

Emacs TeQ: (T_EX + Quail)

Input Method written in Quail for entering L^AT_EX math expressions

Garid Zorigoo

April 13, 2023

Contents

1	Alphabet related stuff :	1
1.1	Greek	1
1.2	Matrix (aka bold)	3
1.3	Vector	4
1.4	Hat	5
1.5	Dots	6
1.6	DDots	7
2	Function Expansion	8
3	Symbols :	8
3.1	Dots related	8
3.2	Geometry	9
3.3	Letter like	9
3.4	Spaces	10
3.5	Arrows:	10
3.5.1	Single:	10
3.5.2	Double:	11
3.5.3	Long arrow with top-bottom entries	11
4	Symbol Modification	12
4.1	Accents (variable decoration?)	12
4.2	Superscripts & Subsripts (power & lower)	13

5	Binary Operation Symbols	13
5.1	Simple Arithmetics:	13
5.2	Binary Relations:	15
5.3	Set symbols	16
5.4	Logic	16
6	Functions	17
6.1	Function	17
6.2	Trigonometry: function	17
6.3	Iterative-like operation:	18
7	Structural:	19
7.1	Parenthesis Related	19
7.2	Texts:	20
7.3	Misc.	20
7.4	xy Diagram related	20
8	Formatting Table into Elisp	20
9	Executable elisp function definition	33
10	Making the el	34

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.	α	<code>\alpha</code>	A.	A	<code>A</code>
b.	β	<code>\beta</code>	B.	B	<code>B</code>
c.	ψ	<code>\psi</code>	C.	Ψ	<code>\Psi</code>
d.	δ	<code>\delta</code>	D.	Δ	<code>\Delta</code>
e.	ϵ	<code>\epsilon</code>	E.	E	<code>E</code>
f.	ϕ	<code>\phi</code>	F.	Φ	<code>\Phi</code>
g.	γ	<code>\gamma</code>	G.	Γ	<code>\Gamma</code>
h.	η	<code>\eta</code>	H.	H	<code>H</code>
i.	ι	<code>\iota</code>	I.	I	<code>I</code>
j.	ξ	<code>\xi</code>	J.	Ξ	<code>\Xi</code>
k.	κ	<code>\kappa</code>	K.	K	<code>K</code>
l.	λ	<code>\lambda</code>	L.	Λ	<code>\Lambda</code>
m.	μ	<code>\mu</code>	M.	M	<code>M</code>
n.	ν	<code>\nu</code>	N.	N	<code>N</code>
o.	o	<code>o</code>	O.	O	<code>O</code>
p.	π	<code>\pi</code>	P.	Π	<code>\Pi</code>
r.	ρ	<code>\rho</code>	R.	P	<code>P</code>
s.	σ	<code>\sigma</code>	S.	Σ	<code>\Sigma</code>
t.	τ	<code>\tau</code>	T.	T	<code>T</code>
th.	θ	<code>\theta</code>	Th.	Θ	<code>\Theta</code>
u.	v	<code>\upsilon</code>	U.	Υ	<code>\Upsilon</code>
w.	ω	<code>\omega</code>	W.	Ω	<code>\Omega</code>
x.	χ	<code>\chi</code>	X.	X	<code>X</code>
z.	ζ	<code>\zeta</code>	Z.	Z	<code>Z</code>
e..	ε	<code>\varepsilon</code>	r..	ϱ	<code>\varrho</code>
f..	φ	<code>\varphi</code>	p..	ϖ	<code>\varpi</code>
s..	ς	<code>\varsigma</code>	t..	ϑ	<code>\vartheta</code>

1.2 Matrix (aka bold)

Table 2: Matrix

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

1.3 Vector

Table 3: Vectors

key	sym	latex (upper)	key	sym	latex (lower)
Av	\vec{A}	<code>\vec{A}</code>	av	\vec{a}	<code>\vec{a}</code>
Bv	\vec{B}	<code>\vec{B}</code>	bv	\vec{b}	<code>\vec{b}</code>
Cv	\vec{C}	<code>\vec{C}</code>	cv	\vec{c}	<code>\vec{c}</code>
Dv	\vec{D}	<code>\vec{D}</code>	dv	\vec{d}	<code>\vec{d}</code>
Ev	\vec{E}	<code>\vec{E}</code>	ev	\vec{e}	<code>\vec{e}</code>
Fv	\vec{F}	<code>\vec{F}</code>	fv	\vec{f}	<code>\vec{f}</code>
Gv	\vec{G}	<code>\vec{G}</code>	gv	\vec{g}	<code>\vec{g}</code>
Hv	\vec{H}	<code>\vec{H}</code>	hv	\vec{h}	<code>\vec{h}</code>
Iv	\vec{I}	<code>\vec{I}</code>	iv	\vec{i}	<code>\vec{i}</code>
Jv	\vec{J}	<code>\vec{J}</code>	jv	\vec{j}	<code>\vec{j}</code>
Kv	\vec{K}	<code>\vec{K}</code>	kv	\vec{k}	<code>\vec{k}</code>
Lv	\vec{L}	<code>\vec{L}</code>	lv	\vec{l}	<code>\vec{l}</code>
Mv	\vec{M}	<code>\vec{M}</code>	mv	\vec{m}	<code>\vec{m}</code>
Nv	\vec{N}	<code>\vec{N}</code>	nv	\vec{n}	<code>\vec{n}</code>
Ov	\vec{O}	<code>\vec{O}</code>	ov	\vec{o}	<code>\vec{o}</code>
Pv	\vec{P}	<code>\vec{P}</code>	pv	\vec{p}	<code>\vec{p}</code>
Qv	\vec{Q}	<code>\vec{Q}</code>	qv	\vec{q}	<code>\vec{q}</code>
Rv	\vec{R}	<code>\vec{R}</code>	rv	\vec{r}	<code>\vec{r}</code>
Sv	\vec{S}	<code>\vec{S}</code>	sv	\vec{s}	<code>\vec{s}</code>
Tv	\vec{T}	<code>\vec{T}</code>	tv	\vec{t}	<code>\vec{t}</code>
Uv	\vec{U}	<code>\vec{U}</code>	uv	\vec{u}	<code>\vec{u}</code>
Vv	\vec{V}	<code>\vec{V}</code>	vv	\vec{v}	<code>\vec{v}</code>
Wv	\vec{W}	<code>\vec{W}</code>	wv	\vec{w}	<code>\vec{w}</code>
Xv	\vec{X}	<code>\vec{X}</code>	xv	\vec{x}	<code>\vec{x}</code>
Yv	\vec{Y}	<code>\vec{Y}</code>	yv	\vec{y}	<code>\vec{y}</code>
Zv	\vec{Z}	<code>\vec{Z}</code>	zv	\vec{z}	<code>\vec{z}</code>

1.4 Hat

Table 4: Vectors and Hats

key	sym	latex (hat)	key	sym	latex (hat)
Ah	\hat{A}	<code>\hat{A}</code>	ah	\hat{a}	<code>\hat{a}</code>
Bh	\hat{B}	<code>\hat{B}</code>	bh	\hat{b}	<code>\hat{b}</code>
Ch	\hat{C}	<code>\hat{C}</code>	ch	\hat{c}	<code>\hat{c}</code>
Dh	\hat{D}	<code>\hat{D}</code>	dh	\hat{d}	<code>\hat{d}</code>
Eh	\hat{E}	<code>\hat{E}</code>	eh	\hat{e}	<code>\hat{e}</code>
Fh	\hat{F}	<code>\hat{F}</code>	fh	\hat{f}	<code>\hat{f}</code>
Gh	\hat{G}	<code>\hat{G}</code>	gh	\hat{g}	<code>\hat{g}</code>
Hh	\hat{H}	<code>\hat{H}</code>	hh	\hat{h}	<code>\hat{h}</code>
Ih	\hat{I}	<code>\hat{I}</code>	ih	\hat{i}	<code>\hat{i}</code>
Jh	\hat{J}	<code>\hat{J}</code>	jh	\hat{j}	<code>\hat{j}</code>
Kh	\hat{K}	<code>\hat{K}</code>	kh	\hat{k}	<code>\hat{k}</code>
Lh	\hat{L}	<code>\hat{L}</code>	lh	\hat{l}	<code>\hat{l}</code>
Mh	\hat{M}	<code>\hat{M}</code>	mh	\hat{m}	<code>\hat{m}</code>
Nh	\hat{N}	<code>\hat{N}</code>	nh	\hat{n}	<code>\hat{n}</code>
Oh	\hat{O}	<code>\hat{O}</code>	oh	\hat{o}	<code>\hat{o}</code>
Ph	\hat{P}	<code>\hat{P}</code>	ph	\hat{p}	<code>\hat{p}</code>
Qh	\hat{Q}	<code>\hat{Q}</code>	qh	\hat{q}	<code>\hat{q}</code>
Rh	\hat{R}	<code>\hat{R}</code>	rh	\hat{r}	<code>\hat{r}</code>
Sh	\hat{S}	<code>\hat{S}</code>	sh	\hat{s}	<code>\hat{s}</code>
Th	\hat{T}	<code>\hat{T}</code>	th	\hat{t}	<code>\hat{t}</code>
Uh	\hat{U}	<code>\hat{U}</code>	uh	\hat{u}	<code>\hat{u}</code>
Vh	\hat{V}	<code>\hat{V}</code>	vh	\hat{v}	<code>\hat{v}</code>
Wh	\hat{W}	<code>\hat{W}</code>	wh	\hat{w}	<code>\hat{w}</code>
Xh	\hat{X}	<code>\hat{X}</code>	xh	\hat{x}	<code>\hat{x}</code>
Yh	\hat{Y}	<code>\hat{Y}</code>	yh	\hat{y}	<code>\hat{y}</code>
Zh	\hat{Z}	<code>\hat{Z}</code>	zh	\hat{z}	<code>\hat{z}</code>

1.5 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>

1.6 DDots

Table 6: DDots

key	sym	latex (vec)	key	sym	latex (hat)
add	\ddot{a}	<code>\ddot{a}</code>	Add	\ddot{A}	<code>\ddot{A}</code>
bdd	\ddot{b}	<code>\ddot{b}</code>	Bdd	\ddot{B}	<code>\ddot{B}</code>
cdd	\ddot{c}	<code>\ddot{c}</code>	Cdd	\ddot{C}	<code>\ddot{C}</code>
ddd	\ddot{d}	<code>\ddot{d}</code>	Ddd	\ddot{D}	<code>\ddot{D}</code>
edd	\ddot{e}	<code>\ddot{e}</code>	Edd	\ddot{E}	<code>\ddot{E}</code>
fdd	\ddot{f}	<code>\ddot{f}</code>	Fdd	\ddot{F}	<code>\ddot{F}</code>
gdd	\ddot{g}	<code>\ddot{g}</code>	Gdd	\ddot{G}	<code>\ddot{G}</code>
hdd	\ddot{h}	<code>\ddot{h}</code>	Hdd	\ddot{H}	<code>\ddot{H}</code>
idd	\ddot{i}	<code>\ddot{i}</code>	Idd	\ddot{I}	<code>\ddot{I}</code>
jdd	\ddot{j}	<code>\ddot{j}</code>	Jdd	\ddot{J}	<code>\ddot{J}</code>
kdd	\ddot{k}	<code>\ddot{k}</code>	Kdd	\ddot{K}	<code>\ddot{K}</code>
ldd	\ddot{l}	<code>\ddot{l}</code>	Ldd	\ddot{L}	<code>\ddot{L}</code>
mdd	\ddot{m}	<code>\ddot{m}</code>	Mdd	\ddot{M}	<code>\ddot{M}</code>
ndd	\ddot{n}	<code>\ddot{n}</code>	Ndd	\ddot{N}	<code>\ddot{N}</code>
odd	\ddot{o}	<code>\ddot{o}</code>	Odd	\ddot{O}	<code>\ddot{O}</code>
pdd	\ddot{p}	<code>\ddot{p}</code>	Pdd	\ddot{P}	<code>\ddot{P}</code>
qdd	\ddot{q}	<code>\ddot{q}</code>	Qdd	\ddot{Q}	<code>\ddot{Q}</code>
rdd	\ddot{r}	<code>\ddot{r}</code>	Rdd	\ddot{R}	<code>\ddot{R}</code>
sdd	\ddot{s}	<code>\ddot{s}</code>	Sdd	\ddot{S}	<code>\ddot{S}</code>
tdd	\ddot{t}	<code>\ddot{t}</code>	Tdd	\ddot{T}	<code>\ddot{T}</code>
udd	\ddot{u}	<code>\ddot{u}</code>	Udd	\ddot{U}	<code>\ddot{U}</code>
vdd	\ddot{v}	<code>\ddot{v}</code>	Vdd	\ddot{V}	<code>\ddot{V}</code>
wdd	\ddot{w}	<code>\ddot{w}</code>	Wdd	\ddot{W}	<code>\ddot{W}</code>
xdd	\ddot{x}	<code>\ddot{x}</code>	Xdd	\ddot{X}	<code>\ddot{X}</code>
ydd	\ddot{y}	<code>\ddot{y}</code>	Ydd	\ddot{Y}	<code>\ddot{Y}</code>
zdd	\ddot{z}	<code>\ddot{z}</code>	Zdd	\ddot{Z}	<code>\ddot{Z}</code>

2 Function Expansion

Table 7: Keys that will execute some elisp functions

key	sym	latex	description
/	$\frac{\Box}{\Box}$	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
gg		quail-TeX-next	go to next space
GG		quail-TeX-prev	go to prev space

3 Symbols :

3.1 Dots related

Table 8: Multiple Dots Related

key	sym	latex	description
...	...	\dots	3 dots
.v	\vdots	\vdots	vertical dots
.d	\ddots	\ddots	diagonale dots
.l	...	\ldots	low dots

3.2 Geometry

Table 9:

key	sym	latex	description
perp	\perp	<code>\perp</code>	
perpn	\nparallel	<code>\not\perp</code>	\perp n (neg)
para	\parallel	<code>\parallel</code>	
paran	\nparallel	<code>\nparallel</code>	\parallel n (neg)
ang	\angle	<code>\angle</code>	
ang.	\sphericalangle	<code>\measuredangle</code>	\angle . (var)
tri	\triangle	<code>\vartriangle</code>	
trin	∇	<code>\triangledown</code>	\triangle n (neg)
squ	\square	<code>\square</code>	
tri.	\blacktriangle	<code>\blacktriangle</code>	\triangle . (var)
trin.	\blacktriangledown	<code>\blacktriangledown</code>	\triangle n. (neg,var)
squ.	\blacksquare	<code>\blacksquare</code>	\square . (var)

3.3 Letter like

Table 10: Letter-like Symbol

key	sym	latex	description
inf	∞	<code>\infty</code>	
ex	\exists	<code>\exists</code>	
exn	\nexists	<code>\nexists</code>	\exists + <u>n</u> (neg)
fa	\forall	<code>\forall</code>	
hb	\hbar	<code>\hbar</code>	
hb.	\hslash	<code>\hslash</code>	\hbar + . (var)
dfd	d	<code>\mathrm{d}</code>	
dfd.	∂	<code>\partial</code>	d + . (var)
dff	$\frac{\mathrm{d}}{\mathrm{d}\square}$	<code>\frac{\mathrm{d}}{\mathrm{d}\square}</code>	
dff.	$\frac{\partial}{\partial\square}$	<code>\frac{\partial}{\partial\square}</code>	$\frac{\mathrm{d}}{\mathrm{d}\square}$ + .(var)
ii	\imath	<code>\imath</code>	
jj	\jmath	<code>\jmath</code>	
nab	∇	<code>\nabla</code>	
cm	\checkmark	<code>\checkmark</code>	

3.4 Spaces

Table 11: Space Symbol

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

3.5 Arrows:

3.5.1 Single:

Table 12: Single Line arrows

key	sym	latex	description
<code><-</code>	\leftarrow	<code>\leftarrow</code>	
<code>-></code>	\rightarrow	<code>\rightarrow</code>	
<code>-^</code>	\uparrow	<code>\uparrow</code>	\wedge looks like up arrow head
<code>-v</code>	\downarrow	<code>\downarrow</code>	\vee looks like down arrow head
<code><-></code>	\leftrightarrow	<code>\leftrightarrow</code>	
<code><-n</code>	\nleftarrow	<code>\nleftarrow</code>	negate (n) of prev. section
<code>->n</code>	\rrightarrow	<code>\rrightarrow</code>	arrows + <u>n</u>
<code>-^n</code>	\uparrow	<code>\nuparrow</code>	
<code>-vn</code>	\downarrow	<code>\ndownarrow</code>	
<code><-></code>	\leftrightarrow	<code>\nleftrightarrow</code>	
<code>--></code>	\longrightarrow	<code>\longrightarrow</code>	longer arrows, with 2 dashes
<code><--</code>	\longleftarrow	<code>\longleftarrow</code>	
<code> -></code>	\mapsto	<code>\mapsto</code>	vertical-bar + <code>-></code> (this might rendered wrongly on Github)

3.5.2 Double:

Table 13: Double Line arrows

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

3.5.3 Long arrow with top-bottom entries

Table 14: Long arrow Line arrows

key	sym	latex	description
$\xleftarrow{\quad}$	$\xleftarrow{\quad}$	<code>\xleftarrow[]{ }</code>	these uses triple - or =
$\xrightarrow{\quad}$	$\xrightarrow{\quad}$	<code>\xrightarrow[]{ }</code>	
$\xRightarrow{\quad}$	$\xRightarrow{\quad}$	<code>\xRightarrow[]{ }</code>	mathtools lib required
$\xLeftarrow{\quad}$	$\xLeftarrow{\quad}$	<code>\xLeftarrow[]{ }</code>	mathtools lib required

4 Symbol Modification

4.1 Accents (variable decoration?)

Table 15:

key	sym	latex	description
vec	$\vec{\square}$	<code>\vec{}</code>	
bar	$\bar{\square}$	<code>\bar{}</code>	
hat	$\hat{\square}$	<code>\hat{}</code>	
dot	$\dot{\square}$	<code>\dot{}</code>	
dot.	$\ddot{\square}$	<code>\ddot{}</code>	(var)
dot..	$\dddot{\square}$	<code>\dddot{}</code>	(var)
dot...	$\ddddot{\square}$	<code>\ddddot{}</code>	(var)
dag	\square^\dagger	<code>^\dagger</code>	
dag.	\square^{\ddagger}	<code>^\ddagger</code>	(var)
..	\square^	<code>^*</code>	
deg	\square°	<code>^\circ</code>	
tr	\square^T	<code>^T</code>	
trn	\square^{-T}	<code>^{-T}</code>	(neg)

4.2 Superscripts & Subscripts (power & lower)

Table 16:

key	sym	latex	key	sym	latex
^	\square^\square	$\sim\{$	-	\square_\square	$_ \{$
pp	\square^\square	$\sim\{$	11	\square_\square	$_ \{$
p0	\square^0	~ 0	10	\square_0	$_ 0$
p1	\square^1	~ 1	11	\square_1	$_ 1$
p2	\square^2	~ 2	12	\square_2	$_ 2$
p3	\square^3	~ 3	13	\square_3	$_ 3$
p4	\square^4	~ 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	$\backslash overset\{ \}\{ \}$
---	$\underbrace{\square}$	$\backslash underbrace\{ \}_\{ \}$	^^.	$\overbrace{\square}$	$\backslash overbrace\{ \}^\{ \}$
---	$\underline{\square}$	$\backslash underline\{ \}$	^^..	$\overline{\square}$	$\backslash overline\{ \}$

5 Binary Operation Symbols

5.1 Simple Arithmetics:

Table 17: Simple Arithmetics operations

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

5.2 Binary Relations:

Table 18:

key	sym	latex	description
<.	\leq	<code>\leq</code>	< = > symbols
>.	\geq	<code>\geq</code>	
«	\ll	<code>\ll</code>	
»	\gg	<code>\gg</code>	
=n	\neq	<code>\neq</code>	negation
<n	\nless	<code>\nless</code>	
>n	\ngtr	<code>\ngtr</code>	
<.n	\nleq	<code>\nleq</code>	
>.n	\ngeq	<code>\ngeq</code>	
=?	$\stackrel{?}{=}$	<code>\stackrel{?}{=}</code>	with question mark
<?	$\stackrel{?}{<}$	<code>\stackrel{?}{<}</code>	
>?	$\stackrel{?}{>}$	<code>\stackrel{?}{>}</code>	
<.?	$\stackrel{?}{\leq}$	<code>\stackrel{?}{\leq}</code>	
>.?	$\stackrel{?}{\geq}$	<code>\stackrel{?}{\geq}</code>	
«?	$\stackrel{?}{\ll}$	<code>\stackrel{?}{\ll}</code>	
»?	$\stackrel{?}{\gg}$	<code>\stackrel{?}{\gg}</code>	
=y	$\stackrel{\checkmark}{=}$	<code>\stackrel{\checkmark}{=}</code>	with check mark
<y	$\stackrel{\checkmark}{<}$	<code>\stackrel{\checkmark}{<}</code>	
>y	$\stackrel{\checkmark}{>}$	<code>\stackrel{\checkmark}{>}</code>	
<.y	$\stackrel{\checkmark}{\leq}$	<code>\stackrel{\checkmark}{\leq}</code>	
>.y	$\stackrel{\checkmark}{\geq}$	<code>\stackrel{\checkmark}{\geq}</code>	
«y	$\stackrel{\checkmark}{\ll}$	<code>\stackrel{\checkmark}{\ll}</code>	
»y	$\stackrel{\checkmark}{\gg}$	<code>\stackrel{\checkmark}{\gg}</code>	
=.	\equiv	<code>\equiv</code>	Variations of =
sim	\sim	<code>\sim</code>	
simn	\nsim	<code>\nsim</code>	
=..	\approx	<code>\approx</code>	
3=	\equiv	<code>\equiv</code>	
=:	\coloneqq	<code>\coloneqq</code>	
:=	\coloneqq	<code>\coloneqq</code>	

5.3 Set symbols

Table 19:

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

5.4 Logic

Table 20:

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

6 Functions

6.1 Function

Table 21:

key	sym	latex	description
rank	rank	rank	
arg	arg	\arg	
det	det	\det	
dim	dim	\dim	
exp	exp	$\exp($	
Im	Im	$\mathrm{Im}($	
Re	Re	$\mathrm{Re}($	
ln	ln	$\ln($	
log	log	$\log($	
max	max	$\max($	
min	min	$\min($	
dim	dim	$\dim($	
sqrt	$\sqrt{}$	$\sqrt{}$	
mod	$\square \pmod{\square}$	$\pmod{}$	
mod.	$\square \bmod \square$	\bmod	
mod..	$\square \bmod \square$	\bmod	

6.2 Trigonometry: function

Table 22:

key	sym	latex	key	sym	latex
cos	$\cos(\square)$	$\cos($	cosh	$\cosh(\square)$	$\cosh($
sin	$\sin(\square)$	$\sin($	sinh	$\sinh(\square)$	$\sinh($
tan	$\tan(\square)$	$\tan($	tanh	$\tanh(\square)$	$\tanh($
cot	$\cot(\square)$	$\cot($	coth	$\coth(\square)$	$\coth($
acos	$\arccos(\square)$	$\arccos($	cos.	$\arccos(\square)$	$\arccos($
asin	$\arcsin(\square)$	$\arcsin($	sin.	$\arcsin(\square)$	$\arcsin($
atan	$\arctan(\square)$	$\arctan($	tan.	$\arctan(\square)$	$\arctan($

6.3 Iterative-like operation:

Table 23: Integrals, Sums, Products

key	sym	latex	description
il	\sum	<code>\limits_{ }</code>	(limits apparently doesn't render on G
il.	\sum	<code>\limits_{ }^{ }</code>	. (var)
lim	lim	<code>\lim</code>	
sum	\sum	<code>\sum</code>	
prod	\prod	<code>\prod</code>	
int	\int	<code>\int</code>	
inti	\iint	<code>\iint</code>	$\int + i$
intii	\iiint	<code>\iiint</code>	$\int + ii$
intiii	\iiint	<code>\iiint</code>	$\int + iii$
into	\oint	<code>\oint</code>	$\int + o$
sum.	$\sum_{i=1}^n$	<code>\sum\limits_{ i=1 }^{ n }</code>	. (var)
prod.	$\prod_{i=1}^n$	<code>\prod\limits_{ i=1 }^{ n }</code>	. (var)
int.	\int_{\square}	<code>\int\limits_{ }^{ }</code>	. (var)
int..	$\int_0^{+\infty}$	<code>\int\limits_{ 0 }^{ +\infty }</code>	. (var)
int...	$\int_{-\infty}^{+\infty}$	<code>\int\limits_{ -\infty }^{ +\infty }</code>	. (var)
inti.	\iint_{\square}	<code>\iint\limits_{ }</code>	. (var)
intii.	\iiint_{\square}	<code>\iiint\limits_{ }</code>	. (var)
intiii.	\iiint_{\square}	<code>\iiint\limits_{ }</code>	. (var)
into.	\oint_{\square}	<code>\oint\limits_{ }</code>	. (var)

7 Structural:

7.1 Parenthesis Related

Table 24:

key	sym	latex	description
() .	(\square)	<code>\left(\right)</code>	
() ..	($\square \square$)	<code>\left(\middle\vert \right)</code>	
[] .	[\square]	<code>\left[\right]</code>	
[] ..	[$\square \square$]	<code>\left[\middle\vert \right]</code>	(var)
[] .c	[\square]	<code>\lceil \rceil</code>	(var) (ceil)
[] .f	[\square]	<code>\lfloor \rfloor</code>	(var) (floor)
{ } .	{ \square }	<code>\left\{ \right\}</code>	
{ } ..	{ $\square \square$ }	<code>\left\{ \middle\vert \right\}</code>	(var)
< > .	< \square >	<code>\left< \right></code>	
< > ..	< $\square \square$ >	<code>\left< \middle\vert \right></code>	(var)
(. .)	(\square \square)	<code>\left(</code>	half (
.) .)	(\square)	<code>\right)</code>	half)
[. .]	[\square \square]	<code>\left[</code>	half [
]. .]	[\square]	<code>\right]</code>	half]
{. .}	{ \square \square }	<code>\left\{</code>	half {
}. .}	{ \square }	<code>\right\}</code>	half }
<. .>	< \square \square >	<code>\left<</code>	half <
>. .>	< \square >	<code>\right></code>	half >
(.. .)		<code>\left.</code>	half left .
).. .)		<code>\right.</code>	half right .
.		<code>\middle\vert</code>	Vertical bar related
..	$\square \left \square \right.$	<code>\Bigg\vert</code>	Vertical bar related
.	\square	<code>\left\vert \right\vert</code>	Vertical bar related
..	\square	<code>\left\ \right\ </code>	Vertical bar related

7.2 Texts:

Table 25:

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

7.3 Misc.

Table 26:

key	sym	latex	description
binom	$\binom{\square}{\square}$	<code>\binom{}</code>	Binom
box	$\boxed{\square}$	<code>\boxed{}</code>	Putting box around object
ff	$\frac{\square}{\square}$	<code>\frac{}</code>	Fractions
can	$\cancel{\square}$	<code>\cancel{}</code>	
==		<code>&=\n</code>	helps in align env.

7.4 xy Diagram related

Table 27:

key	sym	latex	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, sym, trans, 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, sym, trans, 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("Vector & Hat", tbl_1_hat)
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." ["\\upsilon" ]) ("U." ["\\Upsilon" ])
("w." ["\\omega" ]) ("W." ["\\Omega" ])
("x." ["\\chi" ]) ("X." ["X" ])
("z." ["\\zeta" ]) ("Z." ["Z" ])
("e.." ["\\varepsilon" ]) ("r.." ["\\varrho" ])
("f.." ["\\varphi" ]) ("p.." ["\\varpi" ])
("s.." ["\\varsigma" ]) ("t.." ["\\vartheta" ])

;; 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{O}" ]) ("Om" ["\\mathbf{O}" ])

;; Vector & Hat
("Av" ["\\vec{A}" ]) ("av" ["\\vec{a}" ])
("Bv" ["\\vec{B}" ]) ("bv" ["\\vec{b}" ])
("Cv" ["\\vec{C}" ]) ("cv" ["\\vec{c}" ])
("Dv" ["\\vec{D}" ]) ("dv" ["\\vec{d}" ])

```



```

("Ev"      ["\\vec{E}"      ]) ("ev"      ["\\vec{e}"      ])
("Fv"      ["\\vec{F}"      ]) ("fv"      ["\\vec{f}"      ])
("Gv"      ["\\vec{G}"      ]) ("gv"      ["\\vec{g}"      ])
("Hv"      ["\\vec{H}"      ]) ("hv"      ["\\vec{h}"      ])
("Iv"      ["\\vec{I}"      ]) ("iv"      ["\\vec{i}"      ])
("Jv"      ["\\vec{J}"      ]) ("jv"      ["\\vec{j}"      ])
("Kv"      ["\\vec{K}"      ]) ("kv"      ["\\vec{k}"      ])
("Lv"      ["\\vec{L}"      ]) ("lv"      ["\\vec{l}"      ])
("Mv"      ["\\vec{M}"      ]) ("mv"      ["\\vec{m}"      ])
("Nv"      ["\\vec{N}"      ]) ("nv"      ["\\vec{n}"      ])
("Ov"      ["\\vec{O}"      ]) ("ov"      ["\\vec{o}"      ])
("Pv"      ["\\vec{P}"      ]) ("pv"      ["\\vec{p}"      ])
("Qv"      ["\\vec{Q}"      ]) ("qv"      ["\\vec{q}"      ])
("Rv"      ["\\vec{R}"      ]) ("rv"      ["\\vec{r}"      ])
("Sv"      ["\\vec{S}"      ]) ("sv"      ["\\vec{s}"      ])
("Tv"      ["\\vec{T}"      ]) ("tv"      ["\\vec{t}"      ])
("Uv"      ["\\vec{U}"      ]) ("uv"      ["\\vec{u}"      ])
("Vv"      ["\\vec{V}"      ]) ("vv"      ["\\vec{v}"      ])
("Wv"      ["\\vec{W}"      ]) ("wv"      ["\\vec{w}"      ])
("Xv"      ["\\vec{X}"      ]) ("xv"      ["\\vec{x}"      ])
("Yv"      ["\\vec{Y}"      ]) ("yv"      ["\\vec{y}"      ])
("Zv"      ["\\vec{Z}"      ]) ("zv"      ["\\vec{z}"      ])

```

;; Vector & Hat

```

("Ah"      ["\\hat{A}"      ]) ("ah"      ["\\hat{a}"      ])
("Bh"      ["\\hat{B}"      ]) ("bh"      ["\\hat{b}"      ])
("Ch"      ["\\hat{C}"      ]) ("ch"      ["\\hat{c}"      ])
("Dh"      ["\\hat{D}"      ]) ("dh"      ["\\hat{d}"      ])
("Eh"      ["\\hat{E}"      ]) ("eh"      ["\\hat{e}"      ])
("Fh"      ["\\hat{F}"      ]) ("fh"      ["\\hat{f}"      ])
("Gh"      ["\\hat{G}"      ]) ("gh"      ["\\hat{g}"      ])
("Hh"      ["\\hat{H}"      ]) ("hh"      ["\\hat{h}"      ])
("Ih"      ["\\hat{I}"      ]) ("ih"      ["\\hat{i}"      ])
("Jh"      ["\\hat{J}"      ]) ("jh"      ["\\hat{j}"      ])
("Kh"      ["\\hat{K}"      ]) ("kh"      ["\\hat{k}"      ])
("Lh"      ["\\hat{L}"      ]) ("lh"      ["\\hat{l}"      ])
("Mh"      ["\\hat{M}"      ]) ("mh"      ["\\hat{m}"      ])
("Nh"      ["\\hat{N}"      ]) ("nh"      ["\\hat{n}"      ])
("Oh"      ["\\hat{O}"      ]) ("oh"      ["\\hat{o}"      ])
("Ph"      ["\\hat{P}"      ]) ("ph"      ["\\hat{p}"      ])
("Qh"      ["\\hat{Q}"      ]) ("qh"      ["\\hat{q}"      ])

```

("Rh"	["\\hat{R}")	("rh"	["\\hat{r}")
("Sh"	["\\hat{S}")	("sh"	["\\hat{s}")
("Th"	["\\hat{T}")	("th"	["\\hat{t}")
("Uh"	["\\hat{U}")	("uh"	["\\hat{u}")
("Vh"	["\\hat{V}")	("vh"	["\\hat{v}")
("Wh"	["\\hat{W}")	("wh"	["\\hat{w}")
("Xh"	["\\hat{X}")	("xh"	["\\hat{x}")
("Yh"	["\\hat{Y}")	("yh"	["\\hat{y}")
("Zh"	["\\hat{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					
("add"	["\\ddot{a}")	("Add"	["\\ddot{A}")
("bdd"	["\\ddot{b}")	("Bdd"	["\\ddot{B}")
("cdd"	["\\ddot{c}")	("Cdd"	["\\ddot{C}")

```

("ddd" ["\\ddot{d}" ]) ("Ddd" ["\\ddot{D}" ])
("edd" ["\\ddot{e}" ]) ("Edd" ["\\ddot{E}" ])
("fdd" ["\\ddot{f}" ]) ("Fdd" ["\\ddot{F}" ])
("gdd" ["\\ddot{g}" ]) ("Gdd" ["\\ddot{G}" ])
("hdd" ["\\ddot{h}" ]) ("Hdd" ["\\ddot{H}" ])
("idd" ["\\ddot{i}" ]) ("Idd" ["\\ddot{I}" ])
("jdd" ["\\ddot{j}" ]) ("Jdd" ["\\ddot{J}" ])
("kdd" ["\\ddot{k}" ]) ("Kdd" ["\\ddot{K}" ])
("ldd" ["\\ddot{l}" ]) ("Ldd" ["\\ddot{L}" ])
("mdd" ["\\ddot{m}" ]) ("Mdd" ["\\ddot{M}" ])
("ndd" ["\\ddot{n}" ]) ("Ndd" ["\\ddot{N}" ])
("odd" ["\\ddot{o}" ]) ("Odd" ["\\ddot{O}" ])
("pdd" ["\\ddot{p}" ]) ("Pdd" ["\\ddot{P}" ])
("qdd" ["\\ddot{q}" ]) ("Qdd" ["\\ddot{Q}" ])
("rdd" ["\\ddot{r}" ]) ("Rdd" ["\\ddot{R}" ])
("sdd" ["\\ddot{s}" ]) ("Sdd" ["\\ddot{S}" ])
("tdd" ["\\ddot{t}" ]) ("Tdd" ["\\ddot{T}" ])
("udd" ["\\ddot{u}" ]) ("Udd" ["\\ddot{U}" ])
("vdd" ["\\ddot{v}" ]) ("Vdd" ["\\ddot{V}" ])
("wdd" ["\\ddot{w}" ]) ("Wdd" ["\\ddot{W}" ])
("xdd" ["\\ddot{x}" ]) ("Xdd" ["\\ddot{X}" ])
("ydd" ["\\ddot{y}" ]) ("Ydd" ["\\ddot{Y}" ])
("zdd" ["\\ddot{z}" ]) ("Zdd" ["\\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
("gg"      quail-TeX-next           ) ; go to next space
("GG"      quail-TeX-prev           ) ; go to prev space

;; Symbols-dots
("..."   ["\\dots"                ] ) ; 3 dots
(".v"      ["\\vdots"                ] ) ; vertical dots
(".d"      ["\\ddots"                ] ) ; diagonale dots
(".l"      ["\\ldots"                ] ) ; low dots

;; Symbols-geo
("perp"    ["\\perp"                 ] ) ;
("perpn"   ["\\not\\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)
("dfd"      ["\\mathrm{d}"        ]) ;
("dfd."     ["\\partial"              ]) ; $\\mathrm{d}$ + _._ (var)
("dff"      ["\\frac{\\mathrm{d}}{\\mathrm{d}}"]) ;
("dff."     ["\\frac{\\partial}{\\partial}"]) ; $\\frac{\\mathrm{d}}{\\mathrm{d}}\\Box$ + .
("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"           ]) ;
("<->"     ["\\nleftrightharpoon"   ]) ;
("<->"     ["\\longrightarrow"       ]) ; longer arrows, with 2 dashes
("<->"     ["\\longleftarrow"        ]) ;

```

```

(|->"      ["\\mapsto"          ] ) ; vertical-bar + ~->~ (this might rendered wrong)
;; 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{"                  ] ) ; (var)
("dot.."    ["\\dddotted{"        ] ) ; (var)
("dot..." ["\\ddddotted{"            ] ) ; (var)
("dag"      ["^\\dagger"           ] ) ;
("dag."     ["^\\ddagger"              ] ) ; (var)
("*.."      ["^*"                      ] ) ;
("deg"      ["^\\circ"               ] ) ;
("tr"       ["^T"                ] ) ;
("trn"      ["^{-T}"              ] ) ; (neg)
;; Operation: arith
("+-"       ["\\pm"                  ] ) ;
("-+"       ["\\mp"                  ] ) ;
("*x"       ["\\times"                ] ) ;
("::"       ["\\div"                 ] ) ;
("**"       ["\\cdot"                 ] ) ;

```

;; Operation: arith

```

("<." ["\\leq" ]) ; < = >
(">." ["\\geq" ]) ; symbols
("<<" ["\\ll" ]) ;
(">>" ["\\gg" ]) ;
("=n" ["\\neq" ]) ; negation
("<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}" ]) ;
("=. " ["\\equiv" ]) ; Variations of =
("sim" ["\\sim" ]) ;
("simn" ["\\nsim" ]) ;
("=. ." ["\\approx" ]) ;
("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_{ }"      ]) ; (limits apparently doesn't render on Github pa
("il."      ["\\limits_{ }^{ }"   ]) ; . (var)
("lim"      ["\\lim"             ]) ;
("sum"      ["\\sum"             ]) ;
("prod"     ["\\prod"            ]) ;
("int"      ["\\int"             ]) ;
("inti"     ["\\iint"            ]) ; $\\int$ + i
("intii"    ["\\iiint"           ]) ; $\\int$ + ii
("intiii"   ["\\iiint"           ]) ; $\\int$ + iii
("into"     ["\\oint"            ]) ; $\\int$ + o
("sum."     ["\\sum\\limits_{ i=1 }^{ n }"]) ; . (var)
("prod."    ["\\prod\\limits_{ i=1 }^{ n }"]) ; . (var)
("int."     ["\\int\\limits_{ }^{ }"]) ; . (var)
("int.."    ["\\int\\limits_{ 0 }^{ +\\infty }"]) ; . (var)
("int..." ["\\int\\limits_{ -\\infty }^{ +\\infty }"]) ; . (var)
("inti."    ["\\iint\\limits_{ }"   ]) ; . (var)
("intii."   ["\\iiint\\limits_{ }"   ]) ; . (var)
("intiii."  ["\\iiint\\limits_{ }"   ]) ; . (var)
("into."    ["\\oint\\limits_{ }"   ]) ; . (var)
;; 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("                   ]) ; half (
(")."       ["\\right)"                ]) ; half )
("[."       ["\\left["                   ]) ; half [
("]."       ["\\right]"                ]) ; half ]

```



```

("{."      ["\\left\\{"          ]) ; half {
("}. "     ["\\right\\}"         ]) ; half }
("<."      ["\\left<"            ]) ; half <
(">."      ["\\right>"           ]) ; half >
("(.. "     ["\\left."            ]) ; half left .
(").. "     ["\\right."           ]) ; half right .
("|."       ["\\middle\\vert"       ]) ; Vertical bar related
("|.. "     ["\\Bigg\\vert_{ }^{ }"]) ; Vertical bar related
("||."      ["\\left\\vert \\right\\vert"]) ; Vertical bar related
("||.."     ["\\left\\Vert \\right\\Vert"]) ; Vertical bar related
;; 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
("ff"       ["\\frac{"         ]) ; Fractions
("can"      ["\\cancel{"         ]) ;
("=="      ["&=\\n"           ]) ; helps in align env.
;; 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-next (key idx)
  (quail-func-init)
  (evil-find-char 1 32)
  (quail-func-end))

```

```

(defun quail-TeX-prev (key idx)
  (quail-func-init)
  (evil-find-char-backward 1 32)
  (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))

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

```

10 Making the el

```

(require 'quail)
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

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

```



```

(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
  "TeQ-Math" "Emacs-Teq-Latex" "TeQ-" t
  "TeQ-Math input"
  nil t t t t 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" ])
("e.." ["\\varepsilon" ]) ("r.." ["\\varrho" ])
("f.." ["\\varphi" ]) ("p.." ["\\varpi" ])
("s.." ["\\varsigma" ]) ("t.." ["\\vartheta" ])
;; 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}"        ]) ("av"      ["\\vec{a}"        ])
("Bv"      ["\\vec{B}"        ]) ("bv"      ["\\vec{b}"        ])
("Cv"      ["\\vec{C}"        ]) ("cv"      ["\\vec{c}"        ])
("Dv"      ["\\vec{D}"        ]) ("dv"      ["\\vec{d}"        ])
("Ev"      ["\\vec{E}"        ]) ("ev"      ["\\vec{e}"        ])
("Fv"      ["\\vec{F}"        ]) ("fv"      ["\\vec{f}"        ])
("Gv"      ["\\vec{G}"        ]) ("gv"      ["\\vec{g}"        ])
("Hv"      ["\\vec{H}"        ]) ("hv"      ["\\vec{h}"        ])
("Iv"      ["\\vec{I}"        ]) ("iv"      ["\\vec{i}"        ])
("Jv"      ["\\vec{J}"        ]) ("jv"      ["\\vec{j}"        ])
("Kv"      ["\\vec{K}"        ]) ("kv"      ["\\vec{k}"        ])
("Lv"      ["\\vec{L}"        ]) ("lv"      ["\\vec{l}"        ])
("Mv"      ["\\vec{M}"        ]) ("mv"      ["\\vec{m}"        ])
("Nv"      ["\\vec{N}"        ]) ("nv"      ["\\vec{n}"        ])
("Ov"      ["\\vec{O}"        ]) ("ov"      ["\\vec{o}"        ])
("Pv"      ["\\vec{P}"        ]) ("pv"      ["\\vec{p}"        ])
("Qv"      ["\\vec{Q}"        ]) ("qv"      ["\\vec{q}"        ])
("Rv"      ["\\vec{R}"        ]) ("rv"      ["\\vec{r}"        ])
("Sv"      ["\\vec{S}"        ]) ("sv"      ["\\vec{s}"        ])
("Tv"      ["\\vec{T}"        ]) ("tv"      ["\\vec{t}"        ])
("Uv"      ["\\vec{U}"        ]) ("uv"      ["\\vec{u}"        ])
("Vv"      ["\\vec{V}"        ]) ("vv"      ["\\vec{v}"        ])
("Wv"      ["\\vec{W}"        ]) ("wv"      ["\\vec{w}"        ])
("Xv"      ["\\vec{X}"        ]) ("xv"      ["\\vec{x}"        ])
("Yv"      ["\\vec{Y}"        ]) ("yv"      ["\\vec{y}"        ])
("Zv"      ["\\vec{Z}"        ]) ("zv"      ["\\vec{z}"        ])
;; Vector & Hat
("Ah"      ["\\hat{A}"        ]) ("ah"      ["\\hat{a}"        ])
("Bh"      ["\\hat{B}"        ]) ("bh"      ["\\hat{b}"        ])
("Ch"      ["\\hat{C}"        ]) ("ch"      ["\\hat{c}"        ])
("Dh"      ["\\hat{D}"        ]) ("dh"      ["\\hat{d}"        ])
("Eh"      ["\\hat{E}"        ]) ("eh"      ["\\hat{e}"        ])
("Fh"      ["\\hat{F}"        ]) ("fh"      ["\\hat{f}"        ])
("Gh"      ["\\hat{G}"        ]) ("gh"      ["\\hat{g}"        ])
("Hh"      ["\\hat{H}"        ]) ("hh"      ["\\hat{h}"        ])
("Ih"      ["\\hat{I}"        ]) ("ih"      ["\\hat{i}"        ])

```

("Jh"	["\\hat{J}"])	("jh"	["\\hat{j}"])
("Kh"	["\\hat{K}"])	("kh"	["\\hat{k}"])
("Lh"	["\\hat{L}"])	("lh"	["\\hat{l}"])
("Mh"	["\\hat{M}"])	("mh"	["\\hat{m}"])
("Nh"	["\\hat{N}"])	("nh"	["\\hat{n}"])
("Oh"	["\\hat{O}"])	("oh"	["\\hat{o}"])
("Ph"	["\\hat{P}"])	("ph"	["\\hat{p}"])
("Qh"	["\\hat{Q}"])	("qh"	["\\hat{q}"])
("Rh"	["\\hat{R}"])	("rh"	["\\hat{r}"])
("Sh"	["\\hat{S}"])	("sh"	["\\hat{s}"])
("Th"	["\\hat{T}"])	("th"	["\\hat{t}"])
("Uh"	["\\hat{U}"])	("uh"	["\\hat{u}"])
("Vh"	["\\hat{V}"])	("vh"	["\\hat{v}"])
("Wh"	["\\hat{W}"])	("wh"	["\\hat{w}"])
("Xh"	["\\hat{X}"])	("xh"	["\\hat{x}"])
("Yh"	["\\hat{Y}"])	("yh"	["\\hat{y}"])
("Zh"	["\\hat{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
("add"      ["\\ddot{a}"      ]) ("Add"      ["\\ddot{A}"      ])
("bdd"      ["\\ddot{b}"      ]) ("Bdd"      ["\\ddot{B}"      ])
("cdd"      ["\\ddot{c}"      ]) ("Cdd"      ["\\ddot{C}"      ])
("ddd"      ["\\ddot{d}"      ]) ("Ddd"      ["\\ddot{D}"      ])
("edd"      ["\\ddot{e}"      ]) ("Edd"      ["\\ddot{E}"      ])
("fdd"      ["\\ddot{f}"      ]) ("Fdd"      ["\\ddot{F}"      ])
("gdd"      ["\\ddot{g}"      ]) ("Gdd"      ["\\ddot{G}"      ])
("hdd"      ["\\ddot{h}"      ]) ("Hdd"      ["\\ddot{H}"      ])
("idd"      ["\\ddot{i}"      ]) ("Idd"      ["\\ddot{I}"      ])
("jdd"      ["\\ddot{j}"      ]) ("Jdd"      ["\\ddot{J}"      ])
("kdd"      ["\\ddot{k}"      ]) ("Kdd"      ["\\ddot{K}"      ])
("ldd"      ["\\ddot{l}"      ]) ("Ldd"      ["\\ddot{L}"      ])
("mdd"      ["\\ddot{m}"      ]) ("Mdd"      ["\\ddot{M}"      ])
("ndd"      ["\\ddot{n}"      ]) ("Ndd"      ["\\ddot{N}"      ])
("odd"      ["\\ddot{o}"      ]) ("Odd"      ["\\ddot{O}"      ])
("pdd"      ["\\ddot{p}"      ]) ("Pdd"      ["\\ddot{P}"      ])
("qdd"      ["\\ddot{q}"      ]) ("Qdd"      ["\\ddot{Q}"      ])
("rdd"      ["\\ddot{r}"      ]) ("Rdd"      ["\\ddot{R}"      ])
("sdd"      ["\\ddot{s}"      ]) ("Sdd"      ["\\ddot{S}"      ])
("tdd"      ["\\ddot{t}"      ]) ("Tdd"      ["\\ddot{T}"      ])
("udd"      ["\\ddot{u}"      ]) ("Udd"      ["\\ddot{U}"      ])
("vdd"      ["\\ddot{v}"      ]) ("Vdd"      ["\\ddot{V}"      ])
("wdd"      ["\\ddot{w}"      ]) ("Wdd"      ["\\ddot{W}"      ])
("xdd"      ["\\ddot{x}"      ]) ("Xdd"      ["\\ddot{X}"      ])
("ydd"      ["\\ddot{y}"      ]) ("Ydd"      ["\\ddot{Y}"      ])
("zdd"      ["\\ddot{z}"      ]) ("Zdd"      ["\\ddot{Z}"      ])
;; Expanding Func
("/"      quail-TeX-frac      ) ; fraction on previous
("eq"      quail-TeX-equation ) ; equation environment
("al"      quail-TeX-aligned  ) ; aligned environment
("el"      quail-TeX-endofline ) ; end of line
("gg"      quail-TeX-next     ) ; go to next space
("GG"      quail-TeX-prev     ) ; go to prev space
;; Symbols-dots
("..."   ["\\dots"         ]) ; 3 dots

```

```

(".v"      ["\\vdots"          ]) ; vertical dots
(".d"      ["\\ddots"          ]) ; diagonale dots
(".l"      ["\\ldots"          ]) ; low dots
;; Symbols-geo
("perp"    ["\\perp"           ]) ;
("perpn"   ["\\not\\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)
("dfd"     ["\\mathrm{d}"        ]) ;
("dfd."    ["\\partial"              ]) ;  $\mathrm{d}$  + _._ (var)
("dff"     ["\\frac{\\mathrm{d}}{\\mathrm{d}}"]) ;
("dff."    ["\\frac{\\partial}{\\partial}"]) ;  $\frac{\mathrm{d}}{\mathrm{d}}\Box$ 
("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

```

```

("<->"      ["\\leftrightarrow"    ]) ;
("<-n"      ["\\nleftarrow"        ]) ; negate (~n~) of prev. section
(">n"       ["\\rightarrow"         ]) ; arrows + _n_
("-^n"      ["\\nuparrow"          ]) ;
("-vn"      ["\\ndownarrow"        ]) ;
("<->"      ["\\nleftrightarrow"    ]) ;
("-->"      ["\\longrightarrow"     ]) ; longer arrows, with 2 dashes
("<--"      ["\\longleftarrow"     ]) ;
("|->"      ["\\mapsto"           ]) ; vertical-bar + ~->~ (this might rendered
;; 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"        ]) ;
("<==>"    ["\\Longlefttrightarrow"    ]) ; longer arrows, with 2 dashes
("<=="     ["\\Longleftarrow"       ]) ;
("==>"     ["\\Longrightarrow"        ]) ;
;; 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{"                 ]) ; (var)
("dot.."    ["\\dddot{"            ]) ; (var)
("dot..." ["\\ddddot{"             ]) ; (var)
("dag"      ["^\\dagger"            ]) ;
("dag."     ["^\\ddagger"            ]) ; (var)
("*.."      ["^*"                     ]) ;
("deg"      ["^\\circ"              ]) ;

```

```

(tr"      ["^T"                ]) ;
(trn"     ["^{-T}"             ]) ; (neg)
;; Operation: arith
("+-"     ["\\pm"              ]) ;
("-+"     ["\\mp"              ]) ;
("x"      ["\\times"           ]) ;
(":"      ["\\div"             ]) ;
("**"     ["\\cdot"            ]) ;
;; Operation: arith
("<."     ["\\leq"              ]) ; < = >
(">."     ["\\geq"              ]) ; symbols
("<<"     ["\\ll"               ]) ;
(">>"     ["\\gg"               ]) ;
("=n"     ["\\neq"              ]) ; negation
("<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}"]) ;
("=. "    ["\\equiv"            ]) ; Variations of =
("sim"     ["\\sim"              ]) ;
("simn"    ["\\nsim"            ]) ;
("=. ."    ["\\approx"           ]) ;
("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."   ["\\subseteq"   ]) ; (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_{ }"          ]) ; (limits apparently doesn't render on Github)
("il."      ["\\limits_{ }^{ }"      ]) ; . (var)
("lim"      ["\\lim"                 ]) ;
("sum"      ["\\sum"                 ]) ;
("prod"     ["\\prod"                ]) ;
("int"      ["\\int"                 ]) ;
("inti"     ["\\iint"                ]) ;  $\int + i$ 
("intii"    ["\\iiint"               ]) ;  $\int + ii$ 
("intiii"   ["\\iiint"               ]) ;  $\int + iii$ 
("into"     ["\\oint"                ]) ;  $\int + o$ 
("sum."     ["\\sum\\limits_{ i=1 }^{ n }"]) ; . (var)
("prod."    ["\\prod\\limits_{ i=1 }^{ n }"]) ; . (var)
("int."     ["\\int\\limits_{ }^{ }"]) ; . (var)
("int.."    ["\\int\\limits_{ 0 }^{ +\\infty }"]) ; . (var)
("int..." ["\\int\\limits_{ -\\infty }^{ +\\infty }"]) ; . (var)
("inti."    ["\\iint\\limits_{ }"]    ]) ; . (var)
("intii."   ["\\iiint\\limits_{ }"]   ]) ; . (var)
("intiii."  ["\\iiint\\limits_{ }"]   ]) ; . (var)
("into."    ["\\oint\\limits_{ }"]    ]) ; . (var)
;; 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(" ] ) ; half (
(")." ["\\right)"] ] ) ; half )
("[." ["\\left[" ] ) ; half [
("]." ["\\right]"] ] ) ; half ]
("{." ["\\left\\{" ] ) ; half {
("}." ["\\right\\}"] ] ) ; half }
("<." ["\\left<"] ] ) ; half <
(">." ["\\right>"] ] ) ; half >
("(.." ["\\left." ] ) ; half left .
(").." ["\\right." ] ) ; half right .
("|." ["\\middle\\vert"] ] ) ; Vertical bar related
("|.." ["\\Bigg\\vert_{ }^{ }"] ) ; Vertical bar related
("||." ["\\left\\vert \\right\\vert"] ) ; Vertical bar related
("||.." ["\\left\\Vert \\right\\Vert"] ) ; Vertical bar related
;; 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"] ) ("lnn" ["_n"] )
("px" ["^x"] ) ("li" ["_i"] )
("__" ["\\underset{ }{ }"] ) ("^^" ["\\overset{ }{ }"] )
("__." ["\\underbrace{ }_{ }"] ) ("^^." ["\\overbrace{ }^{ }"] )

```

```

("__.." ["\\underline{ }" ] ) ("^^.." ["\\overline{ }" ] )
;; Structural: misc
("binom" ["\\binom{" ] ) ; Binom
("box" ["\\boxed{" ] ) ; Putting box around object
("ff" ["\\frac{" ] ) ; Fractions
("can" ["\\cancel{" ] ) ;
("==" ["&=\\\\" ] ) ; helps in align env.
;; Structural: xy
("xy" ["\\xymatrix{\\n\\n}" ] ) ;
("bu" ["\\bullet" ] ) ;
("ar" ["\\ar" ] ) ;
)

```