

L^AT_EX, GNU/Linux и русский стиль.

© Е.М. Балдин*



Эта статья была опубликована в ноябрьском номере русскоязычного журнала Linux Format (<http://www.linuxformat.ru>) за 2006 год. Статья размещена с разрешения редакции журнала на сайте <http://www.inp.nsk.su/~baldin/> и до апреля месяца все вопросы с размещением статьи в других местах следует решать с редакцией Linux Format. Затем все права на текст возвращаются ко мне.

Текст, представленный здесь, не является точной копией статьи в журнале. Текущий текст в отличии от журнального варианта корректор не просматривал. Все вопросы по содержанию, а так же замечания и предложения следует задавать мне по электронной почте <mailto:E.M.Baldin@inp.nsk.su>.

Текст на текущий момент является просто *текстом*, а не книгой. Поэтому результирующая доводка в целях улучшения восприятия текста не проводилась.

*e-mail: E.M.Baldin@inp.nsk.su

Эмблемы T_EX и METAfont, созданные Дуайном Бибби, взяты со странички Д.Э. Кнута. Цветной пингвин взят из пакета ps2pdf от Ральфа Найпрашека (Rolf Niepraschk)

Оглавление

3. Набор математики	1
3.1. Набор формул	2
3.2. Кириллица в формулах	3
3.3. Школьная математика	4
3.3.1. Индексы	4
3.3.2. Математические символы	5
3.3.3. Дроби	6
3.3.4. Корни	7
3.3.5. Квадратное уравнение	7
3.3.6. Функции	8
3.3.7. Производная и интеграл	8
3.3.8. Скобки	9
3.4. Перенос формул	10
3.5. Заключение	11

Набор математики

Полиграфисты относят математические работы к каторжным. . .

Д.Э. Кнут. Математическая типография.

Иногда от незнакомых с $\text{\TeX}$ технологиями людей приходится слышать, что $\text{\LaTeX}$ годится *только* для набора математики. При знакомстве же с истинными $\text{\TeX}$ технологиями возникает понимание, что $\text{\LaTeX}$ настолько хорош, что с его помощью можно набирать *даже* математику.

Набор математики всегда считалась вершиной типографского искусства. Дело в том, что формулы для концентрации информации и дополнительной выразительности в отличие от обычного текста являются многоуровневыми. Д.Э. Кнут к своей программе компьютерной типографии создал язык для описания формул. После короткого периода обучения пользователь в состоянии читать и набирать формулы на этом языке практически любой сложности.

$\text{\LaTeX}$ не единственная программная среда, использующая $\text{\TeX}$ -нотацию. Эта же нотация рекомендуется при наборе всех сколько-нибудь сложных формул на страницах Википедии (<http://ru.wikipedia.org> статья «Википедия:Формулы»).

Становлению $\text{\TeX}$ как стандарта для набора формул в значительной степени способствовало Американское математическое сообщество (The American Mathematical Society — AMS), которое в начале восьмидесятых годов прошлого столетия субсидировало разработку расширения $\text{\TeX}$ известного как $\mathcal{A}\mathcal{M}\mathcal{S}\text{-}\text{\TeX}$. В 1987 году наработки $\mathcal{A}\mathcal{M}\mathcal{S}\text{-}\text{\TeX}$ были добавлены в $\text{\LaTeX}$ в виде пакета **amsmath**. Вместе с **amsmath** в $\text{\LaTeX}$ было добавлено множество улучшений, позволяющих набирать действительно изощрённую математику. Поэтому при использовании в тексте математики в шапке документа следует в обязательном порядке загружать пакет **amsmath**:

```
\usepackage{amsmath}
```

В дальнейшем предполагается, что этот пакет уже загружен.

Полностью описать *все* команды языка описания формул в рамках короткой статьи нереально, так как математика, как и способы её описания, безгранична. Поэтому основное внимание будет уделено базовым правилам и русскому стилю в формулах. В любой сколько-нибудь большой книге по ЛАТЭХ будет полный список всех команд. Если серьёзно работать с математикой, то подобная книжка в любом случае понадобится.

3.1. Набор формул

При формировании текста формулы подразделяются на *строчные* и *выносные*. Строчные формулы набираются внутри абзаца вместе текстом. По описанию формулы ЛАТЭХ создаёт бокс, который обрабатывается наравне с обычными текстовыми боксами. Как правило, строковые формулы это небольшие вставки, вроде $E = mc^2$. Выносные или *выключенные* формулы выводятся за пределы абзаца.

Строчная формула в тексте ограничивается¹ с помощью символа доллара \$«формула»\$ или с помощью команд-скобок \ («формула» \). При наборе предпочтительно использовать второй вариант оформления, так как он позволяет легко определить где начинается, а где кончается формула. «Долларовое» (\$) окружение лучше тем, что оно чуть-чуть короче, кроме этого команда \$ *крепкая*² в отличии от команд-скобок.

Однострочные выносные формулы формируются с помощью окружения `equation`. Так как в этом случае формула вынесена за пределы абзаца, то её можно пронумеровать. Например:

```
\begin{equation}
\label{eq:math:ex1}
\int\limits_{-\infty}^{\infty}
e^{-x^2/2}dx=\sqrt{2\pi}
\end{equation}
```

$$\int_{-\infty}^{\infty} e^{-x^2/2} dx = \sqrt{2\pi} \quad (3.1)$$

Нумерация формул удобна для того, чтобы позже в тексте на неё можно было легко сослаться с помощью команды `\eqref{eq:math:1}`³. Если же формул немного и не хочется никакой нумерации, то можно воспользоваться окружением `equation*`⁴.

¹Есть более формальное оформление строчной формулы как окружения: `\begin{math}` «формула» `\end{math}`. Но в силу понятных причин никто подобное описание не использует в пользу кратких обозначений.

²Когда начинаешь изучать команды ЛАТЭХ, то довольно быстро сталкиваешься с понятиями «хрупкости»/«крепкости». Крепкие команды в отличии от хрупких можно использовать в качестве аргументов других команд. С другой стороны хрупкие команды тоже можно использовать как параметры, защитив их с помощью команды `\protect`. Эти понятия в большинстве своём пережитки прошлого и их постепенно изживают, но пока их следует иметь в виду.

³Метка выставляется с помощью команды `\label`.

⁴К `equation` добавляется звёздочка. Подобный приём в создании команд применяется достаточно часто. Команда со * обычно не нумеруется и не отображается ни в каких автоматически составляемых списках.

При создании выключенной формулы размер шрифта для улучшения читаемости немного увеличивается. ЛАТЭХ имеет несколько стилей для оформления математических формул. При желании можно выбрать необходимый стиль в ручную:

`\displaystyle` — стиль, используемый для выносных формул,

`\textstyle` — стиль строчных формул,

`\scriptstyle` — в этом стиле набираются индексы,

`\scriptscriptstyle` — индексы второго уровня.

С помощью этих команд можно увеличить размер шрифта для формул внутри абзаца, или заставить индексы выглядеть как базовые символы. Для примера сравните:

```
\begin{equation*}
\frac{1}{1+
\frac{1}{1+
\frac{1}{1+
\frac{1}{2}}}}
\end{equation*}
```

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

```
\begin{equation*}
\frac{1}{\displaystyle 1+
\frac{1}{\displaystyle 1+
\frac{1}{\displaystyle 1+
\frac{\displaystyle 1}{
\displaystyle 2}}}}
\end{equation*}
```

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

Пробелы в формулах отмечают только конец команды, а сами по себе смысла не имеют — ЛАТЭХ, как правило, гораздо лучше знает как сформировать результат.

3.2. Кириллица в формулах

Всё дело в имеющихся шрифтах — они красивые, разнообразные, но в большинстве своём англоязычные. В настоящее время кириллические математические шрифты в «дикий природе» отсутствуют, поэтому приходится пользоваться их текстовыми версиями.

Стиль **mathtext** (пакет **t2**), позволяет использовать кириллицу в формулах без дополнительных ухищрений. Стиль может быть подключён с опцией *warn* — в этом случае он сообщает обо всех случаях использования кириллических букв в формулах. **mathtext** следует загружать до **babel** и/или **fontenc**.

```
\usepackage [ warn ] { mathtext }
```

```
\[
v_{cp}=\frac{S_{\text{конец}}-S_{\text{начало}}}
{\delta t}
\]
```

$$v_{\text{ср}} = \frac{S_{\text{конец}} - S_{\text{начало}}}{\delta t}$$

Здесь для создание выключенной формулы используется команда `\[«формула»\]` — краткий аналог окружения `equation*`. В отличие от латиницы русские буквы в формулах печатаются прямым шрифтом — это было сделано специально. Чтобы изменить это умолчание в преамбуле следует добавить команду для переопределения шрифта:

```
\DeclareSymbolFont{T2Aletters}{T2A}{cmr}{m}{it}
```

Стиль **amstext** (загружается автоматически при загрузке **amsmath**) определяет команду `\text`, которая позволяет вставлять в формулу обычный текст. Текст может быть и русским:

```
\[v_{cp}=
\frac{\text{конец пути}-
\text{начало пути}}
{\text{время в пути}}\]
```

$$v_{\text{ср}} = \frac{\text{конец пути} - \text{начало пути}}{\text{время в пути}}$$

Преимущество такого подхода заключается в том, что внутри команды `\text` пробелы воспринимаются как нормальные символы и слова не сливаются. Использование `\text` предпочтительно и для целей переносимости.

3.3. Школьная математика

Математика в школе — это явление, через которое проходит каждый. Именно поэтому фактически любой вменяемый россиянин умеет обращаться с дробями, знает теорему Пифагора, с лёгкостью решает квадратные уравнения и что-то слышал про интеграл и производную. Разберёмся с этим поподробнее.

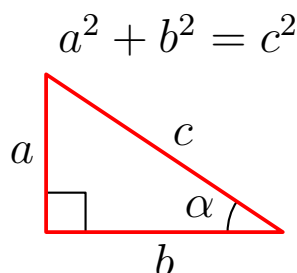
3.3.1. Индексы

Букв в латинском алфавите не так уж и много, а научных понятий без числа. Один из способов отличать обозначения друг от друга, это индексы, как верхние, так и нижние:

```
\[A_{\text{нижний индекс}}\quad
B^{\text{верхний индекс}}\quad
C_n^k\]
```

$$A_{\text{нижний индекс}} \quad B^{\text{верхний индекс}} \quad C_n^k$$

Обратите внимание, что если в индексе ровно один знак, то фигурные скобки вокруг него можно и нужно опустить. Теперь можем записать теорему Пифагора:



3.3.2. Математические символы

Кроме символов латиницы и кириллицы математики используют множество самых разнообразных значков. Да и латиница не так уж и проста. Если воспользоваться пакетом **amsfonts** то она может стать:

```
\begin{itemize}
\item \(\ABCD\) "--- обычной,
\item \(\mathbf{ABCD}\) "--- жирной,
\item \(\mathbb{ABCD}\) "--- ажурной,
\item \(\mathcal{ABCD}\) "---
прописной.
\end{itemize}
```

- $ABCD$ — обычной,
- $\mathbf{ABCD}$ — жирной,
- $\mathbb{ABCD}$ — ажурной,
- $\mathcal{ABCD}$ — прописной.

Это далеко не все возможные шрифтовые стили которые можно применять в математической моде. Но лучше особо не перегружать формулы всякой «готикой» (например, команда `\mathfrak`).

Не единой латиницей жив математик. Традиционно везде, где только можно, используются греческие буквы. В $\text{\LaTeX}$ присутствует полный набор и за исклю-

Греческие символы									
α	<code>\alpha</code>	β	<code>\beta</code>	γ	<code>\gamma</code>	δ	<code>\delta</code>	ε	<code>\epsilon</code>
ζ	<code>\zeta</code>	η	<code>\eta</code>	θ	<code>\theta</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>	$\omicron$	<code>\omicron</code>
π	<code>\pi</code>	ρ	<code>\rho</code>	σ	<code>\sigma</code>	τ	<code>\tau</code>	υ	<code>\upsilon</code>
φ	<code>\phi</code>	χ	<code>\chi</code>	ψ	<code>\psi</code>	ω	<code>\omega</code>	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>

чением трёх букв начертание вполне привычное. Для исправления непривычных начертаний эти буквы были переопределены с помощью пакета **amssymb**:

```
% Переопределение \kappa, \epsilon и \phi на русский лад
\renewcommand{\kappa}{\varkappa}
\renewcommand{\epsilon}{\varepsilon}
\renewcommand{\phi}{\varphi}
```

Спецсимволов в $\text{\LaTeX}$ великое множество. В стандартной поставке $\text{\TeX}$ Live идёт «Всеобъемлющий список символов $\text{\LaTeX}$ » (The Comprehensive $\text{\LaTeX}$ Symbols List — файл `symbols-a4.pdf`) в котором перечислено 3300 распространённых символа, используемых пользователями $\text{\LaTeX}$. Почти наверняка любой операнд, который вам нужен, там уже есть. Ниже будут перечислены только та часть символов, которая с моей точки зрения может пригодиться в наборе школьной математики. Пакет **amssymb** для использования обязателен.

«Школьные» символы									
$\hat{a}$	<code>\hat{a}</code>	$\bar{a}$	<code>\bar{a}</code>	$\vec{a}$	<code>\vec{a}</code>	$\dot{a}$	<code>\dot{a}</code>	$\tilde{a}$	<code>\tilde{a}</code>
$\pm$	<code>\pm</code>	$\mp$	<code>\mp</code>	$\times$	<code>\times</code>	$\cdot$	<code>\cdot</code>	$\div$	<code>\div</code>
$\vee$	<code>\vee</code>	$\wedge$	<code>\wedge</code>	$\neg$	<code>\neg</code>	$\forall$	<code>\forall</code>	$\exists$	<code>\exists</code>
$\leq$	<code>\leq</code>	$\geq$	<code>\geq</code>	$\ll$	<code>\ll</code>	$\gg$	<code>\gg</code>	$\neq$	<code>\neq</code>
$\equiv$	<code>\equiv</code>	$\sim$	<code>\sim</code>	$\simeq$	<code>\simeq</code>	$\approx$	<code>\approx</code>	$\propto$	<code>\propto</code>
$\parallel$	<code>\parallel</code>	$\perp$	<code>\perp</code>	$\angle$	<code>\angle</code>	$\triangle$	<code>\triangle</code>	$\sphericalangle$	<code>\sphericalangle</code>
∞	<code>\infty</code>	ℓ	<code>\ell</code>	$\sum$	<code>\sum</code>	$\prod$	<code>\prod</code>	$\emptyset$	<code>\emptyset</code>

Для соответствия русским традициям два символа были переопределены:

```
% Переопределение \le и \ge на русский лад
\renewcommand{\le}{\leqslant}
\renewcommand{\ge}{\geqslant}
```

3.3.3. Дроби

Дроби формируются с помощью команды `\frac`⁵:

```
\[
  дробь=\frac{числитель}{знаменатель}
\]
```

$$\text{дробь} = \frac{\text{числитель}}{\text{знаменатель}}$$

Как и практически вся математика в $\text{\LaTeX}$ дробь записывается как читается само выражение.

⁵От слова fraction — дробь.

3.3.4. Корни

Для рисования знака корня используется команда

$$\backslash \text{sqrt} [\text{«степень»}] \{ \text{«подкоренное выражение»} \}$$

Степень можно упустить. В этом случае рисуется обычный квадратный корень.

```
\[
\overline{
\underline{\Large
\sqrt[3]{a}+\sqrt[2]{b}+\sqrt[99]{g}
}
}
\]
```

$$\overline{\sqrt[3]{a} + \sqrt[2]{b} + \sqrt[99]{g}}$$

Обратите внимание, что знак корня размещается в соответствии с размерами подкоренного выражения. Если в выражении присутствует только один корень, то это самое разумное поведение, но в случае нескольких корней, как вышеприведённом примере, то необходимо выравнивание.

Для выравнивания по высоте используется команда `\mathstrut`⁶. В результате её применения вставляется невидимый символ нулевой толщины высотой в точности равной высоте круглой скобки:

```
\[ \Large
\sqrt[3]{\mathstrut a}+
\sqrt[2]{\mathstrut b}+
\sqrt[99]{\mathstrut g}
\]
```

$$\sqrt[3]{a} + \sqrt[2]{b} + \sqrt[99]{g}$$

3.3.5. Квадратное уравнение

И наконец вершина школьной математики — это решение квадратного уравнения $ax^2 + bx + c = 0$:

```
\[
x_{1,2}=\frac{-b\pm\sqrt{b^2-4ac}}{2a}
\]
```

$$x_{1,2} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

Теперь можно смело писать методички по школьной математике ☺.

⁶От английского strut — подпорка или страта.

3.3.6. Функции

Все символы в математической моде печатаются курсивом, поэтому названия функций для выделения печатаются прямым шрифтом. Кроме смены шрифта функции с обеих сторон должны правильно «отбиваться» пробелами, иначе будет некрасиво. При загрузке русского языка с помощью пакета **babel** кроме стандартных имён функций доопределяется несколько сокращений применяемых в русскоязычной литературе. Среди часто употребляемых функций можно упомянуть: `cos`, `arccos`, `sin`, `arcsin`, `tg`, `arctg`, `ctg`, `arcctg`, `sh`, `ch`, `th`, `cth`, `exp`, `ln`, `log`, `lim`, `min` и `max`. В математической моде эти функции можно использовать в качестве команд:

```
\begin{equation*}
\begin{split}
&\&\log_2 10=\ln 10/\ln 2\simeq 3.32 \\
&\&\lim_{x\rightarrow 0}\frac{\sin x}{x}=1 \\
&\&(a+b)^n=\sum_{k=1}^n C_n^k a^k b^{n-k}
\end{split}
\end{equation*}
```

$$\log_2 10 = \ln 10 / \ln 2 \simeq 3.32$$

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

$$(a+b)^n = \sum_{k=1}^n C_n^k a^k b^{n-k}$$

Обратите внимание на обработку индексов для функции `log` (логарифм) и `lim` (предел). Для доопределения новых функций правильнее всего воспользоваться в преамбуле командой `\DeclareMathOperator`:

```
% В преамбуле — определение новых функций
\DeclareMathOperator{\log-like}{log-like}
\DeclareMathOperator*{\lim-like}{lim-like}
```

В зависимости от варианта команды индексы отображаются как для логарифма (команда без звёздочки) или как для предела (команда со звёздочкой).

3.3.7. Производная и интеграл

В старших классах в конце обучения чуть-чуть касаются понятий интегрирования и дифференцирования. Возможно для того, чтобы правильно подсчитать сдачу в магазине, эти знания не являются необходимыми. Но для изучения физики и, как следствие, химии и биологии без интегралов никак — поверьте мне на слово.

Производная, обычно отмечается штрихом. В физике производная по времени выделяется точкой, для того чтобы отличать её от производной по координате. Можно честно написать `\frac{d F(x)}{dx}`. Для частной производной вместо буквы *d* используется спецсимвол `\partial`:

```
\[ f'\quad f''\quad
\dot{f}\quad \ddot{f}\quad
\frac{d f}{d x}\quad
\frac{\partial f}{\partial x} \]
```

$$f' \quad f'' \quad \dot{f} \quad \ddot{f} \quad \frac{df}{dx} \quad \frac{\partial f}{\partial x}$$

Производная есть обратная функция от интегрирования:

```
\[
\frac{d}{dx}\int F(x) dx=F(x)
\]
```

$$\frac{d}{dx} \int F(x) dx = F(x)$$

Приглядевшись к имеющемуся здесь примеру, можно отметить, что в отличии от русских математических традиций представленный здесь интеграл не прямой, а наклонный. Это можно исправить, напимер, загрузив пакет **wasysym** с опцией **integrals**. К сожалению получающиеся интегралы «не смотрятся». Поэтому пока лучше использовать начертания по умолчанию в надежде, что в будущем ситуация изменится к лучшему.

Неопределённый интеграл хорошо, но с определённым тоже надо работать. Качественное оформление пределов интегрирования важно для восприятия формулы.

```
\[
\int_0^{\infty}\quad
\int\limits_0^{\infty}\quad
\sum_{i=1}^n\quad
\sum\nolimits_{i=1}^n\quad
\]
```

$$\int_0^{\infty} \quad \int\limits_0^{\infty} \quad \sum_{i=1}^n \quad \sum_{i=1}^n$$

По умолчанию пределы размещаются справа от интеграла. Ситуацию можно поправить с помощью команды **\limits**. Команда **\nolimits** делает всё ровно наоборот.

3.3.8. Скобки

Для визуальной группировки символов внутри формулы скобки вещь незаменимая. Особенно здорово, если скобки автоматически подбирают свой размер под выражение, которое они окружают. Парные команды **\left** и **\right** включают режим подобной подстройки:

```

\[\left(
  \left[
    \left\langle
      \left\{
        \left\uparrow
          \left\lceil
            \left|
              \left\lfloor
                \text{что-то}^{10}
                \right\rfloor^9
                \right|^{8}
                \downarrow^7
              \right\}^6
              \right\rangle^5
            \right]^4
          \right)^3
        \right]^2
      \right)
    \right]^5
  \right\rangle^4
\right]^3
\right)^2\]

```

$$\left(\left[\left\langle\left\{\left\uparrow\left\lceil\left|\left\lfloor\text{что-то}^{10}\right\rfloor^9\right|^{8}\downarrow^7\right\}^6\right\rangle^5\right]^4\right)^3\right]^2\right)^5\right\rangle^4\right]^3\right)^2\]$$

Иногда хочется размер выставить в ручную, тогда перед скобкой можно выставить одну из следующих команд:

```

\[( \big( \Big( \bigg( \Bigg( \]

```

$$(((\left($$

Эстеты в зависимости от ситуации в конце команды могут добавить модификатор позиционирования разделителя как левого — `l` (отбивка как для `\left`), правого — `r` (отбивка как для `\right`) и среднего — `m`.

3.4. Перенос формул

В русскоязычной литературе принято, что при переносе строчной формулы на другую строку знак, по которому формула разрывается дублируется на следующей строке. Например так:

```

a + b =
= c

```

По умолчанию этого не происходит. Проще всего решить эту проблему с помощью следующего макроса⁷, который необходимо определить в преамбуле:

```

% перенос формул в тексте
\newcommand*{\hm}[1]{#1\nobreak\discretionary}{%
  {\hbox{$\mathsurround=0pt #1$}}{}}

```

⁷Рецепт от Евгения Миньковского из fido7.ru.tex.

Здесь определена команда `\hm`, которую следует добавлять в местах потенциального переноса формулы, примерно, так: `\(a + b \hm{=} c\)`. Сделать это можно во время окончательной доводке текста. В любом случае для полировки рукописи ручная работа необходима.

Разрыв математических формул при переносе предпочтителен на знаках отношения ($=$, $>$, $<$, $\leq$, $\geq$, $\neq$, $\simeq$); во вторую очередь на отточии, знаках сложения и вычитания; в третью — на знаке умножения в виде крестика. Не рекомендуется разбивать формулу на знаке деления и на каких-либо других знаках, кроме упомянутых выше.

3.5. Заключение

Изложенных правил и приёмов вполне хватит для набора в рамках школьной математики. Для более изощрённых формул требуются более продвинутые приёмы и конструкции. Всё это будет, но чуть попозже.

Врезка: Вики

«Движок» который использует Википедия для отображения формул называется WikiTeX. Основной сайт проекта, естественно, представляет из себя вики по адресу <http://wikisophia.org/>. Используя это программное обеспечение в связки с L^AT_EX, можно не только сносно отображать математические формулы на WWW без особых ухищрений, но и отрисовывать шахматные партии, химические формулы, фейнмановские диаграммы, нотные записи и многое другое.

T_EX сразу разрабатывался как программа, которая может формировать изображения для разных устройств, даже для тех, о которых на момент создания этого текстового процессора профессору Д.Э.Кнуту ничего известно не было. Поэтому T_EX обретается в самых неожиданных местах.

Classes
[\[edit\]](#)

The following classes have been implemented:

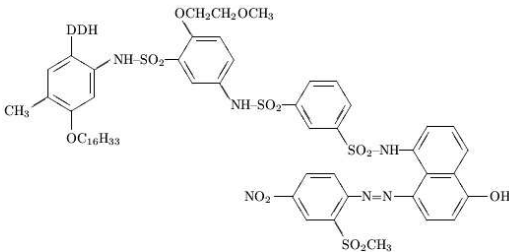
Amsmath
[\[edit\]](#)

Plenary [AMS-L^AT_EX](#), including commutative diagrams ([doc](#) | [template](#)).

$$\forall s, t \in \omega^* [s \geq t \Leftrightarrow s \text{ is an initial sequence of } t]$$

Chem
[\[edit\]](#)

[XyM^TE_X](#) is an intuitive chemistry package by Shinsaku Fujita ([doc](#) | [template](#)).



Chess
[\[edit\]](#)

[Skak](#) by Torben Hoffmann supports SAN and FEN notation ([doc](#) | [doc](#) | [doc](#) | [template](#)).



Рис. 3.1. WikiTeX за работой.

Сделайте так, чтобы Ваш форум или вики заговорил на языке L^AT_EX.