

# Emacs TeQ: (T<sub>E</sub>X + Quail)

Input Method written in Quail for entering L<sup>A</sup>T<sub>E</sub>X math expressions

Garid Zorigoo

April 12, 2023

## Contents

<b>1</b>	<b>Alphabet related stuff :</b>	<b>1</b>
1.1	Greek . . . . .	1
1.2	Matrix (aka bold) . . . . .	3
1.3	Vector & Hat . . . . .	4
1.4	Dots . . . . .	5
1.5	DDots . . . . .	6
<b>2</b>	<b>Function Expansion</b>	<b>7</b>
<b>3</b>	<b>Symbols :</b>	<b>7</b>
3.1	Dots related . . . . .	7
3.2	Geometry . . . . .	7
3.3	Letter like . . . . .	8
3.4	Spaces . . . . .	8
3.5	Arrows: . . . . .	9
3.5.1	Single: . . . . .	9
3.5.2	Double: . . . . .	10
3.5.3	Long arrow with top-bottom entries . . . . .	10
<b>4</b>	<b>Symbol Modification</b>	<b>11</b>
4.1	Accents (variable decoration?) . . . . .	11
4.2	Superscripts & Subscripts (power & lower) . . . . .	12
<b>5</b>	<b>Binary Operation Symbols</b>	<b>12</b>
5.1	Simple Arithmetics: . . . . .	12
5.2	Binary Relations: . . . . .	14
5.3	Set symbols . . . . .	15

5.4	Logic . . . . .	15
<b>6</b>	<b>Functions</b>	<b>16</b>
6.1	Function . . . . .	16
6.2	Trigonometry: function . . . . .	16
6.3	Iterative-like operation: . . . . .	17
<b>7</b>	<b>Structural:</b>	<b>18</b>
7.1	Parenthesis Related . . . . .	18
7.2	Texts: . . . . .	19
7.3	Misc. . . . .	19
7.4	xy Diagram related . . . . .	19
<b>8</b>	<b>Formatting Table into Elisp</b>	<b>19</b>
<b>9</b>	<b>Executable elisp function definition</b>	<b>31</b>
<b>10</b>	<b>Making the el</b>	<b>32</b>

## 1 Alphabet related stuff :

### 1.1 Greek

`gifs/example-greek.gif`

Table 1: Main Greek letters

key	sym	latex (lower greek)	key	sym	latex (upper greek)
a.	$\alpha$	<code>\alpha</code>	A.	$A$	<code>A</code>
b.	$\beta$	<code>\beta</code>	B.	$B$	<code>B</code>
c.	$\psi$	<code>\psi</code>	C.	$\Psi$	<code>\Psi</code>
d.	$\delta$	<code>\delta</code>	D.	$\Delta$	<code>\Delta</code>
e.	$\epsilon$	<code>\epsilon</code>	E.	$E$	<code>E</code>
f.	$\phi$	<code>\phi</code>	F.	$\Phi$	<code>\Phi</code>
g.	$\gamma$	<code>\gamma</code>	G.	$\Gamma$	<code>\Gamma</code>
h.	$\eta$	<code>\eta</code>	H.	$H$	<code>H</code>
i.	$\iota$	<code>\iota</code>	I.	$I$	<code>I</code>
j.	$\xi$	<code>\xi</code>	J.	$\Xi$	<code>\Xi</code>
k.	$\kappa$	<code>\kappa</code>	K.	$K$	<code>K</code>
l.	$\lambda$	<code>\lambda</code>	L.	$\Lambda$	<code>\Lambda</code>
m.	$\mu$	<code>\mu</code>	M.	$M$	<code>M</code>
n.	$\nu$	<code>\nu</code>	N.	$N$	<code>N</code>
o.	$o$	<code>o</code>	O.	$O$	<code>O</code>
p.	$\pi$	<code>\pi</code>	P.	$\Pi$	<code>\Pi</code>
r.	$\rho$	<code>\rho</code>	R.	$P$	<code>P</code>
s.	$\sigma$	<code>\sigma</code>	S.	$\Sigma$	<code>\Sigma</code>
t.	$\tau$	<code>\tau</code>	T.	$T$	<code>T</code>
th.	$\theta$	<code>\theta</code>	Th.	$\Theta$	<code>\Theta</code>
u.	$v$	<code>\upsilon</code>	U.	$\Upsilon$	<code>\Upsilon</code>
w.	$\omega$	<code>\omega</code>	W.	$\Omega$	<code>\Omega</code>
x.	$\chi$	<code>\chi</code>	X.	$X$	<code>X</code>
z.	$\zeta$	<code>\zeta</code>	Z.	$Z$	<code>Z</code>

Table 2: Variation Greek letters

key	sym	latex (lower greek)
e..	$\varepsilon$	<code>\varepsilon</code>
f..	$\varphi$	<code>\varphi</code>
s..	$\varsigma$	<code>\varsigma</code>
t..	$\vartheta$	<code>\vartheta</code>
r..	$\varrho$	<code>\varrho</code>
p..	$\varpi$	<code>\varpi</code>
k..	$\varkappa$	<code>\varkappa</code>

## 1.2 Matrix (aka bold)

Table 3: Matrix

key	sym	latex (upper bold)	key	sym	latex (lower bold)
Am	<b>A</b>	<code>\mathbf{A}</code>	am	<b>a</b>	<code>\mathbf{a}</code>
Bm	<b>B</b>	<code>\mathbf{B}</code>	bm	<b>b</b>	<code>\mathbf{b}</code>
Cm	<b>C</b>	<code>\mathbf{C}</code>	cm	<b>c</b>	<code>\mathbf{c}</code>
Dm	<b>D</b>	<code>\mathbf{D}</code>	dm	<b>d</b>	<code>\mathbf{d}</code>
Em	<b>E</b>	<code>\mathbf{E}</code>	em	<b>e</b>	<code>\mathbf{e}</code>
Fm	<b>F</b>	<code>\mathbf{F}</code>	fm	<b>f</b>	<code>\mathbf{f}</code>
Gm	<b>G</b>	<code>\mathbf{G}</code>	gm	<b>g</b>	<code>\mathbf{g}</code>
Hm	<b>H</b>	<code>\mathbf{H}</code>	hm	<b>h</b>	<code>\mathbf{h}</code>
Im	<b>I</b>	<code>\mathbf{I}</code>	im	<b>i</b>	<code>\mathbf{i}</code>
Jm	<b>J</b>	<code>\mathbf{J}</code>	jm	<b>j</b>	<code>\mathbf{j}</code>
Km	<b>K</b>	<code>\mathbf{K}</code>	km	<b>k</b>	<code>\mathbf{k}</code>
Lm	<b>L</b>	<code>\mathbf{L}</code>	lm	<b>l</b>	<code>\mathbf{l}</code>
Mm	<b>M</b>	<code>\mathbf{M}</code>	mm	<b>m</b>	<code>\mathbf{m}</code>
Nm	<b>N</b>	<code>\mathbf{N}</code>	nm	<b>n</b>	<code>\mathbf{n}</code>
Om	<b>O</b>	<code>\mathbf{O}</code>	om	<b>o</b>	<code>\mathbf{o}</code>
Pm	<b>P</b>	<code>\mathbf{P}</code>	pm	<b>p</b>	<code>\mathbf{p}</code>
Qm	<b>Q</b>	<code>\mathbf{Q}</code>	qm	<b>q</b>	<code>\mathbf{q}</code>
Rm	<b>R</b>	<code>\mathbf{R}</code>	rm	<b>r</b>	<code>\mathbf{r}</code>
Sm	<b>S</b>	<code>\mathbf{S}</code>	sm	<b>s</b>	<code>\mathbf{s}</code>
Tm	<b>T</b>	<code>\mathbf{T}</code>	tm	<b>t</b>	<code>\mathbf{t}</code>
Um	<b>U</b>	<code>\mathbf{U}</code>	um	<b>u</b>	<code>\mathbf{u}</code>
Vm	<b>V</b>	<code>\mathbf{V}</code>	vm	<b>v</b>	<code>\mathbf{v}</code>
Wm	<b>W</b>	<code>\mathbf{W}</code>	wm	<b>w</b>	<code>\mathbf{w}</code>
Xm	<b>X</b>	<code>\mathbf{X}</code>	xm	<b>x</b>	<code>\mathbf{x}</code>
Ym	<b>Y</b>	<code>\mathbf{Y}</code>	ym	<b>y</b>	<code>\mathbf{y}</code>
Zm	<b>Z</b>	<code>\mathbf{Z}</code>	zm	<b>z</b>	<code>\mathbf{z}</code>
Om	<b>0</b>	<code>\mathbf{0}</code>	Om	<b>0</b>	<code>\mathbf{0}</code>

### 1.3 Vector & Hat

Table 4: Vectors and Hats

key	sym	latex (vec)	key	sym	latex (hat)
av	$\vec{a}$	<code>\vec{a}</code>	ah	$\hat{a}$	<code>\hat{a}</code>
bv	$\vec{b}$	<code>\vec{b}</code>	bh	$\hat{b}$	<code>\hat{b}</code>
cv	$\vec{c}$	<code>\vec{c}</code>	ch	$\hat{c}$	<code>\hat{c}</code>
dv	$\vec{d}$	<code>\vec{d}</code>	dh	$\hat{d}$	<code>\hat{d}</code>
ev	$\vec{e}$	<code>\vec{e}</code>	eh	$\hat{e}$	<code>\hat{e}</code>
fv	$\vec{f}$	<code>\vec{f}</code>	fh	$\hat{f}$	<code>\hat{f}</code>
gv	$\vec{g}$	<code>\vec{g}</code>	gh	$\hat{g}$	<code>\hat{g}</code>
hv	$\vec{h}$	<code>\vec{h}</code>	hh	$\hat{h}$	<code>\hat{h}</code>
iv	$\vec{i}$	<code>\vec{i}</code>	ih	$\hat{i}$	<code>\hat{i}</code>
jv	$\vec{j}$	<code>\vec{j}</code>	jh	$\hat{j}$	<code>\hat{j}</code>
kv	$\vec{k}$	<code>\vec{k}</code>	kh	$\hat{k}$	<code>\hat{k}</code>
lv	$\vec{l}$	<code>\vec{l}</code>	lh	$\hat{l}$	<code>\hat{l}</code>
mv	$\vec{m}$	<code>\vec{m}</code>	mh	$\hat{m}$	<code>\hat{m}</code>
nv	$\vec{n}$	<code>\vec{n}</code>	nh	$\hat{n}$	<code>\hat{n}</code>
ov	$\vec{o}$	<code>\vec{o}</code>	oh	$\hat{o}$	<code>\hat{o}</code>
pv	$\vec{p}$	<code>\vec{p}</code>	ph	$\hat{p}$	<code>\hat{p}</code>
qv	$\vec{q}$	<code>\vec{q}</code>	qh	$\hat{q}$	<code>\hat{q}</code>
rv	$\vec{r}$	<code>\vec{r}</code>	rh	$\hat{r}$	<code>\hat{r}</code>
sv	$\vec{s}$	<code>\vec{s}</code>	sh	$\hat{s}$	<code>\hat{s}</code>
tv	$\vec{t}$	<code>\vec{t}</code>	th	$\hat{t}$	<code>\hat{t}</code>
uv	$\vec{u}$	<code>\vec{u}</code>	uh	$\hat{u}$	<code>\hat{u}</code>
vv	$\vec{v}$	<code>\vec{v}</code>	vh	$\hat{v}$	<code>\hat{v}</code>
wv	$\vec{w}$	<code>\vec{w}</code>	wh	$\hat{w}$	<code>\hat{w}</code>
xv	$\vec{x}$	<code>\vec{x}</code>	xh	$\hat{x}$	<code>\hat{x}</code>
yv	$\vec{y}$	<code>\vec{y}</code>	yh	$\hat{y}$	<code>\hat{y}</code>
zv	$\vec{z}$	<code>\vec{z}</code>	zh	$\hat{z}$	<code>\hat{z}</code>

## 1.4 Dots

Table 5: Dots

key	sym	latex (vec)	key	sym	latex (hat)
ad	$\dot{a}$	<code>\dot{a}</code>	Ad	$\dot{A}$	<code>\dot{A}</code>
bd	$\dot{b}$	<code>\dot{b}</code>	Bd	$\dot{B}$	<code>\dot{B}</code>
cd	$\dot{c}$	<code>\dot{c}</code>	Cd	$\dot{C}$	<code>\dot{C}</code>
dd	$\dot{d}$	<code>\dot{d}</code>	Dd	$\dot{D}$	<code>\dot{D}</code>
ed	$\dot{e}$	<code>\dot{e}</code>	Ed	$\dot{E}$	<code>\dot{E}</code>
fd	$\dot{f}$	<code>\dot{f}</code>	Fd	$\dot{F}$	<code>\dot{F}</code>
gd	$\dot{g}$	<code>\dot{g}</code>	Gd	$\dot{G}$	<code>\dot{G}</code>
hd	$\dot{h}$	<code>\dot{h}</code>	Hd	$\dot{H}$	<code>\dot{H}</code>
id	$\dot{i}$	<code>\dot{i}</code>	Id	$\dot{I}$	<code>\dot{I}</code>
jd	$\dot{j}$	<code>\dot{j}</code>	Jd	$\dot{J}$	<code>\dot{J}</code>
kd	$\dot{k}$	<code>\dot{k}</code>	Kd	$\dot{K}$	<code>\dot{K}</code>
ld	$\dot{l}$	<code>\dot{l}</code>	Ld	$\dot{L}$	<code>\dot{L}</code>
md	$\dot{m}$	<code>\dot{m}</code>	Md	$\dot{M}$	<code>\dot{M}</code>
nd	$\dot{n}$	<code>\dot{n}</code>	Nd	$\dot{N}$	<code>\dot{N}</code>
od	$\dot{o}$	<code>\dot{o}</code>	Od	$\dot{O}$	<code>\dot{O}</code>
pd	$\dot{p}$	<code>\dot{p}</code>	Pd	$\dot{P}$	<code>\dot{P}</code>
qd	$\dot{q}$	<code>\dot{q}</code>	Qd	$\dot{Q}$	<code>\dot{Q}</code>
rd	$\dot{r}$	<code>\dot{r}</code>	Rd	$\dot{R}$	<code>\dot{R}</code>
sd	$\dot{s}$	<code>\dot{s}</code>	Sd	$\dot{S}$	<code>\dot{S}</code>
td	$\dot{t}$	<code>\dot{t}</code>	Td	$\dot{T}$	<code>\dot{T}</code>
ud	$\dot{u}$	<code>\dot{u}</code>	Ud	$\dot{U}$	<code>\dot{U}</code>
vd	$\dot{v}$	<code>\dot{v}</code>	Vd	$\dot{V}$	<code>\dot{V}</code>
wd	$\dot{w}$	<code>\dot{w}</code>	Wd	$\dot{W}$	<code>\dot{W}</code>
xd	$\dot{x}$	<code>\dot{x}</code>	Xd	$\dot{X}$	<code>\dot{X}</code>
yd	$\dot{y}$	<code>\dot{y}</code>	Yd	$\dot{Y}$	<code>\dot{Y}</code>
zd	$\dot{z}$	<code>\dot{z}</code>	Zd	$\dot{Z}$	<code>\dot{Z}</code>

## 1.5 DDots

Table 6: DDots

key	sym	latex (vec)	key	sym	latex (hat)
ad.	$\ddot{a}$	<code>\ddot{a}</code>	Ad.	$\ddot{A}$	<code>\ddot{A}</code>
bd.	$\ddot{b}$	<code>\ddot{b}</code>	Bd.	$\ddot{B}$	<code>\ddot{B}</code>
cd.	$\ddot{c}$	<code>\ddot{c}</code>	Cd.	$\ddot{C}$	<code>\ddot{C}</code>
dd.	$\ddot{d}$	<code>\ddot{d}</code>	Dd.	$\ddot{D}$	<code>\ddot{D}</code>
ed.	$\ddot{e}$	<code>\ddot{e}</code>	Ed.	$\ddot{E}$	<code>\ddot{E}</code>
fd.	$\ddot{f}$	<code>\ddot{f}</code>	Fd.	$\ddot{F}$	<code>\ddot{F}</code>
gd.	$\ddot{g}$	<code>\ddot{g}</code>	Gd.	$\ddot{G}$	<code>\ddot{G}</code>
hd.	$\ddot{h}$	<code>\ddot{h}</code>	Hd.	$\ddot{H}$	<code>\ddot{H}</code>
id.	$\ddot{i}$	<code>\ddot{i}</code>	Id.	$\ddot{I}$	<code>\ddot{I}</code>
jd.	$\ddot{j}$	<code>\ddot{j}</code>	Jd.	$\ddot{J}$	<code>\ddot{J}</code>
kd.	$\ddot{k}$	<code>\ddot{k}</code>	Kd.	$\ddot{K}$	<code>\ddot{K}</code>
ld.	$\ddot{l}$	<code>\ddot{l}</code>	Ld.	$\ddot{L}$	<code>\ddot{L}</code>
md.	$\ddot{m}$	<code>\ddot{m}</code>	Md.	$\ddot{M}$	<code>\ddot{M}</code>
nd.	$\ddot{n}$	<code>\ddot{n}</code>	Nd.	$\ddot{N}$	<code>\ddot{N}</code>
od.	$\ddot{o}$	<code>\ddot{o}</code>	Od.	$\ddot{O}$	<code>\ddot{O}</code>
pd.	$\ddot{p}$	<code>\ddot{p}</code>	Pd.	$\ddot{P}$	<code>\ddot{P}</code>
qd.	$\ddot{q}$	<code>\ddot{q}</code>	Qd.	$\ddot{Q}$	<code>\ddot{Q}</code>
rd.	$\ddot{r}$	<code>\ddot{r}</code>	Rd.	$\ddot{R}$	<code>\ddot{R}</code>
sd.	$\ddot{s}$	<code>\ddot{s}</code>	Sd.	$\ddot{S}$	<code>\ddot{S}</code>
td.	$\ddot{t}$	<code>\ddot{t}</code>	Td.	$\ddot{T}$	<code>\ddot{T}</code>
ud.	$\ddot{u}$	<code>\ddot{u}</code>	Ud.	$\ddot{U}$	<code>\ddot{U}</code>
vd.	$\ddot{v}$	<code>\ddot{v}</code>	Vd.	$\ddot{V}$	<code>\ddot{V}</code>
wd.	$\ddot{w}$	<code>\ddot{w}</code>	Wd.	$\ddot{W}$	<code>\ddot{W}</code>
xd.	$\ddot{x}$	<code>\ddot{x}</code>	Xd.	$\ddot{X}$	<code>\ddot{X}</code>
yd.	$\ddot{y}$	<code>\ddot{y}</code>	Yd.	$\ddot{Y}$	<code>\ddot{Y}</code>
zd.	$\ddot{z}$	<code>\ddot{z}</code>	Zd.	$\ddot{Z}$	<code>\ddot{Z}</code>

## 2 Function Expansion

Table 7: Keys that will execute some elisp functions

key	trans	sym	description
/	quail-TeX-fraction	$\frac{\Box}{\Box}$	fraction on previous
eq	quail-TeX-equation		equation environment
al	quail-TeX-aligned		aligned environment
el	quail-TeX-endofline		end of line

## 3 Symbols :

### 3.1 Dots related

Table 8: Multiple Dots Related

key	trans	sym	description
...	<code>\dots</code>	...	3 dots
.v	<code>\vdots</code>	$\vdots$	vertical dots
.d	<code>\ddots</code>	$\ddots$	diagonale dots
.l	<code>\ldots</code>	...	low dots

### 3.2 Geometry

Table 9:

key	trans	sym	description
perp	<code>\perp</code>	$\perp$	
perpn	<code>\perp</code>	$\nparallel$	$\perp$ n (neg)
para	<code>\parallel</code>	$\parallel$	
paran	<code>\nparallel</code>	$\nparallel$	$\parallel$ n (neg)
ang	<code>\angle</code>	$\angle$	
ang.	<code>\measuredangle</code>	$\sphericalangle$	$\angle$ . (var)
tri	<code>\vartriangle</code>	$\triangle$	
trin	<code>\triangledown</code>	$\nabla$	$\triangle$ n (neg)
squ	<code>\square</code>	$\square$	
tri.	<code>\blacktriangle</code>	$\blacktriangle$	$\triangle$ . (var)
trin.	<code>\blacktriangledown</code>	$\blacktriangledown$	$\triangle$ n. (neg,var)
squ.	<code>\blacksquare</code>	$\blacksquare$	$\square$ . (var)



### 3.3 Letter like

Table 10: Letter-like Symbol

key	trans	sym	description
inf	<code>\infty</code>	$\infty$	
ex	<code>\exists</code>	$\exists$	
exn	<code>\nexists</code>	$\nexists$	$\exists + \underline{n}$ (neg)
fa	<code>\forall</code>	$\forall$	
hb	<code>\hbar</code>	$\hbar$	
hb.	<code>\hslash</code>	$\hbar$	$\hbar + \underline{\cdot}$ (var)
dd	<code>\mathrm{d}</code>	$\mathrm{d}$	
dd.	<code>\partial</code>	$\partial$	$\mathrm{d} + \underline{\cdot}$ (var)
ii	<code>\imath</code>	$\imath$	
jj	<code>\jmath</code>	$\jmath$	
nab	<code>\nabla</code>	$\nabla$	
cm	<code>\checkmark</code>	$\checkmark$	

### 3.4 Spaces

Table 11: Space Symbol

key	trans	sym	description
qu	<code>\quad</code>		
quu	<code>\qquad</code>		

### 3.5 Arrows:

#### 3.5.1 Single:

Table 12: Single Line arrows

key	trans	sym	description
<-	<code>\leftarrow</code>	$\leftarrow$	
->	<code>\rightarrow</code>	$\rightarrow$	
-^	<code>\uparrow</code>	$\uparrow$	$\wedge$ looks like up arrow head
-v	<code>\downarrow</code>	$\downarrow$	$\vee$ looks like down arrow head
<->	<code>\leftrightharpoonup</code>	$\leftrightarrow$	
<-n	<code>\nleftarrow</code>	$\nleftarrow$	negate ( <u>n</u> ) of prev. section
->n	<code>\nrightarrow</code>	$\nrightarrow$	arrows + <u>n</u>
-^n	<code>\nuparrow</code>	$\nuparrow$	
-vn	<code>\ndownarrow</code>	$\ndownarrow$	
<->	<code>\nleftrightharpoonup</code>	$\nleftrightarrow$	
-->	<code>\longrightarrow</code>	$\longrightarrow$	longer arrows, with 2 dashes
<--	<code>\longleftarrow</code>	$\longleftarrow$	
->	<code>\mapsto</code>	$\mapsto$	vertical-bar + -> (this rendered wrong on Github) (check from PDF file)

### 3.5.2 Double:

Table 13: Double Line arrows

key	trans	sym	description
<code>&lt;=</code>	<code>\Leftarrow</code>	$\Leftarrow$	compared to single arrow
<code>=&gt;</code>	<code>\Rightarrow</code>	$\Rightarrow$	these uses = as the arrow shaft
<code>=^</code>	<code>\Uparrow</code>	$\Uparrow$	
<code>=v</code>	<code>\Downarrow</code>	$\Downarrow$	
<code>&lt;=&gt;</code>	<code>\Leftrightarrow</code>	$\Leftrightarrow$	
<code>iff</code>	<code>\Leftrightarrow</code>	$\Leftrightarrow$	
<code>&lt;=n</code>	<code>\nLeftarrow</code>	$\nLeftarrow$	negate ( <u>n</u> ) of prev. section
<code>=&gt;n</code>	<code>\nRightarrow</code>	$\nRightarrow$	arrows + <u>n</u>
<code>&lt;=&gt;n</code>	<code>\nLeftrightarrow</code>	$\nLeftrightarrow$	
<code>iffn</code>	<code>\nLeftrightarrow</code>	$\nLeftrightarrow$	
<code>&lt;==&gt;</code>	<code>\Longleftrightarrow</code>	$\Longleftrightarrow$	longer arrows, with 2 dashes
<code>&lt;==</code>	<code>\Longleftarrow</code>	$\Longleftarrow$	
<code>==&gt;</code>	<code>\Longrightarrow</code>	$\Longrightarrow$	

### 3.5.3 Long arrow with top-bottom entries

Table 14: Long arrow Line arrows

key	trans	sym	description
<code>&lt;--</code>	<code>\xleftarrow[ ]{ }</code>	$\xleftarrow[ ]{ }$	these uses triple - or =
<code>--&gt;</code>	<code>\xrightarrow[ ]{ }</code>	$\xrightarrow[ ]{ }$	
<code>==&gt;</code>	<code>\xRightarrow[ ]{ }</code>	$\xRightarrow[ ]{ }$	mathtools lib required
<code>&lt;==</code>	<code>\xLeftarrow[ ]{ }</code>	$\xLeftarrow[ ]{ }$	mathtools lib required

## 4 Symbol Modification

### 4.1 Accents (variable decoration?)

Table 15:

key	trans	sym	description
vec	<code>\vec{}</code>	$\vec{\square}$	
bar	<code>\bar{}</code>	$\bar{\square}$	
hat	<code>\hat{}</code>	$\hat{\square}$	
dot	<code>\dot{}</code>	$\dot{\square}$	
dot.	<code>\ddot{}</code>	$\ddot{\square}$	
dot..	<code>\ddd\dot{}</code>	$\dddot{\square}$	
dot...	<code>\dddd\dot{}</code>	$\dddd\dot{\square}$	
dag	<code>^\dag</code>	$\square^\dagger$	
dag.	<code>^\ddag</code>	$\square^\ddagger$	
*..	<code>^*</code>	$\square^*$	
deg	<code>^\circ</code>	$\square^\circ$	
tr	<code>^T</code>	$\square^T$	
tr.	<code>^{-T}</code>	$\square^{-T}$	

## 4.2 Superscripts & Subscripts (power & lower)

Table 16:

key	sym	trans	key	sym	trans
^	$\square^\square$	$\sim\{$	-	$\square_\square$	$\_ \{$
pp	$\square^\square$	$\sim\{$	11	$\square_\square$	$\_ \{$
p0	$\square^0$	$\sim 0$	10	$\square_0$	$\_ 0$
p1	$\square^1$	$\sim 1$	11	$\square_1$	$\_ 1$
p2	$\square^2$	$\sim 2$	12	$\square_2$	$\_ 2$
p3	$\square^3$	$\sim 3$	13	$\square_3$	$\_ 3$
p4	$\square^4$	$\sim 4$	14	$\square_4$	$\_ 4$
pn	$\square^n$	$\sim n$	lnn	$\square_n$	$\_ n$
px	$\square^x$	$\sim x$	li	$\square_i$	$\_ i$
--	$\square_\square$	$\backslash \underset{\square}{\square} \{ \} \{ \}$	^^	$\square^\square$	$\backslash \overset{\square}{\square} \{ \} \{ \}$
---	$\underbrace{\square}_\square$	$\backslash \underbrace{\square}_\square \{ \}_\{ \}$	^^.	$\overbrace{\square}^\square$	$\backslash \overbrace{\square}^\square \{ \}^\{ \}$
---	$\underline{\square}$	$\backslash \underline{\square} \{ \}$	^^..	$\overline{\square}$	$\backslash \overline{\square} \{ \}$

## 5 Binary Operation Symbols

### 5.1 Simple Arithmetics:

Table 17: Simple Arithmetics operations

key	trans	sym
+-	$\backslash pm$	$\pm$
-+	$\backslash mp$	$\mp$
*x	$\backslash times$	$\times$
::	$\backslash div$	$\div$
**	$\backslash cdot$	$\cdot$



## 5.2 Binary Relations:

Table 18:

key	trans	sym	description
=.	<code>\equiv</code>	$\equiv$	Variation on
<.	<code>\leq</code>	$\leq$	$< = >$
>.	<code>\geq</code>	$\geq$	symbols
«	<code>\ll</code>	$\ll$	
»	<code>\gg</code>	$\gg$	
=n	<code>\neq</code>	$\neq$	negation
~n	<code>\nsim</code>	$\approx$	
<n	<code>\nless</code>	$\nless$	
>n	<code>\ngtr</code>	$\ngtr$	
<.n	<code>\nleq</code>	$\nleq$	
>.n	<code>\ngeq</code>	$\ngeq$	
=?	<code>\stackrel{?}{=}</code>	$\stackrel{?}{=}$	with question mark
<?	<code>\stackrel{?}{&lt;}</code>	$\stackrel{?}{<}$	
>?	<code>\stackrel{?}{&gt;}</code>	$\stackrel{?}{>}$	
<.?	<code>\stackrel{?}{&lt;}\leq</code>	$\stackrel{?}{<}\leq$	
>.?	<code>\stackrel{?}{&gt;}\geq</code>	$\stackrel{?}{>}\geq$	
«?	<code>\stackrel{?}{\ll}</code>	$\stackrel{?}{\ll}$	
»?	<code>\stackrel{?}{\gg}</code>	$\stackrel{?}{\gg}$	
=y	<code>\stackrel{\checkmark}{=}</code>	$\stackrel{\checkmark}{=}$	with check mark
<y	<code>\stackrel{\checkmark}{&lt;}</code>	$\stackrel{\checkmark}{<}$	
>y	<code>\stackrel{\checkmark}{&gt;}</code>	$\stackrel{\checkmark}{>}$	
<.y	<code>\stackrel{\checkmark}{&lt;}\leq</code>	$\stackrel{\checkmark}{<}\leq$	
>.y	<code>\stackrel{\checkmark}{&gt;}\geq</code>	$\stackrel{\checkmark}{>}\geq$	
«y	<code>\stackrel{\checkmark}{\ll}</code>	$\stackrel{\checkmark}{\ll}$	
»y	<code>\stackrel{\checkmark}{\gg}</code>	$\stackrel{\checkmark}{\gg}$	
~.	<code>\sim</code>	$\sim$	Another variations
~~	<code>\approx</code>	$\approx$	on =
3=	<code>\equiv</code>	$\equiv$	
=:	<code>\coloneqq</code>	$\coloneqq$	
:=	<code>\coloneqq</code>	$\coloneqq$	

### 5.3 Set symbols

Table 19:

key	trans	sym	description
in	<code>\in</code>	$\in$	
in.	<code>\ni</code>	$\ni$	
ni	<code>\ni</code>	$\ni$	
inn	<code>\notin</code>	$\notin$	(neg)
0/	<code>\emptyset</code>	$\emptyset$	
nsr	<code>\mathbb{R}</code>	$\mathbb{R}$	(n)umber (s)et (r)eal
nsc	<code>\mathbb{C}</code>	$\mathbb{C}$	(n)umber (s)et (c)omplex
nsn	<code>\mathbb{N}</code>	$\mathbb{N}$	...
nsp	<code>\mathbb{P}</code>	$\mathbb{P}$	...
nsz	<code>\mathbb{Z}</code>	$\mathbb{Z}$	...
nsi	<code>\mathbb{I}</code>	$\mathbb{I}$	...
sub	<code>\subset</code>	$\subset$	
subn	<code>\nssubseteq</code>	$\not\subseteq$	(neg)
sub.	<code>\subseteq</code>	$\subseteq$	(var)
sub.n	<code>\nsubseteq</code>	$\not\subseteq$	(var, neg)
subn.	<code>\nsubseteq</code>	$\not\subseteq$	(neg, var)
sup	<code>\supset</code>	$\supset$	
supn	<code>\nsupseteq</code>	$\not\supseteq$	(neg)
sup.	<code>\supeseteq</code>	$\supseteq$	(var)
sup.n	<code>\nsupseteq</code>	$\not\supseteq$	(var, neg)
supn.	<code>\nsupseteq</code>	$\not\supseteq$	(neg, var)

### 5.4 Logic

Table 20:

key	trans	sym	description
or	<code>\lor</code>	$\vee$	
and	<code>\lnd</code>	$\wedge$	
not	<code>\neg</code>	$\neg$	
or.	<code>\text{ or }</code>	or	(var)
and.	<code>\text{ and }</code>	and	(var)
not.	<code>\text{ not }</code>	not	(var)



## 6 Functions

### 6.1 Function

Table 21:

key	trans	sym	description
rank	<code>\mathrm{rank}</code>	rank	
arg	<code>\arg</code>	arg	
det	<code>\det</code>	det	
dim	<code>\dim</code>	dim	
exp	<code>\exp(</code>	exp	
Im	<code>\mathrm{Im}(</code>	Im	
Re	<code>\mathrm{Re}(</code>	Re	
ln	<code>\ln(</code>	ln	
log	<code>\log(</code>	log	
max	<code>\max(</code>	max	
min	<code>\min(</code>	min	
dim	<code>\dim(</code>	dim	
sqrt	<code>\sqrt{</code>	$\sqrt{\phantom{x}}$	
mod	<code>\pmod{</code>	$\square \pmod{\square}$	
mod.	<code>\mod</code>	$\square \bmod \square$	
mod..	<code>\bmod</code>	$\square \bmod \square$	

### 6.2 Trigonometry: function

Table 22:

key	sym	trans	key	sym	trans
cos	$\cos(\square)$	<code>\cos(</code>	cosh	$\cosh(\square)$	<code>\cosh(</code>
sin	$\sin(\square)$	<code>\sin(</code>	sinh	$\sinh(\square)$	<code>\sinh(</code>
tan	$\tan(\square)$	<code>\tan(</code>	tanh	$\tanh(\square)$	<code>\tanh(</code>
cot	$\cot(\square)$	<code>\cot(</code>	coth	$\coth(\square)$	<code>\coth(</code>
acos	$\arccos(\square)$	<code>\arccos(</code>	cos.	$\arccos(\square)$	<code>\arccos(</code>
asin	$\arcsin(\square)$	<code>\arcsin(</code>	sin.	$\arcsin(\square)$	<code>\arcsin(</code>
atan	$\arctan(\square)$	<code>\arctan(</code>	tan.	$\arctan(\square)$	<code>\arctan(</code>

### 6.3 Iterative-like operation:

Table 23: Integrals, Sums, Products

key	trans	sym	description
il	<code>\limits_{ }</code>	$\sum_{\square}$	
il	<code>\limits_{ }^{\{ }</code>	$\sum_{\square}^{\{ }$	
lim	<code>\lim</code>	lim	
sum	<code>\sum</code>	$\sum$	
prod	<code>\prod</code>	$\prod$	
int	<code>\int</code>	$\int$	
inti	<code>\iint</code>	$\iint$	
intii	<code>\iiint</code>	$\iiint$	
intiii	<code>\iiiiint</code>	$\iiiiint$	
into	<code>\oint</code>	$\oint$	
sum.	<code>\sum\limits_{ i=1 }^{\{ n }</code>	$\sum_{i=1}^n$	
prod.	<code>\prod\limits_{ i=1 }^{\{ n }</code>	$\prod_{i=1}^n$	
int.	<code>\int\limits_{ }^{\{ }</code>	$\int_{\square}^{\{ }$	
int..	<code>\int\limits_{ 0 }^{\{ +\infty }</code>	$\int_0^{+\infty}$	
int...	<code>\int\limits_{ -\infty }^{\{ +\infty }</code>	$\int_{-\infty}^{+\infty}$	
inti.	<code>\iint\limits_{ }</code>	$\iint_C$	
intii.	<code>\iiint\limits_{ }</code>	$\iiint_C$	
intiii.	<code>\iiiiint\limits_{ }</code>	$\iiiiint_C$	
into.	<code>\oint\limits_{ }</code>	$\oint_C$	

## 7 Structural:

### 7.1 Parenthesis Related

Table 24:

key	trans	sym	description
().	<code>\left( \right)</code>	$(\square)$	
()..	<code>\left( \middle\vert \right)</code>	$(\square \square)$	
[].	<code>\left[ \right]</code>	$[\square]$	
[]..	<code>\left[ \middle\vert \right]</code>	$[\square \square]$	(var)
[] .c	<code>\lceil \rceil</code>	$\lceil \square \rceil$	(var) (ceil)
[] .f	<code>\lfloor \rfloor</code>	$\lfloor \square \rfloor$	(var) (floor)
{ }.	<code>\left\{ \right\}</code>	$\{\square\}$	
{ }..	<code>\left\{ \middle\vert \right\}</code>	$\{\square \square\}$	(var)
< >.	<code>\left&lt; \right&gt;</code>	$\langle \square \rangle$	
< >..	<code>\left&lt; \middle\vert \right&gt;</code>	$\langle \square \square \rangle$	(var)
.	<code>\left\  \right\ </code>	$\ \square\ $	
..	<code>\left\  \middle\  \right\ </code>	$\ \square\ \square$	(var)
(.	<code>\left(</code>	$(\square$	half (
).	<code>\right)</code>	$\square)$	half )
[.	<code>\left[</code>	$[\square$	half [
].	<code>\right]</code>	$\square]$	half ]
{.	<code>\left\{</code>	$\{\square$	half {
}.	<code>\right\}</code>	$\square\}$	half }
<.	<code>\left&lt;</code>	$\langle \square$	half <
>.	<code>\right&gt;</code>	$\square\rangle$	half >
(..	<code>\left.</code>		half left .
)..	<code>\right.</code>		half right .
.	<code>\Bigg\  \middle\  \right\ </code>	$\Bigg\  \middle\  \right\ $	definite integral range

## 7.2 Texts:

Table 25:

key	trans	sym	description
te	<code>\text{}</code>	$\square + \text{text}$	(te)xt
tr	<code>\mathrm{}</code>	$\square + \mathrm{}$	(t)ext (r)oman
tb	<code>\mathbf{}</code>	$\square + \mathbf{}$	(t)ext (b)old
ti	<code>\mathit{}</code>	$\square + \mathit{}$	(t)ext (i)talics

## 7.3 Misc.

Table 26:

key	trans	sym	description
binom	<code>\binom{ }{ }</code>	$\left(\begin{smallmatrix} \square \\ \square \end{smallmatrix}\right)$	Binom
box	<code>\boxed{ }</code>	$\boxed{\square}$	Putting box around object
fr	<code>\frac{ }{ }</code>	$\frac{\square}{\square}$	Fractions
can	<code>\cancel{ }</code>	$\cancel{\square}$	
&=	<code>&amp;=\n\\</code>		
=&	<code>=&amp;\n\\</code>		

## 7.4 xy Diagram related

Table 27:

key	trans	sym	description
xy	<code>\xymatrix{ \n \n }</code>		
bu	<code>\bullet</code>	•	
ar	<code>\ar</code>		

## 8 Formatting Table into Elisp

```
def format_table_to_elisp_type6col(headcomment, table):
    print(f";; {headcomment}")
    table = table[1:]
    for line in table:
        key, sym, trans, key1, sym, trans1 = line
        key = repr(key).replace("\\'", "'").replace("~", "").replace("\\\\\\texttt{\\\\\\")
        key1 = repr(key1).replace("\\'", "'").replace("~", "").replace("\\\\\\texttt{\\\\\\")
```

```

trans = repr(trans).replace("\'", "\"").replace("~", "")
trans1 = repr(trans1).replace("\'", "\"").replace("~", "")

print(f"({key:<7}  [{trans:<17}])  ({key1:<7}  [{trans1:<17}])")

def format_table_to_elisp_type3col_type1(headcomment, table):
    print(f";; {headcomment}")
    table = table[1:]
    for line in table:
        key, trans, sym, description = line
        key = repr(key).replace("\'", "\"").replace("~", "").replace("\\\\texttt{\\\\"
        trans = repr(trans).replace("\'", "\"").replace("~", "")

        print(f"({key:<8}  [{trans:<22}])  ; {description}")

def format_table_to_elisp_type3col_type2(headcomment, table):
    print(f";; {headcomment}")
    table = table[1:]
    for line in table:
        key, trans, sym, description = line
        key = repr(key).replace("\'", "\"").replace("~", "").replace("\\\\texttt{\\\\"
        trans = trans.replace("~", "")

        print(f"({key:<8}  {trans:<22})  ; {description}")

format_table_to_elisp_type6col("Greek", tbl_1_greek)
format_table_to_elisp_type6col("Matrix", tbl_1_matrix)
format_table_to_elisp_type6col("Vector & Hat", tbl_1_vec)
format_table_to_elisp_type6col("Dot", tbl_alphabet_dot_6column)
format_table_to_elisp_type6col("DDot", tbl_alphabet_ddot_6column)

format_table_to_elisp_type3col_type2("Expanding Func", tbl2_exec_func)

format_table_to_elisp_type3col_type1("Symbols-dots", tbl_3_sym_dots)
format_table_to_elisp_type3col_type1("Symbols-geo", tbl_3_sym_geo)
format_table_to_elisp_type3col_type1("Symbols", tbl_3_sym_letter)
format_table_to_elisp_type3col_type1("Symbols spaces", tbl_3_sym_spc)
format_table_to_elisp_type3col_type1("Symbols arrow1", tbl_3_sym_arrow_1)
format_table_to_elisp_type3col_type1("Symbols arrow2", tbl_3_sym_arrow_2)
format_table_to_elisp_type3col_type1("Symbols arrow3", tbl_3_sym_arrow_3)

```

```

format_table_to_elisp_type3col_type1("Symbols arrow3", tbl_4_sym_mod_1)

format_table_to_elisp_type3col_type1("Operation: arith", tbl_5_op_arith)
format_table_to_elisp_type3col_type1("Operation: arith", tbl_5_op_bin)
format_table_to_elisp_type3col_type1("Operation: arith", tbl_5_op_set)
format_table_to_elisp_type3col_type1("Operation: arith", tbl_5_op_logic)

format_table_to_elisp_type3col_type1("Func: main", tbl_6_func)
format_table_to_elisp_type6col("Func: Trig", tbl_6_func_trig_6col)
format_table_to_elisp_type3col_type1("Func: iter", tbl_6_func_iter)

format_table_to_elisp_type3col_type1("Structural: Parenthesis", tbl_7_parenthesis)
format_table_to_elisp_type3col_type1("Structural: Text", tbl_7_text)
format_table_to_elisp_type3col_type1("Structural: Text", tbl_7_text)
format_table_to_elisp_type6col("Structural: Sub-sup-scripts", tbl_7_supsubscripts)
format_table_to_elisp_type3col_type1("Structural: misc", tbl_7_misc)
format_table_to_elisp_type3col_type1("Structural: xy", tbl_7_xy)

;; Greek
("a." ["\\alpha" ]) ("A." ["A" ])
("b." ["\\beta" ]) ("B." ["B" ])
("c." ["\\psi" ]) ("C." ["\\Psi" ])
("d." ["\\delta" ]) ("D." ["\\Delta" ])
("e." ["\\epsilon" ]) ("E." ["E" ])
("f." ["\\phi" ]) ("F." ["\\Phi" ])
("g." ["\\gamma" ]) ("G." ["\\Gamma" ])
("h." ["\\eta" ]) ("H." ["H" ])
("i." ["\\iota" ]) ("I." ["I" ])
("j." ["\\xi" ]) ("J." ["\\Xi" ])
("k." ["\\kappa" ]) ("K." ["K" ])
("l." ["\\lambda" ]) ("L." ["\\Lambda" ])
("m." ["\\mu" ]) ("M." ["M" ])
("n." ["\\nu" ]) ("N." ["N" ])
("o." ["o" ]) ("O." ["O" ])
("p." ["\\pi" ]) ("P." ["\\Pi" ])
("r." ["\\rho" ]) ("R." ["p" ])
("s." ["\\sigma" ]) ("S." ["\\Sigma" ])
("t." ["\\tau" ]) ("T." ["T" ])
("th." ["\\theta" ]) ("Th." ["\\Theta" ])

```

```

("u." ["\\epsilon" ]) ("U." ["\\Upsilon" ])
("w." ["\\omega" ]) ("W." ["\\Omega" ])
("x." ["\\chi" ]) ("X." ["X" ])
("z." ["\\zeta" ]) ("Z." ["Z" ])
;; Matrix
("Am" ["\\mathbf{A}" ]) ("am" ["\\mathbf{a}" ])
("Bm" ["\\mathbf{B}" ]) ("bm" ["\\mathbf{b}" ])
("Cm" ["\\mathbf{C}" ]) ("cm" ["\\mathbf{c}" ])
("Dm" ["\\mathbf{D}" ]) ("dm" ["\\mathbf{d}" ])
("Em" ["\\mathbf{E}" ]) ("em" ["\\mathbf{e}" ])
("Fm" ["\\mathbf{F}" ]) ("fm" ["\\mathbf{f}" ])
("Gm" ["\\mathbf{G}" ]) ("gm" ["\\mathbf{g}" ])
("Hm" ["\\mathbf{H}" ]) ("hm" ["\\mathbf{h}" ])
("Im" ["\\mathbf{I}" ]) ("im" ["\\mathbf{i}" ])
("Jm" ["\\mathbf{J}" ]) ("jm" ["\\mathbf{j}" ])
("Km" ["\\mathbf{K}" ]) ("km" ["\\mathbf{k}" ])
("Lm" ["\\mathbf{L}" ]) ("lm" ["\\mathbf{l}" ])
("Mm" ["\\mathbf{M}" ]) ("mm" ["\\mathbf{m}" ])
("Nm" ["\\mathbf{N}" ]) ("nm" ["\\mathbf{n}" ])
("Om" ["\\mathbf{O}" ]) ("om" ["\\mathbf{o}" ])
("Pm" ["\\mathbf{P}" ]) ("pm" ["\\mathbf{p}" ])
("Qm" ["\\mathbf{Q}" ]) ("qm" ["\\mathbf{q}" ])
("Rm" ["\\mathbf{R}" ]) ("rm" ["\\mathbf{r}" ])
("Sm" ["\\mathbf{S}" ]) ("sm" ["\\mathbf{s}" ])
("Tm" ["\\mathbf{T}" ]) ("tm" ["\\mathbf{t}" ])
("Um" ["\\mathbf{U}" ]) ("um" ["\\mathbf{u}" ])
("Vm" ["\\mathbf{V}" ]) ("vm" ["\\mathbf{v}" ])
("Wm" ["\\mathbf{W}" ]) ("wm" ["\\mathbf{w}" ])
("Xm" ["\\mathbf{X}" ]) ("xm" ["\\mathbf{x}" ])
("Ym" ["\\mathbf{Y}" ]) ("ym" ["\\mathbf{y}" ])
("Zm" ["\\mathbf{Z}" ]) ("zm" ["\\mathbf{z}" ])
("Om" ["\\mathbf{0}" ]) ("Om" ["\\mathbf{0}" ])
;; Vector & Hat
("av" ["\\vec{a}" ]) ("ah" ["\\hat{a}" ])
("bv" ["\\vec{b}" ]) ("bh" ["\\hat{b}" ])
("cv" ["\\vec{c}" ]) ("ch" ["\\hat{c}" ])
("dv" ["\\vec{d}" ]) ("dh" ["\\hat{d}" ])
("ev" ["\\vec{e}" ]) ("eh" ["\\hat{e}" ])
("fv" ["\\vec{f}" ]) ("fh" ["\\hat{f}" ])
("gv" ["\\vec{g}" ]) ("gh" ["\\hat{g}" ])

```

("hv"	["\\vec{h}"	)	("hh"	["\\hat{h}"	)
("iv"	["\\vec{i}"	)	("ih"	["\\hat{i}"	)
("jv"	["\\vec{j}"	)	("jh"	["\\hat{j}"	)
("kv"	["\\vec{k}"	)	("kh"	["\\hat{k}"	)
("lv"	["\\vec{l}"	)	("lh"	["\\hat{l}"	)
("mv"	["\\vec{m}"	)	("mh"	["\\hat{m}"	)
("nv"	["\\vec{n}"	)	("nh"	["\\hat{n}"	)
("ov"	["\\vec{o}"	)	("oh"	["\\hat{o}"	)
("pv"	["\\vec{p}"	)	("ph"	["\\hat{p}"	)
("qv"	["\\vec{q}"	)	("qh"	["\\hat{q}"	)
("rv"	["\\vec{r}"	)	("rh"	["\\hat{r}"	)
("sv"	["\\vec{s}"	)	("sh"	["\\hat{s}"	)
("tv"	["\\vec{t}"	)	("th"	["\\hat{t}"	)
("uv"	["\\vec{u}"	)	("uh"	["\\hat{u}"	)
("vv"	["\\vec{v}"	)	("vh"	["\\hat{v}"	)
("wv"	["\\vec{w}"	)	("wh"	["\\hat{w}"	)
("xv"	["\\vec{x}"	)	("xh"	["\\hat{x}"	)
("yv"	["\\vec{y}"	)	("yh"	["\\hat{y}"	)
("zv"	["\\vec{z}"	)	("zh"	["\\hat{z}"	)
;; Dot					
("ad"	["\\dot{a}"	)	("Ad"	["\\dot{A}"	)
("bd"	["\\dot{b}"	)	("Bd"	["\\dot{B}"	)
("cd"	["\\dot{c}"	)	("Cd"	["\\dot{C}"	)
("dd"	["\\dot{d}"	)	("Dd"	["\\dot{D}"	)
("ed"	["\\dot{e}"	)	("Ed"	["\\dot{E}"	)
("fd"	["\\dot{f}"	)	("Fd"	["\\dot{F}"	)
("gd"	["\\dot{g}"	)	("Gd"	["\\dot{G}"	)
("hd"	["\\dot{h}"	)	("Hd"	["\\dot{H}"	)
("id"	["\\dot{i}"	)	("Id"	["\\dot{I}"	)
("jd"	["\\dot{j}"	)	("Jd"	["\\dot{J}"	)
("kd"	["\\dot{k}"	)	("Kd"	["\\dot{K}"	)
("ld"	["\\dot{l}"	)	("Ld"	["\\dot{L}"	)
("md"	["\\dot{m}"	)	("Md"	["\\dot{M}"	)
("nd"	["\\dot{n}"	)	("Nd"	["\\dot{N}"	)
("od"	["\\dot{o}"	)	("Od"	["\\dot{O}"	)
("pd"	["\\dot{p}"	)	("Pd"	["\\dot{P}"	)
("qd"	["\\dot{q}"	)	("Qd"	["\\dot{Q}"	)
("rd"	["\\dot{r}"	)	("Rd"	["\\dot{R}"	)
("sd"	["\\dot{s}"	)	("Sd"	["\\dot{S}"	)
("td"	["\\dot{t}"	)	("Td"	["\\dot{T}"	)



```

("ud"      ["\\dot{u}"      ]) ("Ud"      ["\\dot{U}"      ])
("vd"      ["\\dot{v}"      ]) ("Vd"      ["\\dot{V}"      ])
("wd"      ["\\dot{w}"      ]) ("Wd"      ["\\dot{W}"      ])
("xd"      ["\\dot{x}"      ]) ("Xd"      ["\\dot{X}"      ])
("yd"      ["\\dot{y}"      ]) ("Yd"      ["\\dot{Y}"      ])
("zd"      ["\\dot{z}"      ]) ("Zd"      ["\\dot{Z}"      ])
;; DDot
("ad."     ["\\ddot{a}"     ]) ("Ad."     ["\\ddot{A}"     ])
("bd."     ["\\ddot{b}"     ]) ("Bd."     ["\\ddot{B}"     ])
("cd."     ["\\ddot{c}"     ]) ("Cd."     ["\\ddot{C}"     ])
("dd."     ["\\ddot{d}"     ]) ("Dd."     ["\\ddot{D}"     ])
("ed."     ["\\ddot{e}"     ]) ("Ed."     ["\\ddot{E}"     ])
("fd."     ["\\ddot{f}"     ]) ("Fd."     ["\\ddot{F}"     ])
("gd."     ["\\ddot{g}"     ]) ("Gd."     ["\\ddot{G}"     ])
("hd."     ["\\ddot{h}"     ]) ("Hd."     ["\\ddot{H}"     ])
("id."     ["\\ddot{i}"     ]) ("Id."     ["\\ddot{I}"     ])
("jd."     ["\\ddot{j}"     ]) ("Jd."     ["\\ddot{J}"     ])
("kd."     ["\\ddot{k}"     ]) ("Kd."     ["\\ddot{K}"     ])
("ld."     ["\\ddot{l}"     ]) ("Ld."     ["\\ddot{L}"     ])
("md."     ["\\ddot{m}"     ]) ("Md."     ["\\ddot{M}"     ])
("nd."     ["\\ddot{n}"     ]) ("Nd."     ["\\ddot{N}"     ])
("od."     ["\\ddot{o}"     ]) ("Od."     ["\\ddot{O}"     ])
("pd."     ["\\ddot{p}"     ]) ("Pd."     ["\\ddot{P}"     ])
("qd."     ["\\ddot{q}"     ]) ("Qd."     ["\\ddot{Q}"     ])
("rd."     ["\\ddot{r}"     ]) ("Rd."     ["\\ddot{R}"     ])
("sd."     ["\\ddot{s}"     ]) ("Sd."     ["\\ddot{S}"     ])
("td."     ["\\ddot{t}"     ]) ("Td."     ["\\ddot{T}"     ])
("ud."     ["\\ddot{u}"     ]) ("Ud."     ["\\ddot{U}"     ])
("vd."     ["\\ddot{v}"     ]) ("Vd."     ["\\ddot{V}"     ])
("wd."     ["\\ddot{w}"     ]) ("Wd."     ["\\ddot{W}"     ])
("xd."     ["\\ddot{x}"     ]) ("Xd."     ["\\ddot{X}"     ])
("yd."     ["\\ddot{y}"     ]) ("Yd."     ["\\ddot{Y}"     ])
("zd."     ["\\ddot{z}"     ]) ("Zd."     ["\\ddot{Z}"     ])
;; Expanding Func
("/")      quail-TeX-fraction      ) ; fraction on previous
("eq"      quail-TeX-equation      ) ; equation environment
("al"      quail-TeX-aligned       ) ; aligned environment
("el"      quail-TeX-endofline     ) ; end of line
;; Symbols-dots
("..."   ["\\dots"              ]) ; 3 dots

```

```

("v"      ["\\vdots"      ]) ; vertical dots
("d"      ["\\ddots"      ]) ; diagonale dots
("l"      ["\\ldots"      ]) ; low dots
;; Symbols-geo
("perp"   ["\\perp"       ]) ;
("perpn"  ["\\perp"       ]) ; $\\perp$ ~n~ (neg)
("para"   ["\\parallel"   ]) ;
("paran"  ["\\nparallel"    ]) ; $\\parallel$ ~n~ (neg)
("ang"    ["\\angle"      ]) ;
("ang."   ["\\measuredangle" ]) ; $\\angle$ ~.~ (var)
("tri"    ["\\vartriangle"  ]) ;
("trin"   ["\\triangledown" ]) ; $\\vartriangle$ ~n~ (neg)
("squ"    ["\\square"      ]) ;
("tri."   ["\\blacktriangle"  ]) ; $\\vartriangle$ ~.~ (var)
("trin."  ["\\blacktriangledown"]) ; $\\vartriangle$ ~n.~ (neg,var)
("squ."   ["\\blacksquare"   ]) ; $\\square$ ~.~ (var)
;; Symbols
("inf"    ["\\infty"        ]) ;
("ex"     ["\\exists"      ]) ;
("exn"    ["\\nexists"     ]) ; $\\exists$ + _n_ (neg)
("fa"     ["\\forall"      ]) ;
("hb"     ["\\hbar"        ]) ;
("hb."    ["\\hslash"       ]) ; $\\hbar$ + _._ (var)
("dd"     ["\\mathrm{d}"   ]) ;
("dd."    ["\\partial"     ]) ; $\\mathrm{d}$ + _._ (var)
("ii"     ["\\imath"       ]) ;
("jj"     ["\\jmath"       ]) ;
("nab"    ["\\nabla"       ]) ;
("cm"     ["\\checkmark"   ]) ;
;; Symbols spaces
("qu"     ["\\quad"        ]) ;
("quu"    ["\\qquad"       ]) ;
;; Symbols arrow1
("<->"   ["\\leftarrow"         ]) ;
("<->"   ["\\rightarrow"        ]) ;
("<-~"    ["\\uparrow"          ]) ; ~~~ looks like up arrow head
("<-v"    ["\\downarrow"         ]) ; ~v~ looks like down arrow head
("<->"    ["\\leftrightharpoon" ]) ;
("<-n"    ["\\nleftarrow"        ]) ; negate (~n~) of prev. section
("<->n"   ["\\nrightarrow"       ]) ; arrows + _n_

```

```

("-^n"      ["\\nuparrow"      ]) ;
("-vn"      ["\\ndownarrow"    ]) ;
("<->"      ["\\leftrightharpoon"] ) ;
("-->"      ["\\longrightrightarrow"] ) ; longer arrows, with 2 dashes
("<--"      ["\\longleftarrow"  ]) ;
("|->"      ["\\mapsto"          ]) ; vertical-bar + ~->~
(""         [""]              ]) ; (this rendered wrong on Github)
(""         [""]              ]) ; (check from PDF file)
;; Symbols arrow2
("<="       ["\\Leftarrow"      ]) ; compared to single arrow
(">="       ["\\Rightarrow"     ]) ; these uses ~~= as the arrow shaft
("="^       ["\\Upward"        ]) ;
("="v       ["\\Downward"      ]) ;
("<=>"     ["\\Leftrightarrow"    ]) ;
("iff"      ["\\Leftrightarrow"    ]) ;
("<=n"     ["\\nLeftarrow"       ]) ; negate (~n~) of prev. section
(">=n"     ["\\nRightarrow"    ]) ; arrows + _n_
("<=>n"    ["\\nLeftrightarrow"  ]) ;
("iffn"     ["\\nLeftrightarrow"    ]) ;
("<==>"    ["\\Longleftarrow"     ]) ; longer arrows, with 2 dashes
("<=="     ["\\Longleftarrow"  ]) ;
(">==>"    ["\\Longrightrightarrow"] ) ;
;; Symbols arrow3
("<---"     ["\\xleftarrow[ ]{ }"    ]) ; these uses triple - or =
("<--->"   ["\\xrightarrow[ ]{ }"] ) ;
("===>"    ["\\xRightarrow[ ]{ }"] ) ; ~mathtools~ lib required
("<===>"    ["\\xLeftrightarrow[ ]{ }"] ) ; ~mathtools~ lib required
;; Symbols arrow3
("vec"      ["\\vec{"            ]) ;
("bar"      ["\\bar{"          ]) ;
("hat"      ["\\hat{"        ]) ;
("dot"      ["\\dot{"        ]) ;
("dot."     ["\\ddot{"         ]) ;
("dot.."    ["\\dddotted{"    ]) ;
("dot..." ["\\ddddotted{"     ]) ;
("dag"      ["^\\dagger"     ]) ;
("dag."     ["^\\ddagger"      ]) ;
("*.."      ["^*"              ]) ;
("deg"      ["^\\circ"         ]) ;
("tr"       ["^T"           ]) ;

```

```

("tr."      ["^{\-T}"          ]) ;
;; Operation: arith
("+-"       ["\\pm"            ]) ;
("-+"       ["\\mp"            ]) ;
("*x"       ["\\times"         ]) ;
("::"       ["\\div"           ]) ;
("**"       ["\\cdot"          ]) ;
;; Operation: arith
("=. "      ["\\equiv"         ]) ; Variation on
("<."       ["\\leq"            ]) ; < = >
(">."       ["\\geq"            ]) ; symbols
("<<"       ["\\ll"             ]) ;
(">>"       ["\\gg"             ]) ;
("=n"       ["\\neq"            ]) ; negation
("n"        ["\\nsim"           ]) ;
("<n"       ["\\nless"          ]) ;
(">n"       ["\\ngtr"           ]) ;
("<.n"      ["\\nleq"            ]) ;
(">.n"      ["\\ngeq"            ]) ;
("=?"       ["\\stackrel{?}{=}"  ]) ; with question mark
("<?"       ["\\stackrel{?}{<}"  ]) ;
(">?"       ["\\stackrel{?}{>}"  ]) ;
("<?.?"     ["\\stackrel{?}{\\leq}"]) ;
(">?.?"     ["\\stackrel{?}{\\geq}"]) ;
("<<?"      ["\\stackrel{?}{\\ll}"] ) ;
(">>?"      ["\\stackrel{?}{\\gg}"] ) ;
("=y"       ["\\stackrel{\\checkmark}{=}"]) ; with check mark
("<y"       ["\\stackrel{\\checkmark}{<}"]) ;
(">y"       ["\\stackrel{\\checkmark}{>}"]) ;
("<.y"      ["\\stackrel{\\checkmark}{\\leq}"]) ;
(">.y"      ["\\stackrel{\\checkmark}{\\geq}"]) ;
("<<y"      ["\\stackrel{\\checkmark}{\\ll}"]) ;
(">>y"      ["\\stackrel{\\checkmark}{\\gg}"]) ;
(". "       ["\\sim"              ]) ; Another variations
(""        ["\\approx"           ]) ; on =
("3="      ["\\equiv"           ]) ;
("=: "     ["\\coloneqq"         ]) ;
(":= "     ["\\coloneqq"         ]) ;
;; Operation: arith
("in"      ["\\in"              ]) ;

```

```

("in."      ["\\ni"          ]) ;
("ni"       ["\\ni"          ]) ;
("inn"      ["\\notin"       ]) ; (neg)
("0/"       ["\\emptyset"     ]) ;
("nsr"      ["\\mathbb{R}"     ]) ; (n)umber (s)et (r)eal
("nsc"      ["\\mathbb{C}"     ]) ; (n)umber (s)et (c)omplex
("nsn"      ["\\mathbb{N}"     ]) ; ...
("nsp"      ["\\mathbb{P}"     ]) ; ...
("nsz"      ["\\mathbb{Z}"     ]) ; ...
("nsi"      ["\\mathbb{I}"     ]) ; ...
("sub"      ["\\subset"       ]) ;
("subn"     ["\\nssubseteq"    ]) ; (neg)
("sub."     ["\\subseq"       ]) ; (var)
("sub.n"    ["\\nsubseq"      ]) ; (var, neg)
("subn."    ["\\nsubseq"      ]) ; (neg, var)
("sup"      ["\\supset"       ]) ;
("supn"     ["\\nsupseteq"    ]) ; (neg)
("sup."     ["\\supeseteq"    ]) ; (var)
("sup.n"    ["\\nsupseteq"    ]) ; (var, neg)
("supn."    ["\\nsupseteq"    ]) ; (neg, var)
;; Operation: arith
("or"       ["\\lor"         ]) ;
("and"      ["\\land"        ]) ;
("not"      ["\\neg"         ]) ;
("or."      ["\\text{ or }"   ]) ; (var)
("and."     ["\\text{ and }"  ]) ; (var)
("not."     ["\\text{ not }"  ]) ; (var)
;; Func: main
("rank"     ["\\mathrm{rank}"  ]) ;
("arg"      ["\\arg"          ]) ;
("det"      ["\\det"         ]) ;
("dim"      ["\\dim"         ]) ;
("exp"      ["\\exp("        ]) ;
("Im"       ["\\mathrm{Im}("  ]) ;
("Re"       ["\\mathrm{Re}("  ]) ;
("ln"       ["\\ln("         ]) ;
("log"      ["\\log("        ]) ;
("max"      ["\\max("        ]) ;
("min"      ["\\min("        ]) ;
("dim"      ["\\dim("        ]) ;

```

```

("sqrt"      ["\\sqrt("           ]) ;
("mod"       ["\\pmod("           ]) ;
("mod."      ["\\mod"             ]) ;
("mod.."     ["\\bmod"            ]) ;
;; Func: Trig
("cos"       ["\\cos("            ]) ("cosh"    ["\\cosh("            ])
("sin"       ["\\sin("            ]) ("sinh"    ["\\sinh("            ])
("tan"       ["\\tan("            ]) ("tanh"    ["\\tanh("            ])
("cot"       ["\\cot("            ]) ("coth"    ["\\coth("            ])
("acos"      ["\\arccos("         ]) ("cos."    ["\\arccos("         ])
("asin"      ["\\arcsin("         ]) ("sin."    ["\\arcsin("         ])
("atan"      ["\\arctan("         ]) ("tan."    ["\\arctan("         ])
;; Func: iter
("il"        ["\\limits_{ }"       ]) ;
("il"        ["\\limits_{ }^{ }"    ]) ;
("lim"       ["\\lim"              ]) ;
("sum"       ["\\sum"              ]) ;
("prod"      ["\\prod"             ]) ;
("int"       ["\\int"              ]) ;
("inti"      ["\\iint"             ]) ;
("intii"     ["\\iiint"            ]) ;
("intiii"    ["\\iiint"            ]) ;
("into"      ["\\oint"             ]) ;
("sum."      ["\\sum\\limits_{ i=1 }^{ n }"]) ;
("prod."     ["\\prod\\limits_{ i=1 }^{ n }"]) ;
("int."      ["\\int\\limits_{ }^{ }"]) ;
("int.."     ["\\int\\limits_{ 0 }^{ +\\infty }"]) ;
("int..." ["\\int\\limits_{ -\\infty }^{ +\\infty }"]) ;
("inti."     ["\\iint\\limits_{ }"]) ;
("intii."    ["\\iiint\\limits_{ }"]) ;
("intiii."   ["\\iiint\\limits_{ }"]) ;
("into."     ["\\oint\\limits_{ }"]) ;
;; Structural: Parenthesis
("(".."      ["\\left( \\right)"      ]) ;
("(".."      ["\\left( \\middle\\vert \\right)"]) ;
("[]."       ["\\left[ \\right]"           ]) ;
("[]."       ["\\left[ \\middle\\vert \\right)"]) ; (var)
("[].c"      ["\\lceil \\rceil"         ]) ; (var) (ceil)
("[].f"      ["\\lfloor \\rfloor"    ]) ; (var) (floor)
("{ }."      ["\\left\\{ \\right\\}"]) ;

```

```

("{}.." ["\\left\\{ \\middle\\vert \\right\\}"]) ; (var)
("<>." ["\\left< \\right>" ]) ;
("<>.." ["\\left< \\middle\\vert \\right>"]) ; (var)
("||.." ["\\left\\vert \\right\\vert"]) ;
("||.." ["\\left\\Vert \\right\\Vert"]) ; (var)
("(. ." ["\\left(" ]) ; half (
("). ." ["\\right)" ]) ; half )
("[. ." ["\\left[" ]) ; half [
("]. ." ["\\right]" ]) ; half ]
("{. ." ["\\left\\{" ]) ; half {
("}. ." ["\\right\\}" ]) ; half }
("<." ["\\left<" ]) ; half <
(">." ["\\right>" ]) ; half >
("(. ." ["\\left." ]) ; half left .
("). ." ["\\right." ]) ; half right .
("|. ." ["\\Bigg\\vert_{ }^{ }"]) ; definite integral range
;; Structural: Text
("te" ["\\text{" ]) ; (te)xt
("tr" ["\\mathrm{" ]) ; (t)ext (r)oman
("tb" ["\\mathbf{" ]) ; (t)ext (b)old
("ti" ["\\mathit{" ]) ; (t)ext (i)talics
;; Structural: Text
("te" ["\\text{" ]) ; (te)xt
("tr" ["\\mathrm{" ]) ; (t)ext (r)oman
("tb" ["\\mathbf{" ]) ; (t)ext (b)old
("ti" ["\\mathit{" ]) ; (t)ext (i)talics
;; Structural: Sub-sup-scripts
("^" ["^{ " ]) ("_" ["_{ " ])
("pp" ["^{ " ]) ("l1" ["_1 " ])
("p0" ["^0 " ]) ("l0" ["_0 " ])
("p1" ["^1 " ]) ("l1" ["_1 " ])
("p2" ["^2 " ]) ("l2" ["_2 " ])
("p3" ["^3 " ]) ("l3" ["_3 " ])
("p4" ["^4 " ]) ("l4" ["_4 " ])
("pn" ["^n " ]) ("lnn" ["_n " ])
("px" ["^x " ]) ("li" ["_i " ])
("__" ["\\underset{ }{ }"]) ("^^" ["\\overset{ }{ }"])
("__." ["\\underbrace{ }_{ }"]) ("^^." ["\\overbrace{ }^{ }"])
("___." ["\\underline{ } " ]) ("^^.." ["\\overline{ } " ])
;; Structural: misc

```

```

("binom"  ["\\binom{}{"      ]) ; Binom
("box"    ["\\boxed{}{"      ]) ; Putting box around object
("fr"     ["\\frac{}{"      ]) ; Fractions
("can"    ["\\cancel"        ]) ;
("&="     ["&=\\n\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\"]]) ;
("=&"     ["&=\\n\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\"]]) ;
;; Structural: xy
("xy"     ["\\xymatrix{\\n\\n}" ]) ;
("bu"     ["\\bullet"         ]) ;
("ar"     ["\\ar"             ]) ;

```

## 9 Executable elisp function definition

```

(defun quail-func-init ()
  (quail-delete-region)
  (setq quail-current-str nil
        quail-converting nil
        quail-conversion-str ""))

(defun quail-func-end ()
  (throw 'quail-tag nil))

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
(defun quail-TeX-equation (key idx)
  (quail-func-init)
  (insert "\\begin{equation}\\n\\n\\end{equation}")
  (previous-line)
  (quail-func-end))

(defun quail-TeX-aligned (key idx)
  (quail-func-init)
  (insert "\\begin{aligned}\\n\\n\\end{aligned}")
  (previous-line)
  (quail-func-end))

(defun quail-TeX-endofline (key idx)
  (quail-func-init)
  (end-of-line)
  (insert "\\\\n"))

```



```

(quail-func-end))

(defun quail-TeX-func (key idx)
  (quail-func-init)

  (backward-sexp) (kill-sexp)
  (if (looking-back "[a-zA-Z]" 0)
      (progn
        (backward-word)
        (if (= (preceding-char) ?\\ )
            (progn (message "yes") (kill-word 1)
                  (backward-delete-char 1) (insert "\\frac{\\")
                  (yank 1) (yank 2) (insert "}{"))
            (progn (message "no") (forward-word)
                  (insert "\\frac{" (yank) (insert "}{"))))
        )
      (progn (message "no")
              ;(forward-word)
              (insert "\\frac{" (yank) (insert "}{"))
        )
    (backward-char)

  (quail-func-end))

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

```

## 10 Making the el

```

(require 'quail)

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

(defun quail-func-init ()
  (quail-delete-region)
  (setq quail-current-str nil
        quail-converting nil
        quail-conversion-str ""))

(defun quail-func-end ()

```

```

(throw 'quail-tag nil))

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
(defun quail-TeX-equation (key idx)
  (quail-func-init)
  (insert "\\begin{equation}\n\n\\end{equation}")
  (previous-line)
  (quail-func-end))

(defun quail-TeX-aligned (key idx)
  (quail-func-init)
  (insert "\\begin{aligned}\n\n\\end{aligned}")
  (previous-line)
  (quail-func-end))

(defun quail-TeX-endofline (key idx)
  (quail-func-init)
  (end-of-line)
  (insert "\\\n")
  (quail-func-end))

(defun quail-TeX-frac (key idx)
  (quail-func-init)

  (backward-sexp) (kill-sexp)
  (if (looking-back "[a-zA-Z]" 0)
      (progn
        (backward-word)
        (if (= (preceding-char) ?\\ )
            (progn (message "yes") (kill-word 1)
                  (backward-delete-char 1) (insert "\\frac{\\")
                  (yank 1) (yank 2) (insert "}{"))
            (progn (message "no") (forward-word)
                  (insert "\\frac{" (yank) (insert "}{"))))
        )
      (progn (message "no")
              ;(forward-word)
              (insert "\\frac{" (yank) (insert "}{"))
        )
    (backward-char)

```

```

(quail-func-end))

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

(quail-define-package
  "TeX-Math" "Emacs-TeX-Latex" "TeX-" t
  "TeX-Math input"
  nil t t t t nil nil nil nil nil t)

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

(quail-define-rules
  ;; Greek Alphabets
  ;; Greek
  ("a." ["\\alpha" ]) ("A." ["A" ])
  ("b." ["\\beta" ]) ("B." ["B" ])
  ("c." ["\\psi" ]) ("C." ["\\Psi" ])
  ("d." ["\\delta" ]) ("D." ["\\Delta" ])
  ("e." ["\\epsilon" ]) ("E." ["E" ])
  ("f." ["\\phi" ]) ("F." ["\\Phi" ])
  ("g." ["\\gamma" ]) ("G." ["\\Gamma" ])
  ("h." ["\\eta" ]) ("H." ["H" ])
  ("i." ["\\iota" ]) ("I." ["I" ])
  ("j." ["\\xi" ]) ("J." ["\\Xi" ])
  ("k." ["\\kappa" ]) ("K." ["K" ])
  ("l." ["\\lambda" ]) ("L." ["\\Lambda" ])
  ("m." ["\\mu" ]) ("M." ["M" ])
  ("n." ["\\nu" ]) ("N." ["N" ])
  ("o." ["o" ]) ("O." ["O" ])
  ("p." ["\\pi" ]) ("P." ["\\Pi" ])
  ("r." ["\\rho" ]) ("R." ["P" ])
  ("s." ["\\sigma" ]) ("S." ["\\Sigma" ])
  ("t." ["\\tau" ]) ("T." ["T" ])
  ("th." ["\\theta" ]) ("Th." ["\\Theta" ])
  ("u." ["\\upsilon" ]) ("U." ["\\Upsilon" ])
  ("w." ["\\omega" ]) ("W." ["\\Omega" ])
  ("x." ["\\chi" ]) ("X." ["X" ])
  ("z." ["\\zeta" ]) ("Z." ["Z" ])
  ;; Matrix

```

("Am"	["\\mathbf{A}"	])	("am"	["\\mathbf{a}"	])
("Bm"	["\\mathbf{B}"	])	("bm"	["\\mathbf{b}"	])
("Cm"	["\\mathbf{C}"	])	("cm"	["\\mathbf{c}"	])
("Dm"	["\\mathbf{D}"	])	("dm"	["\\mathbf{d}"	])
("Em"	["\\mathbf{E}"	])	("em"	["\\mathbf{e}"	])
("Fm"	["\\mathbf{F}"	])	("fm"	["\\mathbf{f}"	])
("Gm"	["\\mathbf{G}"	])	("gm"	["\\mathbf{g}"	])
("Hm"	["\\mathbf{H}"	])	("hm"	["\\mathbf{h}"	])
("Im"	["\\mathbf{I}"	])	("im"	["\\mathbf{i}"	])
("Jm"	["\\mathbf{J}"	])	("jm"	["\\mathbf{j}"	])
("Km"	["\\mathbf{K}"	])	("km"	["\\mathbf{k}"	])
("Lm"	["\\mathbf{L}"	])	("lm"	["\\mathbf{l}"	])
("Mm"	["\\mathbf{M}"	])	("mm"	["\\mathbf{m}"	])
("Nm"	["\\mathbf{N}"	])	("nm"	["\\mathbf{n}"	])
("Om"	["\\mathbf{O}"	])	("om"	["\\mathbf{o}"	])
("Pm"	["\\mathbf{P}"	])	("pm"	["\\mathbf{p}"	])
("Qm"	["\\mathbf{Q}"	])	("qm"	["\\mathbf{q}"	])
("Rm"	["\\mathbf{R}"	])	("rm"	["\\mathbf{r}"	])
("Sm"	["\\mathbf{S}"	])	("sm"	["\\mathbf{s}"	])
("Tm"	["\\mathbf{T}"	])	("tm"	["\\mathbf{t}"	])
("Um"	["\\mathbf{U}"	])	("um"	["\\mathbf{u}"	])
("Vm"	["\\mathbf{V}"	])	("vm"	["\\mathbf{v}"	])
("Wm"	["\\mathbf{W}"	])	("wm"	["\\mathbf{w}"	])
("Xm"	["\\mathbf{X}"	])	("xm"	["\\mathbf{x}"	])
("Ym"	["\\mathbf{Y}"	])	("ym"	["\\mathbf{y}"	])
("Zm"	["\\mathbf{Z}"	])	("zm"	["\\mathbf{z}"	])
("Om"	["\\mathbf{0}"	])	("Om"	["\\mathbf{0}"	])
;; Vector & Hat					
("av"	["\\vec{a}"	])	("ah"	["\\hat{a}"	])
("bv"	["\\vec{b}"	])	("bh"	["\\hat{b}"	])
("cv"	["\\vec{c}"	])	("ch"	["\\hat{c}"	])
("dv"	["\\vec{d}"	])	("dh"	["\\hat{d}"	])
("ev"	["\\vec{e}"	])	("eh"	["\\hat{e}"	])
("fv"	["\\vec{f}"	])	("fh"	["\\hat{f}"	])
("gv"	["\\vec{g}"	])	("gh"	["\\hat{g}"	])
("hv"	["\\vec{h}"	])	("hh"	["\\hat{h}"	])
("iv"	["\\vec{i}"	])	("ih"	["\\hat{i}"	])
("jv"	["\\vec{j}"	])	("jh"	["\\hat{j}"	])
("kv"	["\\vec{k}"	])	("kh"	["\\hat{k}"	])
("lv"	["\\vec{l}"	])	("lh"	["\\hat{l}"	])

("mv"	["\\vec{m}"	])	("mh"	["\\hat{m}"	])
("nv"	["\\vec{n}"	])	("nh"	["\\hat{n}"	])
("ov"	["\\vec{o}"	])	("oh"	["\\hat{o}"	])
("pv"	["\\vec{p}"	])	("ph"	["\\hat{p}"	])
("qv"	["\\vec{q}"	])	("qh"	["\\hat{q}"	])
("rv"	["\\vec{r}"	])	("rh"	["\\hat{r}"	])
("sv"	["\\vec{s}"	])	("sh"	["\\hat{s}"	])
("tv"	["\\vec{t}"	])	("th"	["\\hat{t}"	])
("uv"	["\\vec{u}"	])	("uh"	["\\hat{u}"	])
("vv"	["\\vec{v}"	])	("vh"	["\\hat{v}"	])
("wv"	["\\vec{w}"	])	("wh"	["\\hat{w}"	])
("xv"	["\\vec{x}"	])	("xh"	["\\hat{x}"	])
("yv"	["\\vec{y}"	])	("yh"	["\\hat{y}"	])
("zv"	["\\vec{z}"	])	("zh"	["\\hat{z}"	])
;; Dot					
("ad"	["\\dot{a}"	])	("Ad"	["\\dot{A}"	])
("bd"	["\\dot{b}"	])	("Bd"	["\\dot{B}"	])
("cd"	["\\dot{c}"	])	("Cd"	["\\dot{C}"	])
("dd"	["\\dot{d}"	])	("Dd"	["\\dot{D}"	])
("ed"	["\\dot{e}"	])	("Ed"	["\\dot{E}"	])
("fd"	["\\dot{f}"	])	("Fd"	["\\dot{F}"	])
("gd"	["\\dot{g}"	])	("Gd"	["\\dot{G}"	])
("hd"	["\\dot{h}"	])	("Hd"	["\\dot{H}"	])
("id"	["\\dot{i}"	])	("Id"	["\\dot{I}"	])
("jd"	["\\dot{j}"	])	("Jd"	["\\dot{J}"	])
("kd"	["\\dot{k}"	])	("Kd"	["\\dot{K}"	])
("ld"	["\\dot{l}"	])	("Ld"	["\\dot{L}"	])
("md"	["\\dot{m}"	])	("Md"	["\\dot{M}"	])
("nd"	["\\dot{n}"	])	("Nd"	["\\dot{N}"	])
("od"	["\\dot{o}"	])	("Od"	["\\dot{O}"	])
("pd"	["\\dot{p}"	])	("Pd"	["\\dot{P}"	])
("qd"	["\\dot{q}"	])	("Qd"	["\\dot{Q}"	])
("rd"	["\\dot{r}"	])	("Rd"	["\\dot{R}"	])
("sd"	["\\dot{s}"	])	("Sd"	["\\dot{S}"	])
("td"	["\\dot{t}"	])	("Td"	["\\dot{T}"	])
("ud"	["\\dot{u}"	])	("Ud"	["\\dot{U}"	])
("vd"	["\\dot{v}"	])	("Vd"	["\\dot{V}"	])
("wd"	["\\dot{w}"	])	("Wd"	["\\dot{W}"	])
("xd"	["\\dot{x}"	])	("Xd"	["\\dot{X}"	])
("yd"	["\\dot{y}"	])	("Yd"	["\\dot{Y}"	])

```

("zd"      ["\\dot{z}"      ]) ("Zd"      ["\\dot{Z}"      ])
;; DDot
("ad."     ["\\ddot{a}"     ]) ("Ad."     ["\\ddot{A}"     ])
("bd."     ["\\ddot{b}"     ]) ("Bd."     ["\\ddot{B}"     ])
("cd."     ["\\ddot{c}"     ]) ("Cd."     ["\\ddot{C}"     ])
("dd."     ["\\ddot{d}"     ]) ("Dd."     ["\\ddot{D}"     ])
("ed."     ["\\ddot{e}"     ]) ("Ed."     ["\\ddot{E}"     ])
("fd."     ["\\ddot{f}"     ]) ("Fd."     ["\\ddot{F}"     ])
("gd."     ["\\ddot{g}"     ]) ("Gd."     ["\\ddot{G}"     ])
("hd."     ["\\ddot{h}"     ]) ("Hd."     ["\\ddot{H}"     ])
("id."     ["\\ddot{i}"     ]) ("Id."     ["\\ddot{I}"     ])
("jd."     ["\\ddot{j}"     ]) ("Jd."     ["\\ddot{J}"     ])
("kd."     ["\\ddot{k}"     ]) ("Kd."     ["\\ddot{K}"     ])
("ld."     ["\\ddot{l}"     ]) ("Ld."     ["\\ddot{L}"     ])
("md."     ["\\ddot{m}"     ]) ("Md."     ["\\ddot{M}"     ])
("nd."     ["\\ddot{n}"     ]) ("Nd."     ["\\ddot{N}"     ])
("od."     ["\\ddot{o}"     ]) ("Od."     ["\\ddot{O}"     ])
("pd."     ["\\ddot{p}"     ]) ("Pd."     ["\\ddot{P}"     ])
("qd."     ["\\ddot{q}"     ]) ("Qd."     ["\\ddot{Q}"     ])
("rd."     ["\\ddot{r}"     ]) ("Rd."     ["\\ddot{R}"     ])
("sd."     ["\\ddot{s}"     ]) ("Sd."     ["\\ddot{S}"     ])
("td."     ["\\ddot{t}"     ]) ("Td."     ["\\ddot{T}"     ])
("ud."     ["\\ddot{u}"     ]) ("Ud."     ["\\ddot{U}"     ])
("vd."     ["\\ddot{v}"     ]) ("Vd."     ["\\ddot{V}"     ])
("wd."     ["\\ddot{w}"     ]) ("Wd."     ["\\ddot{W}"     ])
("xd."     ["\\ddot{x}"     ]) ("Xd."     ["\\ddot{X}"     ])
("yd."     ["\\ddot{y}"     ]) ("Yd."     ["\\ddot{Y}"     ])
("zd."     ["\\ddot{z}"     ]) ("Zd."     ["\\ddot{Z}"     ])
;; Expanding Func
("/")      quail-TeX-fraction ) ; fraction on previous
("eq"      quail-TeX-equation ) ; equation environment
("al"      quail-TeX-aligned  ) ; aligned environment
("el"      quail-TeX-endofline) ; end of line
;; Symbols-dots
("..."   ["\\dots"        ]) ; 3 dots
(".v"      ["\\vdots"        ]) ; vertical dots
(".d"      ["\\ddots"        ]) ; diagonale dots
(".l"      ["\\ldots"        ]) ; low dots
;; Symbols-geo
("perp"    ["\\perp"         ]) ;

```

```

("perpn" ["\\perp" ] ) ;  $\perp$   $\sim n$  (neg)
("para" ["\\parallel" ] ) ;
("paran" ["\\nparallel" ] ) ;  $\parallel$   $\sim n$  (neg)
("ang" ["\\angle" ] ) ;
("ang." ["\\measuredangle" ] ) ;  $\angle$   $\sim. (var)
("tri" ["\\vartriangle" ] ) ;
("trin" ["\\triangledown" ] ) ;  $\vartriangle$   $\sim n$  (neg)
("squ" ["\\square" ] ) ;
("tri." ["\\blacktriangle" ] ) ;  $\vartriangle$   $\sim. (var)
("trin." ["\\blacktriangledown" ] ) ;  $\vartriangle$   $\sim n. (neg,var)
("squ." ["\\blacksquare" ] ) ;  $\square$   $\sim. (var)
;; Symbols
("inf" ["\\infty" ] ) ;
("ex" ["\\exists" ] ) ;
("exn" ["\\nexists" ] ) ;  $\exists$   $\sim n$  (neg)
("fa" ["\\forall" ] ) ;
("hb" ["\\hbar" ] ) ;
("hb." ["\\hslash" ] ) ;  $\hbar$   $\sim. (var)
("dd" ["\\mathrm{d}" ] ) ;
("dd." ["\\partial" ] ) ;  $\mathrm{d}$   $\sim. (var)
("ii" ["\\imath" ] ) ;
("jj" ["\\jmath" ] ) ;
("nab" ["\\nabla" ] ) ;
("cm" ["\\checkmark" ] ) ;
;; Symbols spaces
("qu" ["\\quad" ] ) ;
("quu" ["\\qquad" ] ) ;
;; Symbols arrow1
("<->" ["\\leftarrow" ] ) ;
("<->" ["\\rightarrow" ] ) ;
("<-^" ["\\uparrow" ] ) ;  $\sim\sim$  looks like up arrow head
("<-v" ["\\downarrow" ] ) ;  $\sim v$  looks like down arrow head
("<->" ["\\leftrightarrow" ] ) ;
("<-n" ["\\nleftarrow" ] ) ; negate ( $\sim n$ ) of prev. section
("<->n" ["\\nrightarrow" ] ) ; arrows +  $\sim n$ 
("<-^n" ["\\nuparrow" ] ) ;
("<-vn" ["\\ndownarrow" ] ) ;
("<->" ["\\nleftrightarrow" ] ) ;
("<->" ["\\longrightarrow" ] ) ; longer arrows, with 2 dashes
("<->" ["\\longleftarrow" ] ) ;$$$$$$ 
```

```

(|->"      ["\\mapsto"          ]) ; vertical-bar + ~->~
(""        [""                  ]) ; (this rendered wrong on Github)
(""        [""                  ]) ; (check from PDF file)
;; Symbols arrow2
("<="      ["\\Leftarrow"       ]) ; compared to single arrow
(">="      ["\\Rightarrow"      ]) ; these uses ~~= as the arrow shaft
("∧"   ["\\Uparrow"          ]) ;
("&v"       ["\\Downarrow"       ]) ;
("<=>"    ["\\Leftrightarrow"   ]) ;
("iff"     ["\\Leftrightarrow"   ]) ;
("<=n"     ["\\nLeftarrow"      ]) ; negate (~n~) of prev. section
(">=n"     ["\\nRightarrow"     ]) ; arrows + _n_
("<=>n"    ["\\nLeftrightarrow"  ]) ;
("iffn"    ["\\nLeftrightarrow"  ]) ;
("<==>"    ["\\Longleftarrow"    ]) ; longer arrows, with 2 dashes
("<=="     ["\\Longleftarrow"    ]) ;
("==>"     ["\\Longrightarrow"   ]) ;
;; Symbols arrow3
("<---"    ["\\xleftarrow[ ]{ }"  ]) ; these uses triple - or =
("---->"   ["\\xrightarrow[ ]{ }"  ]) ;
("===>"    ["\\xrightarrow[ ]{ }"  ]) ; ~mathtools~ lib required
("<===>"   ["\\xLeftarrow[ ]{ }"   ]) ; ~mathtools~ lib required
;; Symbols arrow3
("vec"     ["\\vec{"                ]) ;
("bar"     ["\\bar{"              ]) ;
("hat"     ["\\hat{"             ]) ;
("dot"     ["\\dot{"           ]) ;
("dot."    ["\\ddot{"          ]) ;
("dot.."   ["\\dddotted{"      ]) ;
("dot..." ["\\ddddotted{"         ]) ;
("dag"     ["^\\dagger"         ]) ;
("dag."    ["^\\ddagger"        ]) ;
("*.."     ["^*"                  ]) ;
("deg"     ["^\\circ"            ]) ;
("tr"      ["^T"               ]) ;
("tr."     ["^{-T}"             ]) ;
;; Operation: arith
("+_"      ["\\pm"                ]) ;
("-+"      ["\\mp"                ]) ;
("*x"      ["\\times"            ]) ;

```



```

("::"      ["\\div"          ]) ;
("**"      ["\\cdot"         ]) ;
;; Operation: arith
("="       ["\\equiv"        ]) ; Variation on
("<."      ["\\leq"           ]) ; < = >
(">."      ["\\geq"           ]) ; symbols
("<<"      ["\\ll"            ]) ;
(">>"      ["\\gg"            ]) ;
("=n"      ["\\neq"           ]) ; negation
("n"       ["\\nsim"          ]) ;
("<n"      ["\\nless"           ]) ;
(">n"      ["\\ngtr"          ]) ;
("<.n"     ["\\nleq"           ]) ;
(">.n"     ["\\ngeq"           ]) ;
("=?"      ["\\stackrel{?}{=}" ]) ; with question mark
("<?"      ["\\stackrel{?}{<}" ]) ;
(">?"      ["\\stackrel{?}{>}" ]) ;
("<.??"    ["\\stackrel{?}{\\leq}" ]) ;
(">.??"    ["\\stackrel{?}{\\geq}" ]) ;
("<<?"     ["\\stackrel{?}{\\ll}" ]) ;
(">>?"     ["\\stackrel{?}{\\gg}" ]) ;
("=y"      ["\\stackrel{\\checkmark}{=}" ]) ; with check mark
("<y"      ["\\stackrel{\\checkmark}{<}" ]) ;
(">y"      ["\\stackrel{\\checkmark}{>}" ]) ;
("<.y"     ["\\stackrel{\\checkmark}{\\leq}" ]) ;
(">.y"     ["\\stackrel{\\checkmark}{\\geq}" ]) ;
("<<y"     ["\\stackrel{\\checkmark}{\\ll}" ]) ;
(">>y"     ["\\stackrel{\\checkmark}{\\gg}" ]) ;
("."        ["\\sim"                ]) ; Another variations
("")       ["\\approx"             ]) ; on =
("3="      ["\\equiv"          ]) ;
("=: "     ["\\coloneqq"         ]) ;
(":= "     ["\\coloneqq"         ]) ;
;; Operation: arith
("in"      ["\\in"              ]) ;
("in."     ["\\ni"               ]) ;
("ni"      ["\\ni"               ]) ;
("inn"     ["\\notin"            ]) ; (neg)
("O/"      ["\\emptyset"         ]) ;
("nsr"     ["\\mathbb{R}"         ]) ; (n)umber (s)et (r)eal

```

```

("nsc"      ["\\mathbb{C}"      ]) ; (n)umber (s)et (c)omplex
("nsn"      ["\\mathbb{N}"      ]) ; ...
("nsp"      ["\\mathbb{P}"      ]) ; ...
("nsz"      ["\\mathbb{Z}"      ]) ; ...
("nsi"      ["\\mathbb{I}"      ]) ; ...
("sub"      ["\\subset"        ]) ;
("subn"     ["\\nssubseteq"     ]) ; (neg)
("sub."     ["\\subseteq"      ]) ; (var)
("sub.n"    ["\\nssubseteq"     ]) ; (var, neg)
("subn."    ["\\nssubseteq"     ]) ; (neg, var)
("sup"      ["\\supset"        ]) ;
("supn"     ["\\nsupseteq"     ]) ; (neg)
("sup."     ["\\supeseteq"     ]) ; (var)
("sup.n"    ["\\nsupseteq"     ]) ; (var, neg)
("supn."    ["\\nsupseteq"     ]) ; (neg, var)
;; Operation: arith
("or"       ["\\lor"          ]) ;
("and"      ["\\land"         ]) ;
("not"      ["\\neg"          ]) ;
("or."      ["\\text{ or }"    ]) ; (var)
("and."     ["\\text{ and }"   ]) ; (var)
("not."     ["\\text{ not }"   ]) ; (var)
;; Func: main
("rank"     ["\\mathrm{rank}"   ]) ;
("arg"      ["\\arg"           ]) ;
("det"      ["\\det"          ]) ;
("dim"      ["\\dim"          ]) ;
("exp"      ["\\exp("         ]) ;
("Im"       ["\\mathrm{Im}("   ]) ;
("Re"       ["\\mathrm{Re}("   ]) ;
("ln"       ["\\ln("          ]) ;
("log"      ["\\log("         ]) ;
("max"      ["\\max("         ]) ;
("min"      ["\\min("         ]) ;
("dim"      ["\\dim("         ]) ;
("sqrt"     ["\\sqrt("        ]) ;
("mod"      ["\\pmod("        ]) ;
("mod."     ["\\mod"          ]) ;
("mod.."    ["\\bmod"         ]) ;
;; Func: Trig

```

```

("cos"      ["\\cos("          ]) ("cosh"     ["\\cosh("        ])
("sin"      ["\\sin("          ]) ("sinh"     ["\\sinh("        ])
("tan"      ["\\tan("          ]) ("tanh"     ["\\tanh("        ])
("cot"      ["\\cot("          ]) ("coth"     ["\\coth("        ])
("acos"     ["\\arccos("       ]) ("cos."     ["\\arccos("       ])
("asin"     ["\\arcsin("       ]) ("sin."     ["\\arcsin("       ])
("atan"     ["\\arctan("       ]) ("tan."     ["\\arctan("       ])
;; Func: iter
("il"       ["\\limits_{ }"     ]) ;
("il"       ["\\limits_{ }^{ }"  ]) ;
("lim"      ["\\lim"            ]) ;
("sum"      ["\\sum"            ]) ;
("prod"     ["\\prod"           ]) ;
("int"      ["\\int"            ]) ;
("inti"     ["\\iint"           ]) ;
("intii"    ["\\iiint"          ]) ;
("intiii"   ["\\iiint"          ]) ;
("into"     ["\\oint"           ]) ;
("sum."     ["\\sum\\limits_{ i=1 }^{ n }"]) ;
("prod."    ["\\prod\\limits_{ i=1 }^{ n }"]) ;
("int."     ["\\int\\limits_{ }^{ }"]) ;
("int.."    ["\\int\\limits_{ 0 }^{ +\\infty }"]) ;
("int..." ["\\int\\limits_{ -\\infty }^{ +\\infty }"]) ;
("inti."    ["\\iint\\limits_{ }"] ) ;
("intii."   ["\\iiint\\limits_{ }"] ) ;
("intiii."  ["\\iiint\\limits_{ }"] ) ;
("into."    ["\\oint\\limits_{ }"] ) ;
;; Structural: Parenthesis
("(".."     ["\\left( \\right)"      ]) ;
("(".."     ["\\left( \\middle\\vert \\right)"]) ;
("[.."      ["\\left[ \\right]"        ]) ;
("[.."      ["\\left[ \\middle\\vert \\right]"]) ; (var)
("[..c"     ["\\lceil \\rceil"          ]) ; (var) (ceil)
("[..f"     ["\\lfloor \\rfloor"       ]) ; (var) (floor)
("{.."      ["\\left\\{ \\right\\}"      ]) ;
("{.."      ["\\left\\{ \\middle\\vert \\right\\}"]) ; (var)
("<.."      ["\\left< \\right>"          ]) ;
("<.."      ["\\left< \\middle\\vert \\right>"]) ; (var)
("||.."     ["\\left\\vert \\right\\vert"]) ;
("||.."     ["\\left\\Vert \\right\\Vert"]) ; (var)

```

```

("(. "      ["\\left("          ]) ; half (
("). "      ["\\right)"         ]) ; half )
("[. "      ["\\left["          ]) ; half [
("]. "      ["\\right]"         ]) ; half ]
("{. "      ["\\left\\{"        ]) ; half {
("}. "      ["\\right\\}"       ]) ; half }
("<. "      ["\\left<"          ]) ; half <
(">. "      ["\\right>"         ]) ; half >
("(. ."     ["\\left. "         ]) ; half left .
("). ."     ["\\right. "        ]) ; half right .
("|. "      ["\\Bigg\\vert_{ }^{ }"]) ; definite integral range
;; Structural: Text
("te"      ["\\text{"          ]) ; (te)xt
("tr"      ["\\mathrm{"         ]) ; (t)ext (r)oman
("tb"      ["\\mathbf{"         ]) ; (t)ext (b)old
("ti"      ["\\mathit{"         ]) ; (t)ext (i)talics
;; Structural: Text
("te"      ["\\text{"          ]) ; (te)xt
("tr"      ["\\mathrm{"         ]) ; (t)ext (r)oman
("tb"      ["\\mathbf{"         ]) ; (t)ext (b)old
("ti"      ["\\mathit{"         ]) ; (t)ext (i)talics
;; Structural: Sub-sup-scripts
("^"       ["^{ "              ]) ("_"       ["_{ "              ])
("pp"      ["^{ "              ]) ("l1"      ["_{ "              ])
("p0"      ["^0"             ]) ("l0"      ["_0"             ])
("p1"      ["^1"             ]) ("l1"      ["_1"             ])
("p2"      ["^2"             ]) ("l2"      ["_2"             ])
("p3"      ["^3"             ]) ("l3"      ["_3"             ])
("p4"      ["^4"             ]) ("l4"      ["_4"             ])
("pn"      ["^n"             ]) ("l_n"     ["_n"             ])
("px"      ["^x"             ]) ("li"      ["_i"             ])
("__"      ["\\underset{ }{ }"]) ("^^"      ["\\overset{ }{ }"])
("___."    ["\\underbrace{ }_{ }"]) ("^^."    ["\\overbrace{ }^{ }"])
("___.."   ["\\underline{ }" ]) ("^^.."   ["\\overline{ }" ])
;; Structural: misc
("binom"   ["\\binom{}{ "       ]) ; Binom
("box"     ["\\boxed{}{ "       ]) ; Putting box around object
("fr"      ["\\frac{}{ "       ]) ; Fractions
("can"     ["\\cancel "        ]) ;
("&="     ["&=\\n\\\\\\\\\\\\\\\\"]]) ;

```

```

( "&"      ["&=\\n\\\\\\\\\\\\\\"]    )  ;
;; Structural: xy
("xy"      ["\\xymatrix{\\n\\n}"    ] )  ;
("bu"      ["\\bullet"              ] )  ;
("ar"      ["\\ar"                  ] )  ;
)

```