

Emacs TeQ: (T_EX + Quail)

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

Garid Zorigoo

April 10, 2023

Contents

1	Alphabet related stuff :	2
1.1	Greek	2
1.2	Matrix (aka bold)	4
1.3	Vector & Hat	5
2	Function Expansion	6
3	Symbols :	6
3.1	Dots related	6
3.2	Geometry	6
3.3	Letter like	7
3.4	Spaces	7
3.5	Arrows:	8
3.5.1	Single:	8
3.5.2	Double:	8
3.5.3	Long arrow with top-bottom entries	9
4	Symbol Modification	9
4.1	Accents (variable decoration?)	9
5	Binary Operation Symbols	10
5.1	Simple Arithmetics:	10
5.2	Binary Relations:	12
5.3	Set symbols	13
5.4	Logic	13

6	Functions	14
6.1	Function	14
6.2	Trigonometry: function	14
6.3	Iterative-like operation:	15
7	Structural:	16
7.1	Parenthesis Related	16
7.2	Texts:	16
7.3	Superscripts (power) & Subscripts (lower)	17
7.4	Misc.	17
7.5	xy Diagram related	17
8	Formatting Table into Elisp	18
9	Executable elisp function definition	27
10	Making the el	28

1 Alphabet related stuff :

1.1 Greek

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>\varepsilonpsilon</code>
f..	φ	<code>\varphiphi</code>
s..	ς	<code>\varsigmasigma</code>
t..	ϑ	<code>\varthetatheta</code>
r..	ϱ	<code>\varrhorho</code>
p..	ϖ	<code>\varppi</code>
k..	\varkappa	<code>\varkappakappa</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>

2 Function Expansion

Table 5: 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 6: 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 7:

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 8: 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 9: Space Symbol

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

3.5 Arrows:

3.5.1 Single:

Table 10: 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 11: 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 12: Long arrow Line arrows

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

4 Symbol Modification

4.1 Accents (variable decoration?)

Table 13:

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

5 Binary Operation Symbols

5.1 Simple Arithmetics:

Table 14: Simple Arithmetics operations

key	trans	sym
<code>+-</code>	<code>\pm</code>	\pm
<code>-+</code>	<code>\mp</code>	\mp
<code>*x</code>	<code>\times</code>	\times
<code>::</code>	<code>\div</code>	\div
<code>**</code>	<code>\cdot</code>	\cdot

5.2 Binary Relations:

Table 15:

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 16:

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 17:

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 18:

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

6.2 Trigonometry: function

Table 19:

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

6.3 Iterative-like operation:

Table 20: Integrals, Sums, Products

key	trans	sym	description
il	<code>\limits_{ }^{ }</code>	\sum_{here}^{here}	
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_{-\infty}^{-\infty}</code>	$\int_{-\infty}^{-\infty}$	
inti.	<code>\iint\limits_C</code>	\iint_C	
intii.	<code>\iiint\limits_C</code>	\iiint_C	
intiii.	<code>\iiiiint\limits_C</code>	\iiiiint_C	
into.	<code>\oint\limits_C</code>	\oint_C	

7 Structural:

7.1 Parenthesis Related

Table 21:

key	trans	sym	description
().	<code>\left(\right)</code>	(\square)	
()..	<code>\left(\middle\vert \right)</code>	$(\square \square)$	
[].	<code>\left[\right]</code>	$[\square]$	
[]..	<code>\left[\middle\vert \right]</code>	$[\square \square]$	
[]..c	<code>\lceil \rceil</code>	$\lceil \square \rceil$	
[]..f	<code>\lfloor \rfloor</code>	$\lfloor \square \rfloor$	
{ }.	<code>\left\{ \right\}</code>	$\{\square\}$	
{ }..	<code>\left\{ \middle\vert \right\}</code>	$\{\square \square\}$	
.	<code>\left\ \right\ </code>	$\ \square\ $	

7.2 Texts:

Table 22:

key	trans	sym	description
te	<code>\text{}</code>	$a + \text{text}$	
tr	<code>\mathrm{}</code>	$a + \mathrm{text}$	
tb	<code>\mathbf{}</code>	$a + \mathbf{text}$	
ti	<code>\mathit{}</code>	$a + \mathit{text}$	

7.3 Superscripts (power) & Subsripts (lower)

Table 23:

key	sym	trans	key	sym	trans
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	$\backslash \overset{\square}{\square}$
---	$\underbrace{\square}$	$\backslash \underbrace{\square}_{\square}$	^^.	$\overbrace{\square}$	$\backslash \overbrace{\square}^{\square}$
---	$\underline{\square}$	$\backslash \underline{\square}$	^^..	$\overline{\square}$	$\backslash \overline{\square}$

7.4 Misc.

Table 24:

key	trans	sym	description
binom	$\backslash \text{binom}$	$\binom{\square}{\square}$	
box	$\backslash \text{boxed}$	$\boxed{\square}$	
can	$\backslash \text{cancel}$	$\cancel{\square}$	requires <code>cancel</code>
&=	$\&=\backslash n\\\\$		
=&	$\&=\backslash n\\\\$		

7.5 xy Diagram related

Table 25:

key	trans	sym	description
xy	$\backslash \text{xymatrix}\{\backslash n\backslash n\}$		
bu	$\backslash \text{bullet}$	\bullet	
ar	$\backslash \text{ar}$		

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", tbl1_greek)
format_table_to_elisp_type6col("Matrix", tbl1_matrix)
format_table_to_elisp_type6col("Vector & Hat", tbl1_vec)

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

;; 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}"      ])

;; 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
("<->" ["\\leftrightharpoon" ]) ; left-right arrow
("<-n" ["\\nleftarrow" ]) ; not left arrow
("<->n" ["\\nrightarrow" ]) ; not right arrow
("<-^n" ["\\nuparrow" ]) ; not up arrow
("<-vn" ["\\ndownarrow" ]) ; not down arrow
("<->" ["\\nleftrightharpoon" ]) ; not left-right arrow
("<->" ["\\longrightarrow" ]) ;
("<->" ["\\longleftarrow" ]) ;
("\\vert ->" ["\\mapsto" ]) ;
;; Symbols arrow2
("<=" ["\\Leftarrow" ]) ; left arrow
("<=" ["\\Rightarrow" ]) ; right arrow
("<=" ["\\Uparrow" ]) ; up arrow
("<=" ["\\Downarrow" ]) ; down arrow
("<=" ["\\Leftrightarrow" ]) ; left-right arrow
("iff" ["\\Leftrightarrow" ]) ; left-right arrow

```

```

("<=n"      ["\\nLeftarrow"      ]) ; left arrow
("=>n"      ["\\nRrightarrow"     ]) ; 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..."  ["\\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"              ]) ;
("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_{ }~{ }"    ]) ;
("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_{ -\\infty }^{ -\\infty }"]) ;
("inti."    ["\\iint\\limits_{ C }"]) ;
("intii."   ["\\iiint\\limits_{ C }"]) ;
("intiii."  ["\\iiint\\limits_{ C }"]) ;
("into."    ["\\oint\\limits_{ C }"]) ;
;; Structural: Parenthesis
("().)"     ["\\left( \\right)"        ]) ;
("()..)"    ["\\left( \\middle\\vert \\right)"]) ;
("[])."     ["\\left[ \\right]"             ]) ;
("[]..)"    ["\\left[ \\middle\\vert \\right)"]) ;
("[] .c"    ["\\lceil \\rceil"             ]) ;
("[] .f"    ["\\lfloor \\rfloor"       ]) ;
("{ })."    ["\\left\\{ \\right\\}"]) ;
("{ }..)"   ["\\left\\{ \\middle\\vert \\right\\}"]) ;
("\\vert\\vert ." ["\\left\\vert \\right\\vert"]) ;
;; Structural: Text
("te"       ["\\text{ }"                ]) ;
("tr"       ["\\mathrm{ }"              ]) ;
("tb"       ["\\mathbf{ }"             ]) ;
("ti"       ["\\mathit{ }"            ]) ;
;; Structural: Text
("te"       ["\\text{ }"                ]) ;
("tr"       ["\\mathrm{ }"              ]) ;
("tb"       ["\\mathbf{ }"             ]) ;
("ti"       ["\\mathit{ }"            ]) ;
;; Structural: Sub-sup-scripts
("pp"       ["^{ "                  ]) ("11"      ["_{"                  ])
("p0"       ["^0"                  ]) ("10"      ["_0"                  ])
("p1"       ["^1"                  ]) ("11"      ["_1"                  ])
("p2"       ["^2"                  ]) ("12"      ["_2"                  ])
("p3"       ["^3"                  ]) ("13"      ["_3"                  ])

```

```

("p4"      ["^4"                ]) ("l4"      ["_4"                ])
("pn"      ["^n"                ]) ("lnn"     ["_n"                ])
("px"      ["^x"                ]) ("li"      ["_i"                ])
("__"      ["\\underset{ }{ }"]) ("^^"      ["\\overset{ }{ }"])
("___."    ["\\underbrace{ }_{ }"]) ("^^."     ["\\overbrace{ }^{ }"])
("___.."   ["\\underline{ }" ]) ("^^.."    ["\\overline{ }"  ])
;; Structural: misc
("binom"   ["\\binom"           ]) ;
("box"     ["\\boxed"           ]) ;
("can"     ["\\cancel"          ]) ; requires ~cancel~
("&="      ["&=\\n\\\\\\\\\\\\\\\\"]]) ;
("=&"     ["&=\\n\\\\\\\\\\\\\\\\"]]) ;
;; Structural: xy
("xy"      ["\\xymatrix{\\n\\n}" ]) ;
("bu"      ["\\bullet"          ]) ;
("ar"      ["\\ar"              ]) ;

```

9 Executable elisp function definition

```

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

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

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

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

```

```

(quail-func-end))

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

(defun quail-TeX-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
      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 "}{}"))))

```



```

("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}"      ])

;; 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

;; 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
("<->"      ["\\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
("<==>"     ["\\Longlefttrightarrow"  ]) ; left-right arrow
("<==>"     ["\\Longleftarrow"        ]) ; left-right arrow

```

```

(==">"      ["\\Longrightarrow"    ]) ; left-right arrow
;; Symbols arrow3
("<---"      ["\\xleftarrow[ ]{ }"   ]) ;
("---->"     ["\\xrightarrow[ ]{ }"  ]) ;
("====>"     ["\\xRightarrow[ ]{ }"  ]) ; ~mathtools~ lib required
("<===="     ["\\xLeftarrow[ ]{ }"   ]) ; ~mathtools~ lib required
;; Symbols arrow3
("vec"       ["\\vec"                ]) ;
("bar"       ["\\bar"                ]) ;
("hat"       ["\\hat"                ]) ;
("dot"       ["\\dot"                ]) ;
("dot."      ["\\ddot"               ]) ;
("dot.."     ["\\dddotted"           ]) ;
("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="    ["\\supseteq"] ) ;
("sup=n"   ["\\nsupseteq"] ) ;
;; Operation: arith

```

```

("or"      ["\\lor"          ]) ;
("and"     ["\\lnd"         ]) ;
("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_{ }^{ }" ]) ;
("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_{ -\\infty }^{ -\\infty }"]) ;
("inti."     ["\\iint\\limits_{ C }"]) ;
("intii."    ["\\iiint\\limits_{ C }"]) ;
("intiii."   ["\\iiint\\limits_{ C }"]) ;
("into."     ["\\oint\\limits_{ C }"]) ;
;; Structural: Parenthesis
("(".."      ["\\left( \\right)"        ]) ;
("(".."      ["\\left( \\middle\\vert \\right)"]) ;
("[]."       ["\\left[ \\right]"         ]) ;
("[]."       ["\\left[ \\middle\\vert \\right)"]) ;
("[].c"      ["\\lceil \\rceil"           ]) ;
("[].f"      ["\\lfloor \\floor"       ]) ;
("{ }."      ["\\left\\{ \\right\\}"]) ;
("{ }.."     ["\\left\\{ \\middle\\vert \\right\\}"]) ;
("\\vert\\vert ." ["\\left\\vert \\right\\vert"]) ;
;; Structural: Text
("te"       ["\\text{"                ]) ;
("tr"       ["\\mathrm{"              ]) ;
("tb"       ["\\mathbf{"              ]) ;
("ti"       ["\\mathit{"              ]) ;
;; Structural: Text
("te"       ["\\text{"                ]) ;
("tr"       ["\\mathrm{"              ]) ;
("tb"       ["\\mathbf{"              ]) ;
("ti"       ["\\mathit{"              ]) ;
;; 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"                   ]) ("lnn"     ["_n"                   ])
("px"       ["^x"                   ]) ("li"      ["_i"                   ])
("__"       ["\\underset{ }{ }"]) ("^^"      ["\\overset{ }{ }"])
("__."      ["\\underbrace{ }_{ }"]) ("^^."     ["\\overbrace{ }^{ }"])
("__.."     ["\\underline{ }" ]) ("^^.."    ["\\overline{ }" ])

```

```

;; Structural: misc
("binom" ["\\binom" ]) ;
("box" ["\\boxed" ]) ;
("can" ["\\cancel" ]) ; requires ~cancel~
("&=" ["&=\\n\\\\\\\\\\\\\\\\"] ) ;
("=&" ["&=\\n\\\\\\\\\\\\\\\\"] ) ;
;; Structural: xy
("xy" ["\\xymatrix{\\n\\n}" ]) ;
("bu" ["\\bullet" ]) ;
("ar" ["\\ar" ]) ;
)

```