shortparallel` (\shortparallel) 256
- `\shoveleft` 150
- `\shoveright` 150
- `\showallocations` 157
- `\showhyphens` 155, 156
- `\sideset` 167
- `\sigma` (σ) 248
- `\Sigma` (Σ) 248
- `\sim` (\sim) 249
- `\simeq` (\simeq) 249
- `\sin` 93, 95
- `\sinh` 95
- `\sl` [`\redefine` запрещено] 26, 29, 73
- `\slanted` 176
 - в верхних индексах 176
- `\slash` 159
- SMALL CAPS FONT 195
- `\smallfrown` (\smallfrown) 256
- `\smallint` (\smallint) 249
- `\smallmatrix` 147
- `\smallpagebreak` 123, 131
- `\smallsetminus` (\smallsetminus) 255
- `\smallsmile` (\smallsmile) 256
- `\smash` 168
- `\smc` 195
- `\smile` (\smile) 249
- `\snug` 202
- `\sp` 63, 247
- `\Sp` 76
- sp 141
- `\spacehdots` [`\redefine` запрещено] 150
- `\spaceinnerhdots` [`\redefine` запрещено] 150
- `\spadesuit` (\spadesuit) 249
- `\spcheck` 126
- `\specialhead` 185
 - в содержании 183–185

- `\sphat` 126
- `\sphericalangle` (\sphericalangle) 255
- `\split` 103, 105
- `\spreadlines` 132, 135, 150
- `\spreadmatrixlines` 132
- `\spvec` 126
- `\sqcap` (\sqcap) 248
- `\sqcup` (\sqcup) 248
- `\sqrt` 82, 138, 160, 215
- `\sqsubset` (\sqsubset) 256
- `\sqsubseteq` (\sqsubseteq) 249
- `\sqsupset` (\sqsupset) 256
- `\sqsupseteq` (\sqsupseteq) 249
- `\square` (\square) 254
- `ss\ss` (β) 46
- `ss-size` 66, 72
- `\ssize` [`\redefine` запрещено] 72
 - пробелы вокруг бинарного оператора в 140
- `\ssize` 73
- `\star` ($*$) 248
- `\subhead` 185
 - в содержании 184–185
- `\subclass` 181, 183
- `\subset` (\subset) 249
- `\Subset` (\Subset) 256
- `\subseteq` (\subseteq) 249
- `\subseteqq` (\subseteqq) 256
- `\subsetneq` (\subsetneq) 257
- `\subsetneqq` (\subsetneqq) 257
- `\subsubhead` 185
 - в содержании 184
- `\succ` (\succ) 249
- `\succapprox` (\succapprox) 256
- `\succcurlyeq` (\succcurlyeq) 256
- `\succeq` (\succeq) 249
- `\succnapprox` (\succnapprox) 257
- `\succneqq` (\succneqq) 257
- `\succnsim` (\succnsim) 257
- `\succsim` (\succsim) 256
- `\sum` (\sum) 75, 95, 129, 250
 - в скобках 88, 129, 130
 - ограничители вокруг 88
 - с пределами 75, 129
- `\sup` 95
- `\supset` (\supset) 249
- `\Supset` (\Supset) 256
- `\supseteq` (\supseteq) 249
- `\supseteqq` (\supseteqq) 256
- `\supsetneq` (\supsetneq) 257
- `\supsetneqq` (\supsetneqq) 257
- `\surd` (\surd) 249
- `\swarrow` (\swarrow) 250
- Świerczkowski, Stanisław 46
- `\syntax` 156
- `\t` (ˆ) акцент 45, 47, 245
- `t-size` 66, 70, 72, 91, 92, 106
- TAB19
- `\tab` 16
- Tābit ibn Qorra 113
- `\tag` [`\redefine` запрещено] 97
 - в `\align` 98, 100
 - `\alignat` 137
 - `\aligned` 100
 - `\gathered` 100
 - `\multline` 104–105
 - `\split` 103
 - с аргументом в виде литеры 154
 - формулой, заключенной в рамочку 175
 - `\gather` 101
- `\TagsAsMath` 153
- `\TagsAsText` 153
- `\TagsOnLeft` 153
- `\TagsOnRight` 153
- `\tan` 95
- `\tanh` 95
- `\tau` (τ) 248
- `\tbinom` 73
- `\tenpoint` [`\redefine` запрещено] 158
- `\tenrm` [`\redefine` запрещено] 34
- `\TeX` 24
- \TeX 10, 24
- \TeX Users Group, см. также TUG 17, 258
- TeXbook, The* 244
- `\text` 89–96, 123, 158
 - в `\cases` 109
 - верхних индексах 92
 - выключных формулах 93
- `\tfrac` 72
- `\thanks` 181, 182

- несколько 182
- `\therefore` (∴) 256
- `\therostritem` 167
- `\theta` (θ) 53, 248
 - другое написание 53
- `\Theta` (Θ) 248
- `\thetag` 154
- `\thickapprox` (\approx) 256
- `\thickfrac` 74
 - изменяющаяся толщина черты
 - дроби 140, 141
- `\thickfracwithdelims` 141
- `\thickness` 140, 141
- `\thicksim` (\sim) 255
- `\thickspace` [`\redefine` запрещено] 139, 244
- `\thinspace`
 - см. также `\`, 139
- `\tie` (\sim) 43, 159, 247
- `\tilde` (\tilde акцент в математическом режиме) 125, 126
- `\Tilde` 127
- `\tildeaccent` 247
- `\times` (\times) 171, 248
- `\title` 161, 162, 180–182, 192
- `\to` 204
 - в `\oversetbrace`,
 - `\undersetbrace` 166
 - `\sideset` 167
 - `\underset`, `\overset` 165
- `\to` (\rightarrow) 250
- `\toappear` 240
- `\toc` 183–185, 192
- `\top` (\top) 249
- `\topaligned` 135
- `\topcaption` 133, 189
- `\topfoldedtext` 124
- `\topinsert` 133
- `\topmatter` 179
 - вспомогательный материал 160
- `\TopOrBottomTagsOnSplits` 154
- `\topshave` 130
- `\topsmash` [`\redefine` запрещено] 168
- `\transl` 240
- `\translator` 181, 183
- `\triangle` (Δ) 215, 249
- `\triangledown` (∇) 254
- `\triangleleft` (\triangleleft) 248
- `\trianglelefteq` (\trianglelefteq) 256
- `\triangleq` (\triangleq) 255
- `\triangleright` (\triangleright) 248
- `\trianglerighteq` (\trianglerighteq) 256
- TROFF 237
- `\tsize` 72, 73, 86, 91
- TUG, см. также TeX Users Group 17, 258
- TUGboat* 258
- `\twoheadleftarrow` (\twoheadleftarrow) 257
- `\twoheadrightarrow` (\twoheadrightarrow) 257
- `\u` (\u акцент) 44
- `\ulcorner` (\ulcorner) 253
- `\undefine` 153
- `\underarrow` 81
- `\underbar` 29
- `\underbrace` [`\redefine` запрещено] 139, 166, 236, 237
 - формулы под 139
- Underfull `\hbox` 157
 - при использовании
 - `\showhyphens` 155–156
- Underfull `\vbox` 157
- `\underleftarrow` 81
- `\underleftrightharrow` 81
- `\underline` [`\redefine` запрещено] 81
- `\underrightarrow` [`\redefine` запрещено], см. также `\underarrow` 81
- `\underscore` 43, 247
- `\underset` 165, 166
- `\undersetbrace` 166
- `\uparrow` (\uparrow) 155, 250
- `\Uparrow` (\Uparrow) 155, 250
- `\updownarrow` (\updownarrow) 155, 250
- `\Updownarrow` (\Updownarrow) 155, 250
- `\upharpoonleft` (\upharpoonleft) 257
- `\upharpoonright` (\upharpoonright) 257
- `\uplus` (\uplus) 248
- `\uproot` 147
- `\upsilon` (υ) 140, 248
- `\Upsilon` (Υ) 248
- `\upuparrows` (\upuparrows) 257

- `\urcorner` 253
- `\UseAMSsymbols` 152, 252–254
- `\usualspace` 125
- `\v` (\sim акцент) 44
- `\varDelta` (Δ) 248
- `\varepsilon` (ε) 53, 248
- `\varGamma` (Γ) 177, 248
- `\varinjlim` 95
- `\varkappa` (\varkappa) 53, 254
- `\varLambda` (Λ) 248
- `\varliminf` 95
- `\varlimsup` 95
- `\varnothing` (\emptyset) 254
- `\varOmega` (Ω) 177, 248
- `\varphi` (φ) 53, 248
- `\varPhi` (Φ) 248
- `\varpi` (ϖ) 53, 248
- `\varPi` (Π) 248
- `\varprojlim` 95
- `\varpropto` (\propto) 256
- `\varPsi` (Ψ) 248
- `\varrho` (ϱ) 53, 248
- `\varsigma` (ς) 140, 248
- `\varSigma` (Σ) 248
- `\varsubsetneq` (\subsetneq) 257
- `\varsubsetneqq` (\subsetneqq) 257
- `\varsupsetneq` (\supsetneq) 257
- `\varsupsetneqq` (\supsetneqq) 257
- `\vartheta` (ϑ) 53, 248
- `\varTheta` (Θ) 248
- `\vartriangle` (Δ) 254
- `\vartriangleleft` (\triangleleft) 256
- `\vartriangleright` (\triangleright) 256
- `\varUpsilon` (Υ) 248
- `\varXi` (Ξ) 248
- `\vbox` [\redefine запрещено] 157
 - `Overfull` 157
 - `Underfull` 157
- `\vcorrection` 157
- `\vdash` (\vdash) 249
- `\vDash` (\vDash) 256
- `\Vdash` (\Vdash) 256
- `\vdots` (\vdots) 107
- `\vec` (\vec акцент в математическом режиме) 125, 126
- `\Vec` 127
- `\vee` (\vee) 248
- `\veebar` (\veebar) 255
- `\vert` ($|$) [\redefine запрещено] 49, 50, 51, 84, 155, 177, 203, 247, 248, 251
- `\Vert` ($\|$) [\redefine запрещено] 84, 155, 177, 203, 247, 248, 251
- `\vmatrix` 106
- `\Vmatrix` 106
- `\vol` 240
- `\vphantom` [\redefine запрещено] 174
- `\vspace` 132–136, 150
- `\Vdash` (\Vdash) 256
- `\wedge` (\wedge) 248
- `\widehat` 126, 253
- `\widestnumber` 185, 189
 - `c \key` 239
 - `c \no` 239
- `\widetilde` 126, 253
- `\wp` (\wp) 249
- `\wr` (\wr) 248
- x**, в ответ на сообщение об ошибке 38
- X**, в ответ на сообщение об ошибке 38
- `\xalignat` 137
 - `\tag's` на 137
- `\xi` (ξ) 248
- `\Xi` (Ξ) 248
- `\xxalignat` 137
- `\yen` 253
- `\yr` 239
- Yurév, Sergei 46
- `\zeta` (ζ) 140, 248

ОГЛАВЛЕНИЕ

От переводчика и редактора	5
Предисловие ко второму изданию	7
Слова признательности	8
Прочь половые признаки	9
Введение. О сложных видах набора	10
Часть 1. Закуски	15
0. Первое знакомство; ключевая глава	15
Символы, имеющиеся на клавиатуре	
1. Лингвистические уТрХи	18
Обычный текст и управляющие последовательности	
2. Тиснем любой набор	26
Замена шрифтов	
3. Ваш первый ТрХ-практикум	31
Обработка файла ТрХ'ом	
4. ТрХнический контроль ошибок	37
Сообщения об ошибках, и как на них реагировать	
5. Разделяющие пробелы и связывающие тильды	41
Тонкости, касающиеся пробелов и разрыва строк	
6. С каждым зна́ком знако́м	43
Особые символы и акценты	
Часть 2. Дежурные блюда	49
7. МаТрМатический экстаз	49
Математические формулы в тексте	
8. Скверный разрыв? Сделайте “выключку”	58
Выключные формулы	
9. 2 ^й уровень сложности	63
Верхние и нижние индексы	
10. наших проблем громадьё	70
Дроби, биномиальные коэффициенты и т. п.	

11.	Большие дары \TeX 'а	75
	Σ , \int и другие “большие операторы”	
12.	Создание своего пространства	79
	Корректировка пробелов в математических формулах	
13.	Длина меняется сама собой	81
	Ограничители и другие символы, меняющие размеры	
14.	Прямым текстом	89
	Прямой шрифт в формулах	
15.	Держите равенство	97
	Системы уравнений и многострочные выключные формулы	
16.	Столько всяких удобств	102
	Разрыв слишком длинных формул	
17.	Затруднительное положение	106
	Матрицы	
	Часть 3. Напитки и приправы	110
18.	Практическое самоуправление	110
	Определение новых управляющих последовательностей	
19.	\TeX нические приемы	123
	\TeX нический толковый словарь	
	Приложения	179
A.	Стиль AMS для журнальных статей	179
B.	Ответы ко всем упражнениям	194
C.	Библиография	239
D.	Сравнение с ‘plain’ \TeX 'ом	244
E.	Отсутствующие клавиши	247
F.	Специальные знаки	248
G.	Дополнительные шрифты	252
H.	TUG — группа пользователей \TeX 'а	258
I.	На помощь!	259
	Предметный указатель	260