

The okicmd and okithm Packages

Taihei Oki

April 27, 2020

1 The okicmd Package

1.1 Letters

Input	Output	L ^A T _E X equivalent
1	ℓ	<code>\ell</code>
<code>\ell</code>	l	<code>l</code>
<code>\epsilon</code>	ε	<code>\varepsilon</code>
<code>\varepsilon</code>	ϵ	<code>\epsilon</code>
<code>\phi</code>	φ	<code>\varphi</code>
<code>\varphi</code>	ϕ	<code>\phi</code>

1.2 Parentheses

Input	Output	L ^A T _E X (almost) equivalent
<code>\prn{\cdot}</code>	(\cdot)	<code>\left(\cdot\right)</code> ¹
<code>\prn*{\cdot}</code>	(\cdot)	<code>(\cdot)</code>
<code>\prn[\big]{\cdot}</code>	(\cdot)	<code>\bigl(\cdot\bigr)</code>
<code>\prn[Big]{\cdot}</code>	(\cdot)	<code>\Bigl(\cdot\Bigr)</code>
<code>\prn[bigg]{\cdot}</code>	(\cdot)	<code>\biggl(\cdot\biggr)</code>
<code>\prn[Bigg]{\cdot}</code>	(\cdot)	<code>\Biggl(\cdot\Biggr)</code>
<code>\curl{\cdot}</code>	$\{\cdot\}$	<code>\left\{\cdot\right\}</code>
<code>\sqbr{\cdot}</code>	$[\cdot]$	<code>\left[\cdot\right]</code>
<code>\agbr{\cdot}</code>	$\langle\cdot\rangle$	<code>\left\langle\cdot\right\rangle</code>
<code>\dbbr{\cdot}</code>	$\llbracket\cdot\rrbracket$	<code>\left\llbracket\cdot\right\rrbracket</code>
<code>\pipe{\cdot}</code>	$ \cdot $	<code>\left \cdot\right </code>
<code>\dbpp{\cdot}</code>	$\ \cdot \ $	<code>\left\ \cdot\right\ </code>
<code>\floor{\cdot}</code>	$\lfloor\cdot\rfloor$	<code>\left\lfloor\cdot\right\rfloor</code>
<code>\ceil{\cdot}</code>	$\lceil\cdot\rceil$	<code>\left\lceil\cdot\right\rceil</code>

¹The exactly equivalent L^AT_EX code is `\mathopen{\left(\vphantom{\cdot}\kern-\nulldelimiterspace\right.\cdot\mathclose{\left.\kern-\nulldelimiterspace\right)}`. This is the same for other parenthesis commands.

1.3 Logic

Input	Output	L ^A T _E X equivalent
<code>\bigland</code>	\bigwedge	<code>\bigwedge</code>
<code>\biglor</code>	\bigvee	<code>\bigvee</code>
<code>a \defeq b</code>	$a := b$	<code>a \coloneqq b</code>
<code>a \eqdef b</code>	$b =: a$	<code>a \eqqcolon b</code>
<code>P \defiff Q</code>	$P \stackrel{\text{def}}{\iff} Q$	<code>P \overset{\mathrm{def}}{\iff} Q</code>

1.4 Sets

Input	Output	L ^A T _E X (almost) equivalent
<code>\set{a \in S}</code>	$\{a \in S\}$	<code>\left\{a \in S\right\}</code>
<code>\set{a \in S}[a^2 = 1]</code>	$\{a \in S \mid a^2 = 1\}$	<code>\left\{a \in S\mathrel{\left \right.} a^2 = 1\right\}</code> ²
<code>\card{X}</code>	$ X $	<code>\left X\right </code>
<code>X \syndif Y</code>	$X \triangle Y$	<code>X \mathbin{\triangle} Y</code>
<code>\setN</code>	\mathbb{N}	<code>\mathbb{N}</code>
<code>\setZ</code>	\mathbb{Z}	<code>\mathbb{Z}</code>
<code>\setQ</code>	\mathbb{Q}	<code>\mathbb{Q}</code>
<code>\setR</code>	\mathbb{R}	<code>\mathbb{R}</code>
<code>\setC</code>	\mathbb{C}	<code>\mathbb{C}</code>
<code>\setH</code>	\mathbb{H}	<code>\mathbb{H}</code>
<code>\setF</code>	\mathbb{F}	<code>\mathbb{F}</code>
<code>\setK</code>	\mathbb{K}	<code>\mathbb{K}</code>
<code>\setZp</code>	$\mathbb{Z}_{\geq 0}$	<code>\mathbb{Z}_{\geq 0}</code>
<code>\setQp</code>	$\mathbb{Q}_{\geq 0}$	<code>\mathbb{Q}_{\geq 0}</code>
<code>\setRp</code>	$\mathbb{R}_{\geq 0}$	<code>\mathbb{R}_{\geq 0}</code>
<code>\setNpp</code>	$\mathbb{N}_{> 0}$	<code>\mathbb{N}_{> 0}</code>
<code>\setZpp</code>	$\mathbb{Z}_{> 0}$	<code>\mathbb{Z}_{> 0}</code>
<code>\setQpp</code>	$\mathbb{Q}_{> 0}$	<code>\mathbb{Q}_{> 0}</code>
<code>\setRpp</code>	$\mathbb{R}_{> 0}$	<code>\mathbb{R}_{> 0}</code>

1.5 Maps

Input	Output	L ^A T _E X (almost) equivalent
<code>\doms{X}{Y}</code>	$X \rightarrow Y$	<code>\{X\}\to\{Y\}</code>
<code>\funcdoms{f}{X}{Y}</code>	$f : X \rightarrow Y$	<code>\{f\}\vcentcolon\{X\}\to\{Y\}</code>
<code>\restr{f}{S}</code>	$f _S$	<code>\left.f\right _{\{S\}}</code>
<code>\id_K</code>	id_K	<code>\operatorname{id}_K</code>
<code>\dom f</code>	$\text{dom } f$	<code>\operatorname{dom} f</code>
<code>\cod f</code>	$\text{cod } f$	<code>\operatorname{cod} f</code>
<code>\supp f</code>	$\text{supp } f$	<code>\operatorname{supp} f</code>

²The exactly equivalent L^AT_EX code is `\mathopen{\left\{\vphantom{a \in S}a \in S\right\}\kern-\nulldelimiterspace\right.}\{a \in S\mathrel{\left|\right.}\left.\kern-\nulldelimiterspace\right.}\{a^2 = 1\}\middle|\kern-\nulldelimiterspace\right.}\{a^2 = 1\}\mathclose{\left.\kern-\nulldelimiterspace\right.}\{a \in S\right\}`. This is the same for other parenthesis commands with middle bars.

1.6 Lattices

Input	Output	L ^A T _E X equivalent
<code>x \meet y</code>	$x \wedge y$	<code>x \mathbin{\wedge} y</code>
<code>x \join y</code>	$x \vee y$	<code>x \mathbin{\vee} y</code>
<code>\bigmeet</code>	\bigwedge	<code>\bigwedge</code>
<code>\bigjoin</code>	\bigvee	<code>\bigvee</code>

1.7 Algebra

Input	Output	L ^A T _E X (almost) equivalent
<code>\Hom(G)</code>	$\text{Hom}(G)$	<code>\operatorname{Hom}(G)</code>
<code>\End R</code>	$\text{End } R$	<code>\operatorname{End} R</code>
<code>\Aut_k K</code>	$\text{Aut}_k K$	<code>\operatorname{Aut}_k K</code>
<code>\gen{a, b}</code>	$\langle a, b \rangle$	<code>\left\langle a, b \right\rangle</code>
<code>\gen{a, b}[ab = e]</code>	$\langle a, b \mid ab = e \rangle$	<code>\left\langle a, b \mathrel{\middle } ab = e \right\rangle</code>
<code>\abel{G}</code>	G_{ab}	<code>G_{\mathrel{ab}}</code>
<code>\comm{G}</code>	$[G, G]$	<code>\left[G, G\right]</code>
<code>\ord G</code>	$\text{ord } G$	<code>\operatorname{ord} G</code>
<code>\sym_n</code>	\mathfrak{S}_n	<code>\mathfrak{S}_n</code>
<code>\sgn(\sigma)</code>	$\text{sgn}(\sigma)$	<code>\operatorname{sgn}(\sigma)</code>
<code>\mult{R}</code>	R^\times	<code>R^{\times}</code>
<code>\M_{m,n}(R)</code>	$M_{m,n}(R)$	<code>\operatorname{M}_{m,n}(R)</code>
<code>\GL_n(R)</code>	$\text{GL}_n(R)$	<code>\operatorname{GL}_n(R)</code>
<code>\SL_n(R)</code>	$\text{SL}_n(R)$	<code>\operatorname{SL}_n(R)</code>
<code>\O(n)</code>	$O(n)$	<code>\operatorname{O}(n)</code>
<code>\SO(n)</code>	$\text{SO}(n)$	<code>\operatorname{SO}(n)</code>
<code>\U(n)</code>	$U(n)$	<code>\operatorname{U}(n)</code>
<code>\SU(n)</code>	$\text{SU}(n)$	<code>\operatorname{SU}(n)</code>
<code>\GL(q)</code>	$\text{GL}(q)$	<code>\operatorname{GL}(q)</code>
<code>L \extends K</code>	L / K	<code>L \mathbin{/} K</code>

1.8 Number Theory

Input	Output	L ^A T _E X (almost) equivalent
<code>\abs{x}</code>	$ x $	<code>\left x\right </code>
<code>\intset{n}</code>	$[n]$	<code>\left[n\right]</code>
<code>a \coprime b</code>	$a \perp b$	<code>a \mathrel{\bot} b</code>
<code>a \divides b</code>	$a \mid b$	<code>a \mid b</code>
<code>a \ndivides b</code>	$a \nmid b$	<code>a \nmid b</code>

1.9 Linear Algebra

Input	Output
<code>\tr A</code>	$\text{tr } A$
<code>\rank A</code>	$\text{rank } A$
<code>\trank A</code>	$\text{t-rank } A$
<code>\Pf A</code>	$\text{Pf } A$
<code>\diag(a_1, \ldots, a_n)</code>	$\text{diag}(a_1, \dots, a_n)$
<code>\blockdiag(A_1, \ldots, A_n)</code>	$\text{block-diag}(A_1, \dots, A_n)$
<code>\vectorize(A)</code>	$\text{vec}(A)$
<code>\Row(A)</code>	$\text{Row}(A)$
<code>\Col(A)</code>	$\text{Col}(A)$
<code>\onevec</code>	$\mathbb{1}$
<code>\trsp{A}</code>	A^\top
<code>\adjo{A}</code>	A^*
<code>\inpr{x}{y}</code>	$\langle x, y \rangle$

1.10 Analysis

Input	Output
<code>\intoo{a, b}</code>	(a, b)
<code>\intoc{a, b}</code>	$(a, b]$
<code>\intco{a, b}</code>	$[a, b)$
<code>\intcc{a, b}</code>	$[a, b]$
<code>\e</code>	e
<code>\d</code>	d
<code>\dif{f}{x}</code>	$\frac{df}{dx}$
<code>\pdif{f}{x}</code>	$\frac{\partial f}{\partial x}$
<code>\ddif{f}{x}</code>	$\frac{d^2 f}{dx^2}$
<code>\dpdif{f}{x}</code>	$\frac{\partial^2 f}{\partial x^2}$

1.11 Complex Analysis

Input	Output
<code>\i</code>	i
<code>\Re z</code>	$\text{Re } z$
<code>\Im z</code>	$\text{Im } z$
<code>\Arg z</code>	$\text{Arg } z$
<code>\Loc z</code>	$\text{Log } z$
<code>\Sin z</code>	$\text{Sin } z$
<code>\Cos z</code>	$\text{Cos } z$
<code>\Tan z</code>	$\text{Tan } z$
<code>\Res_{z=0} f(z)</code>	$\text{Res}_{z=0} f(z)$

1.12 Optimization

Input	Output
<code>\argmin_{x \in S} f(x)</code>	$\arg \min_{x \in S} f(x)$
<code>\argmax_{x \in S} f(x)</code>	$\arg \max_{x \in S} f(x)$
<code>\Order(n)</code>	$O(n)$
<code>\order(n)</code>	$o(n)$

2 The okithm Package

2.1 Theorems

If the language is set to Japanese like by `\usepackage[main = japanese]{babel}`, okithm will translate all the environment titles (Theorem, Definition, etc.) into Japanese. You can disable theorems by giving the option `notheorem` to okicmd.

```

1 \begin{theorem}[Awesome theorem]
2   The square root  $\sqrt{2}$  of two is irrational.
3 \end{theorem}
4
5 \begin{definition}[Coprime]
6   Integers  $a$  and  $b$  are said to be \emph{coprime} if their greatest common
   divisor is one.
7 \end{definition}
8
9 \begin{lemma}
10  If  $a$  and  $b$  are coprime, so are  $a^2$  and  $b^2$ .
11 \end{lemma}
12
13 \begin{proposition}
14  If  $\sqrt{2} = a/b$ , then  $a^2 = 2b^2$ .
15 \end{proposition}
16
17 \begin{corollary}
18  If  $\sqrt{2} = a/b$  with  $a$  and  $b$  being coprime, then  $a$  is even.
19 \end{corollary}
20
21 \begin{example}
22  If  $a = 2$  and  $b = 1$ , then  $a$  is even but  $\sqrt{2} \neq a/b$ .
23 \end{example}
24
25 \begin{remark}
26  Note that  $a$  and  $b$  must be integers.
27 \end{remark}
28
29 \begin{proof}[of Awesome theorem]
30  Suppose to the contrary that  $\sqrt{2} = a/b$  with coprime  $a$  and  $b$ .
31  Then both  $a$  and  $b$  are even, which contradicts the assumption.
32 \end{proof}

```

Theorem 2.1 (Awesome theorem). *The square root $\sqrt{2}$ of two is irrational.*

Definition 2.2 (Coprime). Integers a and b are said to be *coprime* if their greatest common divisor is one.

Lemma 2.3. If a and b are coprime, so are a^2 and b^2 .

Proposition 2.4. If $\sqrt{2} = a/b$, then $a^2 = 2b^2$.

Corollary 2.5. If $\sqrt{2} = a/b$ with a and b being coprime, then a is even.

Example 2.6. If $a = 2$ and $b = 1$, then a is even but $\sqrt{2} \neq a/b$. □

Remark 2.7. Note that a and b must be integers.

Proof (of Awesome theorem). Suppose to the contrary that $\sqrt{2} = a/b$ with coprime a and b . Then both a and b are even, which contradicts the assumption. □

2.2 Algorithms

You can disable algorithms by setting the option `noalgorithm`.

```

1 \begin{algorithmic}[1
2   \Input{$n$ \in $\mathbb{N}$}
3   \Output{$n(n+1)/2$}
4   \State{$s$ \gets 0}
5   \ForTo{$i = 1$}{$n$}
6     \State{$s$ \gets $s + $i$}
7   \EndFor
8   \State{\Return $s$}
9 \end{algorithmic}

```

Input : $n \in \mathbb{N}$

Output: $n(n+1)/2$

1: $s \leftarrow 0$

2: **for** $i = 1$ **to** n **do**

3: $s \leftarrow s + i$

4: **return** s

2.3 Optimization Problems

You can change `minimize`, `maximize` and `subject to` into `min`, `max` and `s.t.`, respectively, by setting the option `optstyle = short`.