JLM—Jörg's LaTeX Mode

Jörg Sommer <joerg@alea.gnuu.de>

5. August 2007

JLM is an advanced LaTeX mode for Jed. Its aim is to help the user by taking over annoying and stupid things they need to be done, like add a `\usepackage` when you add a command or environment from this package or put dollar signs around mathematical commands and move the cursor to an appropriate point.

JLM does not think for you. If you want a `tabular`, but insert an `itemize`, JLM doesn't prevent this. So, you should be familiar with LaTeX and know what environments and commands are and where to place them in a document—JLM helps writing them.

Inhaltsverzeichnis

1 Installation **4**

1.1 . . . on Unix-like systems . . . 4

1.2 . . . on Windows . . . 4

2 TEX environments—env_insert() **6**

3 `array` like environments **7**

4 TEX commands—cmd_insert() **8**

5 Math stuff **9**

5.1 In text formulas $. . . $ . . . 9

5.2 Active characters after math commands . . . 9

6 Folding **10**

7 Babel **12**

7.1 Hyphenation . . . 12

8 Keys **14**

8.1 Newline with completion . . . 14

8.1.1 environments—`^Ce` . . . 14

8.1.2 commands—`^Cd` . . . 15

8.2 Hot keys for TEX commands . . . 15

8.2.1 sectioning commands—`^Cs` . . . 15

8.2.2 font commands—`^Cf` . . . 15

8.2.3 math commands—`^Cm` and `^Cn` . . . 16

8.2.4 links—`^Cl` . . . 16

8.2.5 folding—`^Co` and `return` . . . 17

8.3 Mathematical arrows . . . 17

8.4 Mathematical sub- and superscripts . . . 18

1 Installation

1.1 . . . on Unix-like systems

Obviously, you need Jed. You can download the current version from http://www.jedsoft.org/jed/download.html, but it is also part of many distributions. You need at least version 0.99.18 with SLang 2 support (run `jed --version` to check it).

0. Download the archive `jjm-rXYZ.tgz`, where XYZ is the version usually the highest available, from http://www.minet.uni-jena.de/~joergs/.

1. Create a directory `.jed/` in your home directory, if it doesn't exist, and save there the files from the archive. Save in the same directory the file `x-keydefs.sl` from http://jedmodes.sf.net/mode/x-keydefs/—use http://jedmodes.cvs.sourceforge.net/*checkout*/jedmodes/mode/x-keydefs/x-keydefs.sl?view=checkout&revision=HEAD.

2. Depending on your system, Jed loads the file `~/.jedrc` or `~/.jed/jed.rc` (on Debian) as your configuration file. You can copy the whole file `.jed/doc/latex-jed.rc` to this location or copy only parts from `latex-jed.rc` into your `jed.rc`.

   Important are only these lines to tell Jed where he can find JLM and that he should use it for files with the extention `.tex` or `.latex`.

   ```
   Jed_Home_Directory = path_concat(getenv("HOME"), ".jed");
   set_jed_library_path(Jed_Home_Directory + "," + get_jed_library_path());
   Jed_Highlight_Cache_Dir = path_concat(Jed_Home_Directory, "dfa");

   add_mode_for_extension("latex", "tex");
   add_mode_for_extension("latex", "latex");
   ```

   (On Debian the first three lines aren't needed.)

Now you can open a LaTeX file and start using JLM. Have fun.

1.2 . . . on Windows

Obviously, you need Jed. You can download it from http://www.paneura.com/~dino/wjed.html.

0. Download the archive `jjm-rXYZ.tgz`, where XYZ is the version usually the highest available, from http://www.minet.uni-jena.de/~joergs/.

1. Create a directory `C:\DokumenteundEinstellungen\USERNAME\Anwendungsdaten\jed` help: Where are the spaces in this URL? in your home directory, if it doesn't exist, and save there the files from the archive. Save in the same directory the file `x-keydefs.sl` from http://jedmodes.sf.net/mode/x-keydefs/—use http://jedmodes.cvs.sourceforge.net/*checkout*/jedmodes/mode/x-keydefs/x-keydefssl?view=checkout&revision=HEAD.

2. Jed loads the file `C:\DokumenteundEinstellungen\USERNAME\jed.rc`. help: Where are the spaces in this URL? You can copy the whole file `doc/latex-jed.rc` from the archive to this location or copy only parts from `latex-jed.rc` into your `jed.rc`.

   Important are only these lines to tell Jed where he can find JLM and that he should use it for files with the extention `.tex` or `.latex`.

   ```
   Jed_Home_Directory = path_concat(getenv("APPDATA"), "jed");
   set_jed_library_path(Jed_Home_Directory + "," + get_jed_library_path());
   Jed_Highlight_Cache_Dir = path_concat(Jed_Home_Directory, "dfa");

   add_mode_for_extension("latex", "tex");
   add_mode_for_extension("latex", "latex");
   ```

Now you can open a LaTeX file and start using JLM. Have fun.

2 TEX environments—env_insert()

3 `array` like environments

Environments they are organised in columns `table`, `array` or `cases`.

4 TEX commands—cmd_insert()

The function cmd_insert() is an important function of the mode. It is used for inserting TEX commands like \LaTeX or \frac. With a database (created with cmd_add()) it decides whether a command is a math command and how much mandatory arguments it has. If a command must be in math mode, e. g. \frac, it inserts dollar signs $ around the command.

The prefix of the command (prefix_argument()) determines the number of optional arguments, e. g. the exponent of \sqrt.

Many commands are predefined in latex_cmds.sl.

todo: packages needed by the command are inserted

5 Math stuff

5.1 In text formulas $...$

If a command is inserted via cmd_insert() and it is known that this is a math command, it is automaticly placed inside $...$. See chapter 4.

If you insert a dollar sign manually and a backslash is before the editing point or the dollar sign is a begin of an in text formula, nothing happens. If the dollar sign is the end of an in text formula, something more happens:

- If the variable BLINK is non-zero the editing point jumps to the begin of the text formula for a short period or until the next key is pressed.
- All open brackets [, { and (in the formula are closed before the dollar sign is inserted or all closing brackets in front of the editing point matching an open bracket are skipped before the dollar sign is inserted. A dollar sign at the point where it should be inserted is skipped and nothing is inserted.

  Some examples:

  - `$\frac{1}{2}•` → `$\frac{1}{2}$•`
  - `$\frac{1}{2•` → `$\frac{1}{2}$•`
  - `$(\{\frac{1}{2•` → `$(\{\frac{1}{2}\})$•`
  - `$\frac{1}{2•}$` → `$\frac{1}{2}$•`

5.2 Active characters after math commands

Some commands defined in `latex_cmds.sl` run a hook after insertion. If you press one of the characters "=-+<>" after you inserted one these commands the editing point is moved before the dollar sign and the character is inserted there. All greek letters have this hook.

Example: You want to write "α is the upper angle in a triangle and $\alpha + \beta = \gamma$." Simply type it. Use the math key for the greek characters, e. g. `^Cma` for `$\alpha$`. After the first `\alpha` the editing point is after the dollar sign (`$\alpha$•`). There you can type help: weiterschreiben. After the next `\alpha` you have the same situation. Press the plus sign and you get this: `$\alpha+•$`. Type on til the `\gamma`. There you must use the right key or use the dollar sign magic section 5.1.

6 Folding

JLM supports folding.

```
|\documentclass{...}
|  |\usepackage{...}
|\begin{document}
|  |...
|  |\begin{...}
|  |  |...
|  |  |\begin{...}
|  |  |  |...
|  |  |\end{...}
|  |  |...
|  |\end{...}
|  |...
|  |\chapter{...}
|  |  |...
|  |  |\section{...}
|  |  |  |...
|  |  |  |\subsection{...}
|  |  |  |  |...
|  |  |  |  |\subsubsection{...}
|  |  |  |  |  |...
|  |  |  |  |\subsubsection{...}
|  |  |  |  |  |...
|  |  |  |\subsection{...}
|  |  |  |  |...
|  |\chapter{...}
|\end{document}
|0  |1 |2 |3 |4 |5
```

7 Babel

7.1 Hyphenation

Compound words are common in german. But compounding words with a dash (`-`) has some drawbacks: You loose automatic hyphenation or get bad hyphen points; see Figure 7.1 The package babel provides commands to circumvent these problems, but they are difficult to type in—they start with a `"` which is bound to quotation mark function todo: add link to this section—and you must remember that you have to use them.

JLM takes care of this and replaces the dash by an suitable command. A dash followed by a dollar sign or alphabetical character starts the functions and it watches out for alphabetical characters, a slash or a closing brace. So you can cancel the operation by moving the editing point or enter a number.

The functions counts the characters on the left side and on the right side. If they are more than an internal set threshold the dash is replaced. If on the left side are fewer characters than the threshold the dash is replaced by `"~` otherwise by `"=`. If a shash or an closing brace follows the dash, it is replaced by `"~` and `""` is written after the slash or brace, respectively.

Some examples:

- `primitiv-rekursiv` becomes `primitiv"=rekursiv`
- `t-produktiv` becomes `t"~produktiv`
- α`-Teilchen` becomes α`"~Teilchen`
- `Ein-/Ausgabe` becomes `Ein"~/""Ausgabe`
- `(Haupt-)Aufgabe` becomes `(Haupt"~)""Aufgabe`

bad hyphenation	good hyphenation
HI- Virus	 HI-Virus
bergauf und - ab	bergauf und -ab
primitiv- rekursiv	primitiv-rekur- siv
 Motorrad-Handbuch	Motor- rad-Handbuch
Ein- /Ausgabe	Ein-/ Ausgabe
(Haupt-)Stromkreis	(Haupt-) Stromkreis

Figure 7.1: Examples of good and bad hyphenation of compound words in german caused by using a simple dash

8 Keys

8.1 Newline with completion

Some environments are structured internally with special commands like `\item` in `enumerate` or `\\` in `tabular`. newline_with_completion() (bound to `^C-Return` and `Shift-Return`) inserts a text before and after the linebreak.

Like the dollar sign (section 5.1) it closes open brackets or skipps closing brackets before it inserts.

Examples:

```
\begin{itemize}
 \item a aa•
 \item $aaaa•
 \item \texttt{aa aa•}
 \item \texttt{$aaa•}
 \item (•)
 \item $(\texttt{aaaa•})
 \item aaa $aa$ •
\end{itemize}

\begin{gather*}
  \begin{cases}
    aaa& aaa•
  \end{cases}\\
  aaa aa (a_{•})
\end{gather*}
```

8.1.1 environments—^Ce

Key	Function	Description
`^Ce<`	boenv()	goes to the `\begin` of the environment
`^Cec`	env_close()	goes to the `\end` of the environment
`^Ce}`	env_close()	inserts an `\end` to close the current environment
`^Cee`	env_prompt()	inserts a new environment with `\begin` and `\end`
`^CeReturn`	env_prompt()	inserts a new environment with `\begin` and `\end`
`^Cer`	env_rename()	renames the current environment
`^Ce>`	eoenv()	goes to the `\end` of the environment

8.1.2 commands—^Cd

8.2 Hot keys for TeX commands

8.2.1 sectioning commands—^Cs

TeX command	Key	TeX command	Key
\appendix	^Csa	\section	^Css
\chapter	^Csc	\subsubsection	^Csb
\minisec	^Csm	\subsection	^Csu
\paragraph	^Csg	\subparagraph	^Csh
\part	^Csp		

8.2.2 font commands—^Cf

todo: Klären, was font_cmd() ist. Liegt auf ^Cfp

TeX cmd.	in text	in mathe	TeX cmd.	in math	everywhere
	mode	mode		mode	
\emph	^Cfe				
			\overline	^Cm_	
\textbf	^Cfb	—	\mathbf	^Cfb	^CfB
			\mathcal	^Cfa/^Cnc	
\textit	^Cfi	—	\mathit	^Cfi	^CfI
			\mathfrak	^Cfk	
\textmd	^Cfm				
\textnormal	^Cfn	—	\mathnormal	^Cfn	^CfN
\textrm	^Cfr	—	\mathrm	^Cfr	^CfR
\textsc	^Cfc				
\textsf	^Cff	—	\mathsf	^Cff	^CfF
\textsl	^Cfs				
\texttt	^Cft	—	\mathtt	^Cft	^CfT
\textup	^Cfu				
\underline	^Cfd/^Cf_	—	\underbar	^Cfd/^Cf_	^CfD
\verb	^Cfv				
\text	^Cfx				

8.2.3 math commands—^Cm and ^Cn

TEX cmd.	key	TEX cmd.	key	TEX cmd.	key
\alpha	^Cma	\geq	^Cm>	\Psi	^CmY
\beta	^Cmb	\hat	^Cm^	\psi	^Cmy
\cap	^Cm-	\in	^Cmi	\rangle	^Cm)
\cdot	^Cm.	\inf	^Cm^_	\rho	^Cmr
\chi	^Cmc	\infty	^CmI/^Cm8	\rightarrow	^Cm^F/^C→
\colon	^Cm:	\int	^Cni	\setminus	^Cm\
\cos	^Cm^C	\kappa	^Cmk	\Sigma	^CmS
\cup	^Cm+	\Lambda	^CmL	\sigma	^Cms
\Delta	^CmD	\lambda	^Cml	\sin	^Cm^S
\delta	^Cmd	\langle	^Cm(	\sqrt	^Cnq
\det	^Cm^D	\leftarrow	^Cm^B/^C←	\subset	^Cm{
\downarrow	^Cm^N/^C↓	\leq	^Cm<	\subseteq	^Cm[
\emptyset	^CmO	\lim	^Cm^L	\sum	^Cns
\epsilon	^Cme	\log	^Cnl	\sup	^Cm^^
\eta	^Cmh	\mu	^Cmm	\supset	^Cm}
\exists	^CmE	\nabla	^CmN	\supseteq	^Cm]
\exp	^Cm^E	\ne	^Cm=	\tan	^Cm^T
\forall	^CmA	\neg	^Cm!	\tau	^Cmt
\frac	^Cnf	\nicefrac	^CnF	\Theta	^CmQ
\frac{1}	^Cn1	\not	^Cm/	\theta	^Cmq
\frac{1}{2}	^Cn2	\nu	^Cmn	\tilde	^Cm~
\frac{1}{3}	^Cn3	\oint	^Cno	\times	^Cm*
\frac{1}{4}	^Cn4	\Omega	^CmO/^CmW	\uparrow	^Cm^P/^C↑
\frac{1}{5}	^Cn5	\omega	^Cmo/^Cmw	\Upsilon	^CmU
\frac{1}{6}	^Cn6	\Phi	^CmV/^CmF	\upsilon	^Cmu
\frac{1}{7}	^Cn7	\phi	^Cmf	\vee	^Cm|/^Cmv
\frac{1}{8}	^Cn8	\Pi	^CmP	\wedge	^Cm&
\frac{1}{9}	^Cn9	\pi	^Cmp	\Xi	^CmX
\Gamma	^CmG	\pmod	^Cnm	\xi	^Cmx
\gamma	^Cmg	\prod	^Cnp	\zeta	^Cmz

8.2.4 links—^Cl

TEX cmd.	key
\cite	^Clb
\index	^Cii
\label	^Cll
\nocite	^Cln
\pageref	^Clp
\url	^Clu

8.2.5 folding—^Co and return

- With the keysequence ^Coo you can fold a region. You can use ESC 1, . . . , ESC 8 to set the level relative from the current that gets folded. With ESC 9 you get a prompt where you can enter an arbitrary level, e. g. -2 to fold the level 2nd levels upstairs.
- With the keysequence ^Cou you can unfold a folded region. You can use it from within the region or at the begin (before the three dots).

  With a prefix argument (set with ESC 1, . . . , ESC 9) you can set the sublevel that should not be unfolded.
- The return key is redefined to unfold a region, if it is inside or before—looking at the three dots—a folded region. Otherwise it acts like everywhere else.

For the definition of levels see chapter 6.

8.3 Mathematical arrows

If you "draw" an arrow it becomes substitured with an corresponding math command. The command is inserted with cmd_insert() (chapter 4) so you have all comforts of cmd_insert().

Input	Substitution	Input	Substitution
->	\rightarrow	-->	\longrightarrow
<-	\leftarrow*	<--	\longleftarrow*
<->	\leftrightarrow	<-->	\longleftrightarrow
=>	\Rightarrow	==>	\Longrightarrow
<=	\Leftarrow*	<==	\Longleftarrow*
<=>	\Leftrightarrow	<==>	\Longleftrightarrow
|->	\mapsto	|-->	\longmapsto
>>	\gg	<<	\ll
‘->	\hookrightarrow		

Some of the input sequences—marked with ∗ in the table—become not substitured immediately, because it's unclear if anything, e.g. a second - or >, follows. They are substitured after the next key press. So don't be confused and write on as if the substitution happend.

Sometimes these substitutions aren't intented. The character that actives the substitution is not alway the last character. The emacs mode of Jed offers two possibilities to work around this substition: ^Q to insert one character without showing it to the substition function and ^Xq to insert a string.

8.4 Mathematical sub- and superscripts

The keys `_` and `^` are automaticly surround the word (all alphanumeric characters and a possible `\` at the begin) before the editing point with $, if it is not still in math mode. If the word is a TeX command it is tried to find a completion from former usages of the command. This is very helpful when you write equations with

```
\sum_{foo}^{bar} 12+2 = \sum_{foo}^{bar} 10+4
```

If you don't want the completion type in what you want. If you accept the completion than hit `Enter` or `Return`. If there is also a completion for the counterpar, it is also presented and you can accept it with `Enter` or `Return` or you type what you want. The visible mark shows you which part is offered and becomes removed if you don't accept.

If you started in text mode, the editing point is placed after accepted completion in text mode. This is helpful when you write things like

```
First, we look at $\alpha_{i}$. The formula becomes
    the truth true, iff $\alpha_{i}$ is even.
```

Than you don't have to skip the dollar sign after the subscript was completed.

The sub- and superscript function treats some TeX commands specially. An sub- or superscript for an `\rightarrow` changes the command to `\xrightarrow` from AMSmath. The commands `\cup`, `\cap`, `\vee` and `\wedge` become `\bigcup`, `\bigcap`, `\bigvee` and `\bigwedge`, respectively.