

# Kp-fonts: OpenType version

Daniel FLIPO

11th December 2021

This bundle provides OpenType versions of Type1 Kp-fonts designed by Christophe Caignaert. See `Kpfonts-Doc-English.pdf` for the full documentation of the original Type1 fonts.

It is usable only with LuaTeX or XeTeX engines; it consists of sixteen Text OpenType fonts, a Roman family **KpRoman** in eight shapes and weights—*Regular, Italic, Bold, BoldItalic, Light, LightItalic, Semibold, SemiboldItalic*—, a Sans-Serif family **KpSans** and a TypeWriter family **KpMono**, each of them in four shapes and weights—*Regular, Italic, Bold* and *BoldItalic*— and five Math OpenType fonts **KpMath-Regular**, **KpMath-Bold**, **KpMath-Light**, **KpMath-Semibold** and **KpMath-Sans**.

**KpRoman** and **KpSans** families have small caps available in two sizes (SmallCaps and PetiteCaps), upper and lowercase digits (0123456789), ancient ligatures  $\text{ct}$ ,  $\text{st}$  and  $\text{Q}$  a long-tailed capital Q. Superior and inferior digits and letters have been added to the OpenType **KpRoman** and **KpSans** fonts for footnotes' calls and abbreviations 1<sup>st</sup>, 2<sup>nd</sup>...

The support of text fonts covers Latin and Latin Extended A (U+0020 to U+017F) but neither IPA nor Greek nor Cyrillic alphabets.

For all Math fonts, Latin and Greek letters are available in Upright and Italic shapes, in Bold and Regular weights:  $\alpha \beta \text{C} \Delta$ ,  $\alpha \beta \text{C} \Delta$ ,  $\alpha \beta \text{C} \Delta$ ,  $\alpha \beta \text{C} \Delta$ .

Blackboard Bold capitals are available in two shapes, Serif and Sans: `\mathbb{ABC}` prints either  $\mathbb{ABC}$  (default) or  $\mathbb{ABC}$  (option `[Style=bbsans]`) Commands `\mathcal{ABC}` and `\mathscr{ABC}` print respectively  $\mathcal{ABC}$  and  $\mathscr{ABC}$  while `\mathfrak{ABCabc}` prints  $\mathfrak{ABCabc}$ .

File `unimath-kpfonts.pdf` shows the full list of Unicode Math symbols provided by Kp-fonts, compared with other common Math fonts. More symbols, specific to Kp-fonts, are listed in section 3.2.

A style file `kpfont-otf.sty` is provided to load Kp-fonts easily. It is derived from `kpfont.sty` but options differ.

Please beware of the *experimental* status of the current version (0.41).

All fonts are covered by OFL licence, style file and documentation are under LPPL-1.3 licence.

# 1 Loading kpfonts-otf.sty

For users of the original kpfonts.sty package, the easiest way to try the OpenType version is to load kpfonts-otf.sty:

```
\usepackage[ options ]{kpfonts-otf}
```

this loads unicode-math (and fontspec) and defines KpRoman (Regular or Light depending on options), KpSans and KpMono as Text fonts, KpMath (Regular or Light depending on options) as Math fonts.

kpfonts-otf.sty also defines all symbols available in latexsym and amssymb under the same names<sup>1</sup> and a bunch of Kp-fonts specifics symbols.

## 1.1 Global options for both Text and Maths

**light:** switches to *light* fonts, metrics are unchanged; *light* fonts might not look perfect on screen but they print fine.

Please compare *regular* (left) and *light* fonts (right):

Normal or light? Just a matter of taste. <i>E = mc<sup>2</sup></i>	Normal or light? Just a matter of taste. <i>E = mc<sup>2</sup></i>
---	---

<i>Normal or light? Just a matter of taste</i>	<i>Normal or light? Just a matter of taste</i>
--	--

<b>Normal or light? Just a matter of taste</b>	<b>Normal or light? Just a matter of taste</b>
--	--

<i>Normal or light? Just a matter of taste</i>	<i>Normal or light? Just a matter of taste</i>
--	--

**nomath:** load neither unicode-math nor KpMath fonts; useful for documents without Maths, or to choose other Math fonts.

**notext:** do not change any Text font, use the defaults.

**nosf:** do not change Sans-Serif Text fonts, use the defaults.

**nott:** do not change Typewriter Text fonts, use the defaults.

**onlyrm:** equivalent to the last two combined.

**fulloldstyle:** equivalent to options oldstyle and oldstylemath.

**fulloldstylenums:** equivalent to options oldstylenums and oldstylenumsmath.

## 1.2 Options for Text fonts *only*

**lighttext:** switches to *light* Text fonts.

Two more weights are provided by kpfonts-otf.sty: with the *lighttext* (or *light*) option, *Semibold* and *Extrabold* vs *Light* and *Semibold* without it. These weights are available through \ltseries, \sbseries and \ebseries commands to be used in a group or alternatively through one argument commands \textlt{ }, \textsb{ } and \texteb{ }.

{\sbseries\itshape Foo} or \textsb{\textit{Foo}} print *Foo*.

---

1. Unicode names often differ from AMS names.

**oldstylenums:** provides lowercase digits as a default.

To get uppercase digits locally: `{\addfontfeature{Numbers=Lining} 123}`.

Examples, upright, italic, bold and bolditalic:

- 0123456789!
- *0123456789!*
- **0123456789!**
- ***0123456789!***

**oldstyle:** provides lowercase digits as a default, long-tailed Q (Quebec) and (for Roman and Sans-Serif fonts only) old style ligatures « ct » et « st ».

Examples:

- *Quest* for an attractive font!
- *Quest* for an attractive font!
- **Quest** for an attractive font!
- **QUEST** FOR AN ATTRACTIVE FONT!
- *Quest* for an attractive font!
- **QUEST** FOR AN ATTRACTIVE FONT!

**veryoldstyle:** same as option `oldstyle` but the round ‘s’ is replaced by the long one ‘ſ’, unless it ends a word (then it remains ‘s’)<sup>2</sup> and ancient ligatures fi, fl, ft are activated. Coding `s=` prints a round ‘s’ anytime; in most cases this coding is not necessary with LuaTeX, f.i. `\textit{some of Highlands’ mysterious castles...}` will print *some of Highlands’ mysterious castles...* which is correct; with XeTeX an = sign must be added at the end of `mysterious=` to get a round ‘s’ there : final ‘s’ followed by a punctuation sign is never turned into ‘ſ’, when followed by a space it is a trickier, see option `longs` below.

**longs** inhibits the transformation of an ‘s’ followed by a space into ‘ſ’. This option, available only with LuaTeX, is silently switched on by `veryoldstyle`; it is used explicitly in the present document to deal with the examples of long ‘s’ shown in the preceding item.

**largesmallcaps:** prints larger SMALL CAPS than the default ones (PETITE CAPS).

**altfligs:** prints alternative shapes for ligatures fi, fl, ffi, ffl instead of fi, fl, ffi, ffl.

**germandbls:** \SS prints SS instead of ß (capital *Eszett*), ditto for small/petite caps.

**eurosym:** replaces the Euro character (€) by the official symbol (€) (available in regular, italic, bold and et bolditalic).

**harfbuzz:** switches `Renderer=Harfbuzz` for HBLuaTeX engine; up to version 0.34, this renderer was silently activated but seldom useful.

### 1.3 Options for Math fonts *only*

**lightmath:** uses *light* Math fonts.

---

2. The OpenType `calt` feature is used to catch ‘s’ ending a word.

**bbsans:** command `\mathbb` prints Sans-Serif Blackboard Bold capitals with Serif fonts too: compare  $\mathbb{C}, \mathbb{K}, \mathbb{N}, \mathbb{Q}, \mathbb{R}, \mathbb{Z}$ , with  $\mathbf{C}, \mathbf{K}, \mathbf{N}, \mathbf{Q}, \mathbf{R}, \mathbf{Z}$  (default).

**frenchstyle:** Latin uppercase letters and all Greek letters are printed upright, only lowercase latin letters are printed in italics; this follows the French typographic usage.

**oldstylenumsmath:** prints lowercase digits in Maths (default is uppercase).

**narrowiints:** prints condensed repeated integrals :

$\int\int\int$  et  $\iiint$  instead of  $\iiint$  et  $\iiint$  (default).

**partialup:** the `\partial` symbol is printed upright  $\partial$  instead of  $\partial$ .

**fancyReIm:** commands `\Re` et `\Im` print  $\Re$  and  $\Im$  respectively instead of  $\mathbb{R}$  et  $\mathbb{I}$ .

**tight:** horizontal spaces tightened in math mode (same settings as `fourier-otf`).

**noDcommand:** do not define `\D` to avoid incompatibilities with other packages.

Please note that the `mathcal` option has been deleted: commands `\mathcal{ABC}` and `\mathscr{ABC}` now print  $ABC$  and  $\mathcal{ABC}$  respectively when `kpfonts-otf.sty` is loaded.

## 2 Another way to load Kp-fonts

Loading Kp-fonts through `kpfonts-otf.sty` offers only a limited choice of options; the standard commands `\setmainfont`, `\setsansfont`, `\setmonofont`, `\setmathfont` offer much more flexibility.

On the other hand, `kpfonts-otf.sty` defines a lot of useful commands to access AMS and specific Kp-fonts symbols. Loading `kpfonts-otf` with the `symbols` option enables to get all these commands defined without loading any font:

```
\usepackage[symbols]{kpfonts-otf}
```

Please note that `unicode-math`<sup>3</sup> (and `fontspec`) *are loaded* by this procedure, no need to do it again, unless specific options are required, then `unicode-math` has to be loaded *before* `kpfonts-otf`, f.i.:

```
\usepackage[math-style=ISO,bold-style=upright]{unicode-math}
\usepackage[symbols]{kpfonts-otf}
```

Then, it is up to the user to load Kp-fonts with whatever option he/she likes using commands

```
\set...font{font}[options].
```

For documents requiring no Math fonts, loading `fontspec` and using the `\set...font` commands is enough, no need to load `kpfonts-otf` at all.

---

3. A careful reading of both manuals `unicode-math.pdf` and `fontspec.pdf` (available in all TeX distributions) is required in order to take full advantage of these packages.

## 2.1 Options for Text fonts

Here are the options available for Text Kp-fonts:

**Numbers=Lowercase** to get lowercase digits 1,2,3 instead of 1,2,3; the default is **Numbers=Lining**.

**SmallCapsFeatures = {Letters=SmallCaps}** the `\textsc{}` command will print larger SMALL CAPS than the default PETITE CAPS.

The default setting<sup>4</sup>, is **SmallCapsFeatures = {Letters=PetiteCaps}**.

**Ligatures=TeX** (default) ' !` ?` -- --- print respectively ' i ÿ - —.

**Ligatures=Common** (default) automatic ligatures ff ffi ffl fi fl.

**StylisticSet=1** provides an alternative for glyphs ffi ffl fi fl (ff is unchanged).

**Ligatures=Required** : adds ft et tt ligatures.

**Ligatures=Rare** adds çt et st ligatures.

**Style=Swash** to get the long-tailed capital Q: Queen, also in small caps (both sizes): QUEEN and QUEEN.

**Style=Historic** replaces any instance of 's' by the long variant f. It is still possible to get a round 's' coding it as 's=' specially at end of words; see options `veryoldstyle` and `longs` p. 3 for more details.

**Ligatures=Historic** switches specific ligatures for the long f: fi, fl, ft.

**StylisticSet=2** : \SS prints SS instead of ß (capital *Eszett*), ditto for small/petite caps.

**StylisticSet=3** : replaces the Euro character (€) by the official symbol (€) (available in regular, italics, bold and et bolditalic).

Options may be are chosen for each font, say:

```
\setmonofont{KpMono}[Numbers=Lowercase,Style=Historic]
```

but can also be shared by different typefaces:

```
\defaultfontfeatures+[KpRoman,KpSans,KpMono]{Numbers=Lowercase}
\defaultfontfeatures+[KpRoman,KpSans]{%
  Ligatures = Rare,
  Style = Swash,
  SmallCapsFeatures = {Letters=PetiteCaps},
}
\setmainfont{KpRoman}
\setsansfont{KpSans}
\setmonofont{KpMono}
```

Notes : 1. `\setmonofont{KpMono}`, `\setsansfont{KpSans}`, `\setmainfont{KpRoman}` rely on files `KpMono.fontspec`, `KpSans.fontspec` and `KpRoman.fontspec` installed by `Kpfonts`.

---

4. Changed in v0.37 to match the original `kpfonts` package.

2. Note the + ending `\defaultfontfeatures+` : options are *added*, not overwriting any other (including those of `fontspec.cfg`).

3. Options can be gathered: `Ligatures={Rare, Historic}` (with braces) is equivalent to `Ligatures=Rare` and `Ligatures=Historic`.

4. These options can also be switched on and off *locally* using `\addfontfeatures` inside a group, f.i. to print lowercase digits 1234576890 with a font loaded with option `Numbers=Lining`:

```
{\addfontfeatures{Numbers=Lowercase}1234576890}
```

Actually, a shortcut is available in this case: `\oldstylenums{1234576890}`.

5. With the `KpRoman`, it is possible to define two more weights *Light* and *Semibold* borrowed from `KpLight`:

```
\newfontfamily\KpLight{KpLight}[<same options as KpRoman>]
\newcommand*\ltseries{\KpLight}
\newcommand*\sbseries{\KpLight\bfseries}
\DeclareTextFontCommand{\textlt}{\ltseries}
\DeclareTextFontCommand{\textsb}{\sbseries}
```

These weights are then available through `\ltseries`, `\sbseries` commands to be used in a group or alternatively through one argument commands `\textlt{}` and `\textsb{}`.

With the `KpLight`, weights *Semibold* and *Extrabold* can be defined similarly.

## 2.2 Options for Math fonts

The following options can be passed either to `unicode-math`<sup>5</sup> or to `\setmathfont{}`:

`math-style = ISO, TeX (défaut), french, upright;`

`bold-style = ISO, TeX (défaut), upright;`

`partial = upright (default italic);`

`nabla = italic (default upright);`

Seven ‘Style Variants’ are available with Kp-fonts, here are the first three:

**Style=mathcal** (+ss01) commands `\mathcal{}` and `\mathscr{}` print *ABC* instead of *A B C* (default), see note below;

**Style=bbsans** (+ss02) `\mathbb{}` prints Sans-Serif Blackboard bold capitals *A B C* for Serif Math fonts `KpMath-Regular` and `KpMath-Light` instead of *A B C* ;

**Style=narrowiints** (+ss03) provides condensed repeated integrals:  $\iiint$  instead of  $\int\int\int$  (default).

**Note:** if you want commands `$$\mathcal{ABC}$$` and `$$\mathscr{ABC}$$` to print *ABC* and *A B C* respectively, you can use `unicode-math`’s option `range` this way:

```
\setmathfont{KpMath-Regular}[options]
\setmathfont{KpMath-Regular}[range={cal,bfcal},StylisticSet=1]
```

---

5. See the manual `unicode-math.pdf`.

Both lines are mandatory: the first one loads **KpMath** as usual while the second one modifies `\mathcal{}` command's output.

The next four tables present the other Style Variants available:

Table 1 – Style=leqslant (+ss04)

Command	Default	Variant
<code>\leq</code>	$\leq$	$\leqslant$
<code>\geq</code>	$\geq$	$\geqslant$
<code>\nleq</code>	$\not\leq$	$\not\leqslant$
<code>\ngeq</code>	$\not\geq$	$\not\geqslant$
<code>\leqq</code>	$\leq\leq$	$\leq\leqslant$
<code>\geqq</code>	$\geq\geq$	$\geq\geqslant$
<code>\leqless</code>	$\leq\lessdot$	$\leq\lessgtr$
<code>\eqgtr</code>	$\leq\gtrdot$	$\leq\gtrless$
<code>\lesseqgtr</code>	$\leq\gtrdot\leq$	$\leq\gtrless\leq$
<code>\gtreqless</code>	$\geq\lessdot\geq$	$\geq\lessgtr\geq$
<code>\lesseqqgtr</code>	$\leq\gtrdot\leq\leq$	$\leq\gtrless\leq\leq$
<code>\gtreqqless</code>	$\geq\lessdot\geq\geq$	$\geq\lessgtr\geq\geq$

Table 2 – Style=smaller (+ss05)

Command	Default	Variant
<code>\mid</code>	$ $	$\! $
<code>\nmid</code>	$\! $	$\! $
<code>\parallel</code>	$\parallel$	$\parallel$
<code>\nparallel</code>	$\nparallel$	$\nparallel$
<code>\parallelslant</code>	$\parallel$	$\parallel$
<code>\nparallelslant</code>	$\nparallel$	$\nparallel$

Table 3 – Style=subsetneq (+ss06)

Command	Default	Variant
<code>\subsetneq</code>	$\subsetneq$	$\subsetneq$
<code>\supsetneq</code>	$\supsetneq$	$\supsetneq$
<code>\subsetneqq</code>	$\subsetneqq$	$\subsetneqq$
<code>\supsetneqq</code>	$\supsetneqq$	$\supsetneqq$

Table 4 – Style=parallelslant (+ss07)

Command	Default	Variant
<code>\parallel</code>	$\parallel$	$\parallel$
<code>\nparallel</code>	$\nparallel$	$\nparallel$
<code>\shortparallel</code>	$\parallel$	$\parallel$
<code>\nshortparallel</code>	$\nparallel$	$\nparallel$

Example: switching styles 4 (leqslant) and 6 (subsetneq) can be achieved coding either `\setmathfont{KpMath-Regular.otf}[StylisticSet={4,6}]` or `\setmathfont{KpMath-Regular.otf}[Style={leqslant,subsetneq}]` but this second syntax is available only if `kpfonts-otf.sty` has been loaded (eventually with the `symbols` option).

Table 5 on the following page shows the available ‘Glyphs Variants’:

Example: with `\setmathfont{KpMath-Regular.otf}[CharacterVariant={3,6}]` commands `\epsilon` and `\phi` print  $\epsilon$  and  $\phi$  instead of  $\epsilon$  et  $\phi$ . The same is true of course for all shapes and and weights (upright, bold, bolditalic, sans-derif, etc.): f.i. with `math-syle=french`, `\epsilon` and `\phi` print  $\epsilon$  and  $\phi$  (upright).

Note about `\hbar`: `unicode-math` defines `\hbar` as `\hslash` (U+210F) while `amsmath` provides two different glyphs (italic h with horizontal or diagonal stroke). `kpfonts-otf` now follows `unicode-math`; the italic h with horizontal stroke can be printed using `\hslash` or `\hbar` together with character variant `cv01` or with `\mathbar` (replacement for AMS' command `\hbar`).

Table 5 – Glyphs Variants

	Default	Variant	Command
cv00	$\Re$ $\Im$	$\Re$ $\Im$	<code>\Re</code> <code>\Im</code>
cv01	$\hbar$	$\hbar$	<code>\hslash</code> or <code>\hbar</code>
cv02	$\emptyset$	$\emptyset$	<code>\emptyset</code>
cv03	$\epsilon$	$\epsilon$	<code>\epsilon</code>
cv04	$\kappa$	$\kappa$	<code>\kappa</code>
cv05	$\pi$	$\pi$	<code>\pi</code>
cv06	$\phi$	$\phi$	<code>\phi</code>
cv07	$\rho$	$\rho$	<code>\rho</code>
cv08	$\sigma$	$\sigma$	<code>\sigma</code>
cv09	$\theta$	$\theta$	<code>\theta</code>
cv10	$\Theta$	$\Theta$	<code>\Theta</code>

### 3 Specific commands defined in `kpfonts-otf.sty`

#### 3.1 Integrals

Kp-fonts offers variants for integral symbols suitable for indefinite integrals, they are coded as `\varint`, `\variint`, `\variiint`, `\variiiiint` and `\varidotsint`. Compare  $\int f(t) dt$  and  $\int f(t) dt$  and also

$$\int f(t) dt \quad \text{and} \quad \int f(t) dt$$

`\D{...}` prints an upright ‘d’ and improves kernings around the differential element: `\displaystyle\varint f(t)\D{t}` prints  $\int f(t) dt$ .

#### 3.2 Specific Math symbols

The next tables present symbols unavailable as Unicode characters, they are coded in Kp-fonts’ private zone.

<code>\mmapsto</code>	$\mapsto$	<code>\longmmapsto</code>	$\mapsto$
<code>\mmapsfrom</code>	$\mapsfrom$	<code>\longmmapsfrom</code>	$\mapsfrom$
<code>\Mmapsto</code>	$\Mmapsto$	<code>\Longmmapsto</code>	$\Mmapsto$
<code>\Mmapsfrom</code>	$\Mmapsfrom$	<code>\Longmmapsfrom</code>	$\Mmapsfrom$
<code>\leftrightdasharrow</code>	$\leftrightarrow$	<code>\leadsto</code>	$\leadsto$
<code>\boxright</code>	$\boxrightarrow$	<code>\boxleft</code>	$\boxleftarrow$
<code>\circleright</code>	$\circrightarrow$	<code>\circleleft</code>	$\circleftarrow$
<code>\Diamondright</code>	$\diamondrightarrow$	<code>\Diamondleft</code>	$\diamondleftarrow$
<code>\boxdotright</code>	$\boxdotrightarrow$	<code>\boxdotleft</code>	$\boxdotleftarrow$
<code>\circledotright</code>	$\circledrightarrow$	<code>\circledotleft</code>	$\circledleftarrow$
<code>\Diamonddotright</code>	$\diamondrightarrow$	<code>\Diamonddotleft</code>	$\diamondleftarrow$



$\backslash\boxRight$	$\boxrightarrow$	$\backslash\boxLeft$	$\boxleftarrow$
$\backslash\boxdotRight$	$\boxdotrightarrow$	$\backslash\boxdotLeft$	$\boxdotleftarrow$
$\backslash\DiamondRight$	$\Diamondrightarrow$	$\backslash\DiamondLeft$	$\Diamondleftarrow$
$\backslash\DiamonddotRight$	$\Diamonddotrightarrow$	$\backslash\DiamonddotLeft$	$\Diamonddotleftarrow$
$\backslash\multimapdot$	$\multimapdot$	$\backslash\multimapdotinv$	$\multimapdotinv$
$\backslash\multimapdotboth$	$\multimapdotboth$	$\backslash\multimapdotbothvert$	$\multimapdotbothvert$
$\backslash\multimapbothvert$	$\multimapbothvert$	$\backslash\multimapdotbothAvert$	$\multimapdotbothAvert$
$\backslash\multimapdotbothAvert$	$\multimapdotbothAvert$	$\backslash\multimapdotbothBvert$	$\multimapdotbothBvert$
$\backslash\capplus$	$\capplus$	$\backslash\sqcupplus$	$\sqcupplus$
$\backslash\parallelslant$	$\parallelslant$	$\backslash\colonsim$	$\colonsim$
$\backslash\parallelbackslant$	$\parallelbackslant$	$\backslash\Colonsim$	$\Colonsim$
$\backslash\eqqColon$	$\eqqColon$	$\backslash\Colondash$	$\Colondash$
$\backslash\sqcapplus$	$\sqcapplus$	$\backslash\colonapprox$	$\colonapprox$
$\backslash\Colonapprox$	$\Colonapprox$	$\backslash\colonapprox$	$\colonapprox$
$\backslash\Colonapprox$	$\Colonapprox$	$\backslash\colonapprox$	$\colonapprox$
$\backslash\Colonapprox$	$\Colonapprox$	$\backslash\colonapprox$	$\colonapprox$
$\backslash\strictif$	$\strictif$	$\backslash\strictfi$	$\strictfi$
$\backslash\circledvee$	$\circledvee$	$\backslash\circledwedge$	$\circledwedge$
$\backslash\openJoin$	$\openJoin$	$\backslash\opentimes$	$\opentimes$
$\backslash\lambdaslash$	$\lambdaslash$	$\backslash\lambdabar$	$\lambdabar$
$\backslash\strictiff$	$\strictiff$	$\backslash\circledbar$	$\circledbar$
$\backslash\circledbar$	$\circledbar$	$\backslash\VVdash$	$\VVdash$
$\backslash\VVdash$	$\VVdash$	$\backslash\Wr$	$\Wr$
$\backslash\Wr$	$\Wr$		
$\backslash\idotsint$	$\idotsint$	$\backslash\varointclockwise$	$\varointclockwise$
$\backslash\varointclockwise$	$\varointclockwise$	$\backslash\oiintclockwise$	$\oiintclockwise$
$\backslash\oiintclockwise$	$\oiintclockwise$	$\backslash\varoiintclockwise$	$\varoiintclockwise$
$\backslash\varoiintclockwise$	$\varoiintclockwise$	$\backslash\oiiiiintclockwise$	$\oiiiiintclockwise$
$\backslash\oiiiiintclockwise$	$\oiiiiintclockwise$	$\backslash\varoiiiiintclockwise$	$\varoiiiiintclockwise$
$\backslash\varoiiiiintclockwise$	$\varoiiiiintclockwise$	$\backslash\sqiiint$	$\sqiiint$
$\backslash\sqiiint$	$\sqiiint$	$\backslash\sqiiiint$	$\sqiiiint$
$\backslash\sqiiiint$	$\sqiiiint$		

The full list of Unicode symbols available with Kp-fonts is shown in file `unimath-kpfonts.pdf`.

### 3.3 Wide accents

— `\widehat` and `\widetilde`

$\hat{x}$   $\widehat{xx}$   $\widehat{xxx}$   $\widehat{xxxx}$   $\widehat{xxxxx}$   $\widehat{xxxxxx}$   $\tilde{x}$   $\widetilde{xx}$   $\widetilde{xxx}$   $\widetilde{xxxx}$   $\widetilde{xxxxx}$   $\widetilde{xxxxxx}$

— `\overline` and `\underline`

$$\bar{x} \quad \overline{xy} \quad \overline{xyz} \quad \overline{A \cup B} \quad \overline{A \cup (B \cap C) \cup D} \quad \underline{m+n+p}$$

— `\wideoverbar`, `\widecheck` et `\widebreve`

$$\bar{x} \quad \overline{xy} \quad \overline{xyz} \quad \check{x} \quad \overline{\overline{xxxx}} \quad \overline{\overline{\overline{xxxxxx}}} \quad \check{x} \quad \overline{\overline{xxxx}} \quad \overline{\overline{\overline{xxxxxx}}}$$

— `\overparen` and `\underparen`

$$\widehat{x} \quad \widehat{xy} \quad \widehat{xyz} \quad \overset{\circ}{A \cup B} \quad \overset{\circ}{A \cup (B \cap C) \cup D} \quad \frac{2}{x+y} \quad \overset{26}{a+b+\dots+z}$$

$$\underline{x} \quad \underline{xz} \quad \underline{xyz} \quad \frac{x+z}{2} \quad \frac{a+b+\dots+z}{26}$$

— `\overbrace` and `\underbrace`

$$\overbrace{a} \quad \overbrace{ab} \quad \overbrace{abc} \quad \overbrace{abcd} \quad \overbrace{abcde} \quad \overbrace{a+b+c}^3 \quad \overbrace{a+b+\dots+z}^{26}$$

$$\underbrace{a} \quad \underbrace{ab} \quad \underbrace{abc} \quad \underbrace{abcd} \quad \underbrace{abcde} \quad \underbrace{a+b+c}_3 \quad \underbrace{a+b+\dots+z}_{26}$$

— `\overrightarrow` and `\overleftarrow`

$$\vec{v} \quad \vec{M} \quad \vec{vv} \quad \vec{AB} \quad \vec{ABC} \quad \vec{ABCD} \quad \vec{ABCDEFGH}$$

$$\overleftarrow{v} \quad \overleftarrow{M} \quad \overleftarrow{vv} \quad \overleftarrow{AB} \quad \overleftarrow{ABC} \quad \overleftarrow{ABCD} \quad \overleftarrow{ABCDEFGH}$$

— Enfin `\widearc` and `\widearccarrow` (ou `\overrightarrowarc`)

$$\widehat{AMB} \quad \widehat{AMB}$$

### 3.4 Math Versions

Different versions of the **KpMath** fonts may be defined in the document's preamble:

`\setmathfont{KpMath-Regular.otf}[version=base, options ]`

`\setmathfont{KpMath-Bold.otf}[version=bold, options ]`

`\setmathfont{KpMath-Semibold.otf}[version=semibold, options ]`

`\setmathfont{KpMath-Sans.otf}[version=sans, options ]`

`\setmathfont{KpMath-Light.otf}[version=light, options ]`

then, it is easy to switch from one version to another one with `\mathversion{name}`.

Example <sup>6</sup> :

```
\setmathfont{KpMath-Regular.otf}[Style=leqslant, CharacterVariant=3]
\setmathfont{KpMath-Bold.otf}[version=bold,
    Style=leqslant, CharacterVariant=3]
\setmathfont{KpMath-Sans.otf}[version=sans,
    Style=leqslant, CharacterVariant=3]
```

Here is the same equation in three versions, normal, bold and Sans-Serif:

$$\mathbb{E}_i(N_i) = \sum_{n \geq 1} P_i(N_i \geq n) = \frac{\varepsilon_i}{1 - \varepsilon_i} < +\infty$$

`\mathversion{bold}`

$$\mathbb{E}_i(N_i) = \sum_{n \geq 1} P_i(N_i \geq n) = \frac{\varepsilon_i}{1 - \varepsilon_i} < +\infty$$

`\mathversion{sans}`

$$\mathbb{E}_i(N_i) = \sum_{n \geq 1} P_i(N_i \geq n) = \frac{\varepsilon_i}{1 - \varepsilon_i} < +\infty$$

---

6. Option `CharacterVariant=3` changes  $\epsilon$  into  $\varepsilon$ .