

Emacs TeQ: (T_EX + Quail)

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

Garid Zorigoo

April 11, 2023

Contents

1	Alphabet related stuff :	1
1.1	Greek	1
1.2	Matrix (aka bold)	3
1.3	Vector & Hat	4
1.4	Dots	5
1.5	DDots	6
2	Function Expansion	7
3	Symbols :	7
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:	9
3.5.3	Long arrow with top-bottom entries	10
4	Symbol Modification	10
4.1	Accents (variable decoration?)	10
4.2	Superscripts & Subscripts (power & lower)	11
5	Binary Operation Symbols	11
5.1	Simple Arithmetics:	11
5.2	Binary Relations:	13
5.3	Set symbols	14

5.4	Logic	14
6	Functions	15
6.1	Function	15
6.2	Trigonometry: function	15
6.3	Iterative-like operation:	16
7	Structural:	17
7.1	Parenthesis Related	17
7.2	Texts:	17
7.3	Misc.	18
7.4	xy Diagram related	18
8	Formatting Table into Elisp	18
9	Executable elisp function definition	29
10	Making the el	31

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>

Table 2: Variation Greek letters

key	sym	latex (lower greek)
e..	ε	<code>\varepsilon</code>
f..	φ	<code>\varphi</code>
s..	ς	<code>\varsigma</code>
t..	ϑ	<code>\vartheta</code>
r..	ϱ	<code>\varrho</code>
p..	ϖ	<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	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 & 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	
para	<code>\parallel</code>	\parallel	
paran	<code>\nparallel</code>	\nparallel	
ang	<code>\angle</code>	\angle	
ang.	<code>\measuredangle</code>	\sphericalangle	

3.3 Letter like

Table 10: Letter-like Symbol

key	trans	sym	description
inf	<code>\infty</code>	∞	
ex	<code>\exists</code>	\exists	
ex.	<code>\nexists</code>	\nexists	
fa	<code>\forall</code>	\forall	
hb	<code>\hbar</code>	\hbar	
hb.	<code>\hslash</code>	\hslash	
dd	<code>\mathrm{d}</code>	d	
dd.	<code>\partial</code>	∂	
ii	<code>\imath</code>	\imath	
jj	<code>\jmath</code>	\jmath	
nab	<code>\nabla</code>	∇	
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	left arrow
->	<code>\rightarrow</code>	\rightarrow	right arrow
-^	<code>\uparrow</code>	\uparrow	up arrow
-v	<code>\downarrow</code>	\downarrow	down arrow
<->	<code>\leftrightarrow</code>	\leftrightarrow	left-right arrow
<-n	<code>\nleftarrow</code>	\nleftarrow	not left arrow
->n	<code>\nrightarrow</code>	\nrightarrow	not right arrow
-^n	<code>\nuparrow</code>	\nuparrow	not up arrow
-vn	<code>\ndownarrow</code>	\ndownarrow	not down arrow
<->	<code>\nleftrightarrow</code>	\nleftrightarrow	not left-right arrow
-->	<code>\longrightarrow</code>	\longrightarrow	
<--	<code>\longleftarrow</code>	\longleftarrow	
->	<code>\mapsto</code>	\mapsto	

3.5.2 Double:

Table 13: Double Line arrows

key	trans	sym	description
<=	<code>\Leftarrow</code>	\Leftarrow	left arrow
=>	<code>\Rightarrow</code>	\Rightarrow	right arrow
=^	<code>\Uparrow</code>	\Uparrow	up arrow
=v	<code>\Downarrow</code>	\Downarrow	down arrow
<=>	<code>\Leftrightarrow</code>	\Leftrightarrow	left-right arrow
iff	<code>\Leftrightarrow</code>	\Leftrightarrow	left-right arrow
<=n	<code>\nLeftarrow</code>	\nLeftarrow	left arrow
=>n	<code>\nRightarrow</code>	\nRightarrow	right arrow
<=>n	<code>\nLeftrightarrow</code>	\nLeftrightarrow	left-right arrow
iffn	<code>\nLeftrightarrow</code>	\nLeftrightarrow	left-right arrow
<==>	<code>\Longleftrightarrow</code>	\Longleftrightarrow	left-right arrow
<==	<code>\Longleftarrow</code>	\Longleftarrow	left-right arrow
==>	<code>\Longrightarrow</code>	\Longrightarrow	left-right arrow

3.5.3 Long arrow with top-bottom entries

Table 14: Long arrow Line arrows

key	trans	sym	description
<--	<code>\xleftarrow[]{ }</code>	$\xleftarrow{\quad}$	
-->	<code>\xrightarrow[]{ }</code>	$\xrightarrow{\quad}$	
==>	<code>\xRightarrow[]{ }</code>	$\xRightarrow{\quad}$	mathtools lib required
<==	<code>\xLeftarrow[]{ }</code>	$\xLeftarrow{\quad}$	mathtools lib required

4 Symbol Modification

4.1 Accents (variable decoration?)

Table 15:

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

4.2 Superscripts & Subsripts (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	~ 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	$\backslash \underset{\square}{\square} \{ \} \{ \}$	^^	\square^\square	$\backslash \overset{\square}{\square} \{ \} \{ \}$
---	\square_\square	$\backslash \underbrace{\square}_\square \{ \} _ \{ \}$	^^.	\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
=n	<code>\neq</code>	\neq	
=.	<code>\equiv</code>	\equiv	
=?	<code>\stackrel{?}{=}</code>	$\stackrel{?}{=}$	
=y	<code>\stackrel{\checkmark}{=}</code>	$\stackrel{\checkmark}{=}$	
3=	<code>\equiv</code>	\equiv	
:=	<code>\coloneqq</code>	\coloneqq	
:=	<code>\coloneqq</code>	\coloneqq	
~.	<code>\sim</code>	\sim	
~n	<code>\nsim</code>	\nsim	
~~	<code>\approx</code>	\approx	
<n	<code>\nless</code>	\nless	
<.	<code>\leq</code>	\leq	
<.n	<code>\nleq</code>	\nleq	
<?	<code>\stackrel{?}{<}</code>	$\stackrel{?}{<}$	
<y	<code>\stackrel{\checkmark}{<}</code>	$\stackrel{\checkmark}{<}$	
<.?	<code>\stackrel{?}{<}\leq</code>	$\stackrel{?}{<}\leq$	
<.y	<code>\stackrel{\checkmark}{<}\leq</code>	$\stackrel{\checkmark}{<}\leq$	
«	<code>\ll</code>	\ll	
«?	<code>\stackrel{?}{\ll}</code>	$\stackrel{?}{\ll}$	
«y	<code>\stackrel{\checkmark}{\ll}</code>	$\stackrel{\checkmark}{\ll}$	
>n	<code>\ngtr</code>	\ngtr	
>.	<code>\geq</code>	\geq	
>.n	<code>\ngeq</code>	\ngeq	
>?	<code>\stackrel{?}{>}</code>	$\stackrel{?}{>}$	
>y	<code>\stackrel{\checkmark}{>}</code>	$\stackrel{\checkmark}{>}$	
>.?	<code>\stackrel{?}{>}\geq</code>	$\stackrel{?}{>}\geq$	
>.y	<code>\stackrel{\checkmark}{>}\geq</code>	$\stackrel{\checkmark}{>}\geq$	
»	<code>\gg</code>	\gg	
»?	<code>\stackrel{?}{\gg}</code>	$\stackrel{?}{\gg}$	
»y	<code>\stackrel{\checkmark}{\gg}</code>	$\stackrel{\checkmark}{\gg}$	

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	
0/	<code>\emptyset</code>	\emptyset	
nsr	<code>\mathbb{R}</code>	\mathbb{R}	
nsc	<code>\mathbb{C}</code>	\mathbb{C}	
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>	$\not\subseteq$	
sub=	<code>\subseteq</code>	\subseteq	
sub=n	<code>\nsubseteq</code>	$\not\subseteq$	
subn=	<code>\nsubseteq</code>	$\not\subseteq$	
sup	<code>\supset</code>	\supset	
supn	<code>\nsupseteq</code>	$\not\supseteq$	
sup=	<code>\supseteq</code>	\supseteq	
sup=n	<code>\nsupseteq</code>	$\not\supseteq$	
supn=	<code>\nsupseteq</code>	$\not\supseteq$	

5.4 Logic

Table 20:

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

6 Functions

6.1 Function

Table 21:

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

6.2 Trigonometry: function

Table 22:

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

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)	parenthesis
()..	<code>\left(\middle\vert \right)</code>	$(\square \square)$	parenthesis
[] .	<code>\left[\right]</code>	$[\square]$	parenthesis
[]..	<code>\left[\middle\vert \right]</code>	$[\square \square]$	parenthesis
[] .c	<code>\lceil \rceil</code>	$[\square]$	parenthesis (ceil)
[] .f	<code>\lfloor \rfloor</code>	$[\square]$	parenthesis (floor)
{ } .	<code>\left\{ \right\}</code>	$\{\square\}$	parenthesis
{ }..	<code>\left\{ \middle\vert \right\}</code>	$\{\square \square\}$	parenthesis (set maker)
< > .	<code>\left< \right></code>	$\langle \square \rangle$	parenthesis
< >..	<code>\left< \middle\vert \right></code>	$\langle \square \square \rangle$	parenthesis
.	<code>\left\vert \right\vert</code>	$ \square $	parenthesis (abs)
..	<code>\left\ \right\ </code>	$ \square $	parenthesis (abs)
(.	<code>\left(</code>	$(\square$	half-parenthesis
) .	<code>\right)</code>	$\square)$	half-parenthesis
[.	<code>\left[</code>	$[\square$	half-parenthesis
] .	<code>\right]</code>	$\square]$	half-parenthesis
{ .	<code>\left\{</code>	$\{\square$	half-parenthesis
} .	<code>\right\}</code>	$\square\}$	half-parenthesis
< .	<code>\left<</code>	$\langle \square$	half-parenthesis
> .	<code>\right></code>	$\square \rangle$	half-parenthesis
.	<code>\Bigg\vert_{\{ \}^{\{ \}}</code>	\square	definite integral range

7.2 Texts:

Table 25:

key	trans	sym	description
te	<code>\text{</code>	$\square + \text{text}$	normal text
tr	<code>\mathrm{</code>	$\square + \text{mathrm}$	math roman (used for sin,cos,tan ...)
tb	<code>\mathbf{</code>	$\square + \mathbf{mathbf}$	math bold
ti	<code>\mathit{</code>	$\square + \textit{mathit}$	math italics

7.3 Misc.

Table 26:

key	trans	sym	description
binom	<code>\binom{ }{ }</code>	$\binom{\square}{\square}$	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>&=</code>	<code>&=\n\\</code>		
<code>=&</code>	<code>&=\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("~", "")
        key1 = repr(key1).replace("\\'", "\'").replace("~", "")
        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("~", "")
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("~", "")
        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." ["\\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"               ] ) ;
("para"    ["\\parallel"           ] ) ;
("paran"   ["\\nparallel"          ] ) ;
("ang"     ["\\angle"              ] ) ;
("ang."    ["\\measuredangle"      ] ) ;

```



```

;; Symbols
("inf"      ["\\infty"      ]) ;
("ex"       ["\\exists"     ]) ;
("ex."      ["\\nexists"    ]) ;
("fa"       ["\\forall"     ]) ;
("hb"       ["\\hbar"       ]) ;
("hb."      ["\\hslash"     ]) ;
("dd"       ["\\mathrm{d}"   ]) ;
("dd."      ["\\partial"    ]) ;
("ii"       ["\\imath"      ]) ;
("jj"       ["\\jmath"      ]) ;
("nab"      ["\\nabla"      ]) ;
("cm"       ["\\checkmark"   ]) ;

;; Symbols spaces
("qu"       ["\\quad"       ]) ;
("quu"      ["\\qquad"      ]) ;

;; Symbols arrow1
("<-"      ["\\leftarrow"      ]) ; left arrow
(">-"      ["\\rightarrow"     ]) ; right arrow
("<^-"      ["\\uparrow"         ]) ; up arrow
("<-v"      ["\\downarrow"       ]) ; down arrow
("<->"      ["\\leftrightarrow"   ]) ; left-right arrow
("<-n"      ["\\nleftarrow"    ]) ; not left arrow
(">-n"      ["\\nrightarrow"   ]) ; not right arrow
("<-^n"      ["\\nuparrow"        ]) ; not up arrow
("<-vn"      ["\\ndownarrow"     ]) ; not down arrow
("<->"      ["\\nleftrightarrow" ]) ; not left-right arrow
("<-->"     ["\\longrightarrow"     ]) ;
("<-->"     ["\\longleftarrow"      ]) ;
("\\vert ->" ["\\mapsto"             ]) ;

;; Symbols arrow2
("<="      ["\\Leftarrow"        ]) ; left arrow
(">="      ["\\Rightarrow"      ]) ; right arrow
("<^="      ["\\Uparrow"          ]) ; up arrow
("<v="      ["\\Downarrow"        ]) ; down arrow
("<=>"      ["\\Leftrightarrow"     ]) ; left-right arrow
("iff"      ["\\Leftrightarrow"     ]) ; left-right arrow
("<=n"      ["\\nLeftarrow"        ]) ; left arrow
(">=n"      ["\\nRightarrow"       ]) ; right arrow
("<=>n"     ["\\nLeftrightarrow"   ]) ; left-right arrow

```

```

("iffn"      ["\\nLeftrightarrow"  ]) ; left-right arrow
("<==>"     ["\\Longlefttrightarrow"]) ; left-right arrow
("<==>"     ["\\Longleftarrow"      ]) ; left-right arrow
("==>"      ["\\Longrightarrow"     ]) ; left-right arrow
;; Symbols arrow3
("<---"     ["\\xleftarrow[ ]{ }"  ]) ;
("--->"     ["\\xrightarrow[ ]{ }" ]) ;
("===>"     ["\\xRrightarrow[ ]{ }" ]) ; ~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
("=n"       ["\\neq"             ]) ;
("=. "      ["\\equiv"          ]) ;
("=?"       ["\\stackrel{?}{=}"   ]) ;
("=y"       ["\\stackrel{\\checkmark}{=}" ]) ;
("3="       ["\\equiv"          ]) ;
("=: "      ["\\coloneqq"        ]) ;
(":= "      ["\\coloneqq"        ]) ;
("=.="      ["\\sim"            ]) ;
("=n="      ["\\nsim"            ]) ;
(" "        ["\\approx"          ]) ;

```

```

("<n"      ["\\nless"          ]) ;
("<."      ["\\leq"           ]) ;
("<.n"     ["\\nleq"            ]) ;
("<?"      ["\\stackrel{?}{<}"]    ]) ;
("<y"      ["\\stackrel{\\checkmark}{<}"]) ;
("<.??"    ["\\stackrel{?}{\\leq}"]) ;
("<.y"     ["\\stackrel{\\checkmark}{\\leq}"]) ;
("<<"      ["\\ll"              ]) ;
("<<?"     ["\\stackrel{?}{\\ll}"]   ]) ;
("<<y"     ["\\stackrel{\\checkmark}{\\ll}"]) ;
(">n"      ["\\ngtr"           ]) ;
(">."      ["\\geq"           ]) ;
(">.n"     ["\\ngeq"          ]) ;
(">?"      ["\\stackrel{?}{>}"]  ]) ;
(">y"      ["\\stackrel{\\checkmark}{>}"]) ;
(">.??"    ["\\stackrel{?}{\\geq}"]) ;
(">.y"     ["\\stackrel{\\checkmark}{\\geq}"]) ;
(">>"      ["\\gg"              ]) ;
(">>?"     ["\\stackrel{?}{\\gg}"]   ]) ;
(">>y"     ["\\stackrel{\\checkmark}{\\gg}"]) ;
;; Operation: arith
("in"      ["\\in"           ]) ;
("in."     ["\\ni"            ]) ;
("ni"      ["\\ni"            ]) ;
("inn"     ["\\notin"        ]) ;
("0/"      ["\\emptyset"        ]) ;
("nsr"     ["\\mathbb{R}"       ]) ;
("nsc"     ["\\mathbb{C}"       ]) ;
("nsn"     ["\\mathbb{N}"       ]) ;
("nsp"     ["\\mathbb{P}"       ]) ;
("nsz"     ["\\mathbb{Z}"       ]) ;
("nsi"     ["\\mathbb{I}"       ]) ;
("sub"     ["\\subset"        ]) ;
("subn"    ["\\nssubseteq"     ]) ;
("sub="    ["\\subseteq"      ]) ;
("sub=n"   ["\\nsubseteq"      ]) ;
("subn="   ["\\nsubseteq"      ]) ;
("sup"     ["\\supset"         ]) ;
("supn"    ["\\nsupseteq"       ]) ;
("sup="    ["\\supeseteq"     ]) ;

```

```

("sup=n" ["\\nsupseteq" ]) ;
("supn=" ["\\nsupseteq" ]) ;
;; Operation: arith
("or" ["\\lor" ]) ;
("and" ["\\land" ]) ;
("not" ["\\neg" ]) ;
("or." ["\\text{ or }" ]) ;
("and." ["\\text{ and }" ]) ;
("not." ["\\text{ not }" ]) ;
;; 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)"      ] ; parenthesis
("()..)"    ["\\left( \\middle\\vert \\right)"] ; parenthesis
("[].)"     ["\\left[ \\right]"       ] ; parenthesis
("[].)"     ["\\left[ \\middle\\vert \\right)"] ; parenthesis
("[].c)"    ["\\lceil \\rceil"         ] ; parenthesis (ceil)
("[].f)"    ["\\lfloor \\rfloor"      ] ; parenthesis (floor)
("{}.)"     ["\\left\\{ \\right\\}"]   ] ; parenthesis
("{}.."     ["\\left\\{ \\middle\\vert \\right\\}"] ; parenthesis (set maker)
("<.>)"     ["\\left< \\right>"        ] ; parenthesis
("<.>.."     ["\\left< \\middle\\vert \\right>"] ; parenthesis
("\\vert\\vert ." ["\\left\\vert \\right\\vert"] ; parenthesis (abs)
("\\vert\\vert .." ["\\left\\Vert \\right\\Vert"] ; parenthesis (abs)
("(.)"      ["\\left("                ]) ; half-parenthesis
(").)"      ["\\right)"              ]) ; half-parenthesis
("[.]"      ["\\left["                ]) ; half-parenthesis
("].)"      ["\\right]"              ]) ; half-parenthesis
("{.}"      ["\\left\\"               ]) ; half-parenthesis
("}.)"      ["\\right\\"             ]) ; half-parenthesis
("<."       ["\\left<"                ]) ; half-parenthesis
(">."       ["\\right>"              ]) ; half-parenthesis
("\\vert ."  ["\\Big\\vert_{ }^{ }"]   ) ; definite integral range
;; Structural: Text
("te"       ["\\text{"                ]) ; normal text
("tr"       ["\\mathrm{"             ]) ; math roman (used for sin,cos,tan ...)
("tb"       ["\\mathbf{"             ]) ; math bold

```

```

("ti"      ["\\mathit{"          ]) ; math italics
;; Structural: Text
("te"      ["\\text{"           ]) ; normal text
("tr"      ["\\mathrm{"         ]) ; math roman (used for sin,cos,tan ...)
("tb"      ["\\mathbf{"         ]) ; math bold
("ti"      ["\\mathit{"         ]) ; math italics
;; 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
("fr"      ["\\frac{ }{ "       ]) ; Fractions
("can"     ["\\cancel"          ]) ;
("&="      ["&=\\n\\\\\\\\\\\\\\\\" ]) ;
("=&"     ["&=\\n\\\\\\\\\\\\\\\\" ]) ;
;; Structural: xy
("xy"      ["\\xymatrix{\\n\\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-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 "}{"))
      )
  )
)

```



```

(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))

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

(quail-define-package
 "TeX-Math" "Emacs-TeX-Latex" "TeX-" t
 "TeX-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"])
;; <i>Matrix</i>					
("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" ]) ;
("para"  ["\\parallel" ]) ;
("paran" ["\\nparallel" ]) ;
("ang"   ["\\angle" ]) ;
("ang."  ["\\measuredangle" ]) ;

;; Symbols
("inf"   ["\\infty" ]) ;
("ex"    ["\\exists" ]) ;
("ex."   ["\\nexists" ]) ;
("fa"    ["\\forall" ]) ;
("hb"    ["\\hbar" ]) ;
("hb."   ["\\hslash" ]) ;
("dd"    ["\\mathrm{d}" ]) ;
("dd."   ["\\partial" ]) ;
("ii"    ["\\imath" ]) ;
("jj"    ["\\jmath" ]) ;
("nab"   ["\\nabla" ]) ;
("cm"    ["\\checkmark" ]) ;

;; Symbols spaces
("qu"    ["\\quad" ]) ;

```

```

("quu"      ["\\qquad"          ]) ;
;; Symbols arrow1
("<-"      ["\\leftarrow"        ]) ; left arrow
(">-"      ["\\rightarrow"        ]) ; right arrow
("-^"       ["\\uparrow"          ]) ; up arrow
("-v"       ["\\downarrow"         ]) ; down arrow
("<->"     ["\\leftrightharrow"         ]) ; left-right arrow
("<-n"     ["\\nleftarrow"             ]) ; not left arrow
(">-n"     ["\\nrightarrow"            ]) ; not right arrow
("-^n"      ["\\nuparrow"               ]) ; not up arrow
("-vn"      ["\\ndownarrow"            ]) ; not down arrow
("<->n"    ["\\nleftrightharrow"        ]) ; not left-right arrow
("-->"     ["\\longrightarrow"         ]) ;
("<--"     ["\\longleftarrow"          ]) ;
("\\vert ->" ["\\mapsto"                 ]) ;
;; Symbols arrow2
("<="      ["\\Leftarrow"              ]) ; left arrow
(">="      ["\\Rightarrow"            ]) ; right arrow
("=^"      ["\\Uparrow"                ]) ; up arrow
("=v"      ["\\Downarrow"             ]) ; down arrow
("<=>"     ["\\Leftrightarrow"         ]) ; left-right arrow
("iff"     ["\\Leftrightarrow"         ]) ; left-right arrow
("<=n"     ["\\nLeftarrow"            ]) ; left arrow
(">=n"     ["\\nRightarrow"           ]) ; right arrow
("<=>n"    ["\\nLeftrightarrow"        ]) ; left-right arrow
("iffn"    ["\\nLeftrightarrow"        ]) ; left-right arrow
("<==>"    ["\\Longleftrightarrow"     ]) ; left-right arrow
("<=="     ["\\Longleftarrow"          ]) ; left-right arrow
("==>"     ["\\Longrightarrow"         ]) ; left-right arrow
;; Symbols arrow3
("<---"     ["\\xleftarrow[ ]{ }"       ]) ;
("---->"    ["\\xrightarrow[ ]{ }"      ]) ;
("===>"     ["\\xRightarrow[ ]{ }"       ]) ; ~mathtools~ lib required
("<===>"    ["\\xLeftrightarrow[ ]{ }"  ]) ; ~mathtools~ lib required
;; Symbols arrow3
("vec"      ["\\vec{"                   ]) ;
("bar"      ["\\bar{"                   ]) ;
("hat"      ["\\hat{"                   ]) ;
("dot"      ["\\dot{"                   ]) ;
("dot."     ["\\ddot{"                  ]) ;

```

```

(dot.." ["\\dddot{" ] ) ;
(dot..." ["\\ddddot{" ] ) ;
(dag" ["^\\dagger" ] ) ;
(dag." ["^\\ddagger" ] ) ;
(*.." ["^*" ] ) ;
(deg" ["^\\circ" ] ) ;
(tr" ["^T" ] ) ;
(tr." ["^{-T}" ] ) ;
;; Operation: arith
(+-" ["\\pm" ] ) ;
(-+" ["\\mp" ] ) ;
(*x" ["\\times" ] ) ;
(::" ["\\div" ] ) ;
(**" ["\\cdot" ] ) ;
;; Operation: arith
(=n" ["\\neq" ] ) ;
(=. " ["\\equiv" ] ) ;
(=?" ["\\stackrel{?}{=}" ] ) ;
(=y" ["\\stackrel{\\checkmark}{=}" ] ) ;
(3=" ["\\equiv" ] ) ;
(=: " ["\\coloneqq" ] ) ;
(:.= " ["\\coloneqq" ] ) ;
(=. = " ["\\sim" ] ) ;
(=n= " ["\\nsim" ] ) ;
(" " ["\\approx" ] ) ;
(<n" ["\\nless" ] ) ;
(<. " ["\\leq" ] ) ;
(<.n" ["\\nleq" ] ) ;
(<?" ["\\stackrel{?}{<}" ] ) ;
(<y" ["\\stackrel{\\checkmark}{<}" ] ) ;
(<.?" ["\\stackrel{?}{\\leq}" ] ) ;
(<.y" ["\\stackrel{\\checkmark}{\\leq}" ] ) ;
(<<" ["\\ll" ] ) ;
(<<?" ["\\stackrel{?}{\\ll}" ] ) ;
(<<y" ["\\stackrel{\\checkmark}{\\ll}" ] ) ;
(>n" ["\\ngtr" ] ) ;
(>. " ["\\geq" ] ) ;
(>.n" ["\\ngeq" ] ) ;
(>?" ["\\stackrel{?}{>}" ] ) ;
(>y" ["\\stackrel{\\checkmark}{>}" ] ) ;

```

```

(">."      ["\\stackrel{?}{\\geq}"]) ;
(">.y"     ["\\stackrel{\\checkmark}{\\geq}"]) ;
(">>"      ["\\gg" ]) ;
(">>?"     ["\\stackrel{?}{\\gg}" ]) ;
(">>y"      ["\\stackrel{\\checkmark}{\\gg}"]) ;
;; Operation: arith
("in"       ["\\in" ]) ;
("in."      ["\\ni" ]) ;
("ni"       ["\\ni" ]) ;
("inn"      ["\\notin" ]) ;
("O/"       ["\\emptyset" ]) ;
("nsr"      ["\\mathbb{R}" ]) ;
("nsc"      ["\\mathbb{C}" ]) ;
("nsn"      ["\\mathbb{N}" ]) ;
("nsp"      ["\\mathbb{P}" ]) ;
("nsz"      ["\\mathbb{Z}" ]) ;
("nsi"      ["\\mathbb{I}" ]) ;
("sub"      ["\\subset" ]) ;
("subn"     ["\\nssubseteq" ]) ;
("sub="     ["\\subseteq" ]) ;
("sub=n"    ["\\nsubseteq" ]) ;
("subn="    ["\\nsubseteq" ]) ;
("sup"      ["\\supset" ]) ;
("supn"     ["\\nsupseteq" ]) ;
("sup="     ["\\supeseteq" ]) ;
("sup=n"    ["\\nsupseteq" ]) ;
("supn="    ["\\nsupseteq" ]) ;
;; Operation: arith
("or"       ["\\lor" ]) ;
("and"      ["\\land" ]) ;
("not"      ["\\neg" ]) ;
("or."      ["\\text{ or }" ]) ;
("and."     ["\\text{ and }" ]) ;
("not."     ["\\text{ not }" ]) ;
;; 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)"      ]) ; parenthesis
("()..."     ["\\left( \\middle\\vert \\right)")] ; parenthesis
("[]."       ["\\left[ \\right]"        ]) ; parenthesis
("[].."      ["\\left[ \\middle\\vert \\right)"]]) ; parenthesis
("[]..c"     ["\\lceil \\rceil"          ]) ; parenthesis (ceil)
("[]..f"     ["\\lfloor \\rfloor"       ]) ; parenthesis (floor)
("{}.)"      ["\\left\\{ \\right\\}"]    ]) ; parenthesis
("{}.."      ["\\left\\{ \\middle\\vert \\right\\}")] ; parenthesis (set maker)
("<.>."      ["\\left< \\right>"          ]) ; parenthesis
("<.>.."      ["\\left< \\middle\\vert \\right>"]]) ; parenthesis
("\\vert\\vert ." ["\\left\\vert \\right\\vert"]) ; parenthesis (abs)
("\\vert\\vert .." ["\\left\\Vert \\right\\Vert"]) ; parenthesis (abs)
("(."        ["\\left("                ]) ; half-parenthesis
(").)"       ["\\right)"               ]) ; half-parenthesis
("[."        ["\\left["                  ]) ; half-parenthesis
("].)"       ["\\right]"                 ]) ; half-parenthesis
("{."        ["\\left\\{"                 ]) ; half-parenthesis
("}.)"       ["\\right\\}"              ]) ; half-parenthesis
("<."        ["\\left<"                     ]) ; half-parenthesis
(">."        ["\\right>"                ]) ; half-parenthesis
("\\vert ."   ["\\Bigg\\vert_{ }^{ }"]) ; definite integral range
;; Structural: Text
("te"       ["\\text{"                ]) ; normal text
("tr"       ["\\mathrm{"              ]) ; math roman (used for sin,cos,tan ...)
("tb"       ["\\mathbf{"             ]) ; math bold
("ti"       ["\\mathit{"             ]) ; math italics
;; Structural: Text
("te"       ["\\text{"                ]) ; normal text
("tr"       ["\\mathrm{"              ]) ; math roman (used for sin,cos,tan ...)
("tb"       ["\\mathbf{"             ]) ; math bold
("ti"       ["\\mathit{"             ]) ; math italics
;; Structural: Sub-sup-scripts
("^"        ["^{ "                      ]) ("_"      ["_{ "                      ])
("pp"       ["^{ "                      ]) ("11"     ["_{ "                      ])
("p0"       ["^0"                     ]) ("10"     ["_0"                     ])
("p1"       ["^1"                     ]) ("11"     ["_1"                     ])
("p2"       ["^2"                     ]) ("12"     ["_2"                     ])
("p3"       ["^3"                     ]) ("13"     ["_3"                     ])
("p4"       ["^4"                     ]) ("14"     ["_4"                     ])
("pn"       ["^n"                     ]) ("l1nn"   ["_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"             ]) ;
)

```