

Antonis Tsolomitis • University of the Aegean • Department of Mathematics

Antonis Tsolomitis • University of the Aegean • Department of Mathematics

A A

[illegible]

The New Computer Modern FontFamily

version 8.1.0

Antonis Tsolomitis

June 4, 2026

Contents

1	Introduction	5
2	License	6
3	Short history of ComputerModern type fonts	7
4	How to load the fonts	7
5	Classic spacing	8
6	The Latin alphabet	8
6.1	Ligatures and stylistic alternatives in Latin	8
6.2	Oldstyle numbers	9
6.3	Old Italic	9
6.4	Diacritics Stacking	9
6.4.1	Coloring diacritics	10
7	Greek	11
7.1	Other character variants	12
7.2	Prosodic symbols	13
8	Greek Sans	13
8.1	Archaic Greek writing	14
8.2	Aegean Numbers	15
8.3	Support for Papyrology	15
8.4	Support for Chemistry	15
9	Cyrillic	16
9.1	Bulgarian Localized Forms	16
9.2	Small Capitals	16
9.3	Example	17
9.4	Design Principles	17

10 Hebrew	18
11 Coptic and Epact Numbers	18
12 Cherokee	18
13 Canadian Aboriginal Syllabics	19
14 Devanagari	19
15 Armenian Script Support	21
15.1 Intended use	23
16 Georgian (Mkhedruli) script support	23
16.1 Design principles	23
16.2 Italics and emphasis	24
16.3 Example: the Pythagorean theorem	24
16.4 Scope	24
17 Medieval Latin and Uncial Greek	25
18 Braille	25
19 IPA symbols	25
20 All Unicode combining marks	27
20.1 Extended and Supplement Combining Diacritical Marks	27
20.2 Diacritical Marks for Symbols and Half Marks	28
21 Editorial Marks	29
21.1 Scope and Intended Use	29
21.2 Combining Enclosing Marks	30
22 Phonetic Extensions and Supplement	30
23 Currency Symbols	31
24 Bold Sans	32
25 LX fonts for presentations	32
26 Food Allergies	33
27 Unicode Math coverage and options	34
27.1 Bold Math	35
27.2 Sans Serif Math	35
27.3 Optical sizes for more glyphs	36
27.4 Math Script	37

27.5	Math Kerning	37
27.6	Blackboard Bold	38
27.6.1	Small Caps for Blackboard Bold	39
27.7	Upright and extensible integrals	40
27.8	Additional and alternative characters in Math	40
27.8.1	Alternative symbol for the empty set	40
27.8.2	Other alternative symbols	41
27.8.3	Additional symbols	41
27.8.4	Additional operators	42
27.8.5	“Smoother” changing radicals	42
27.9	Notes on Mathematics	42
28	The Medieval Latin and Uncial Greek glyph complement	42
29	The Aegean Numbers glyph complement	44
30	Extra Math Relation symbols	45

1 Introduction

The NewComputerModern FontFamily is a huge extension (“huge” in terms of the number of additional glyphs) of the `lm` fonts. It is not just a family adding random missing glyphs but it adds support for several more languages and shapes needed for academic (and not only) work. Currently it supports among others, Greek¹, Cyrillic, Devanagari, Hebrew, Coptic, Cherokee, Canadian Aboriginal, Armenian and Georgian. Since it supports diacritics stacking the number of languages that use the Latin alphabet is greatly expanded. Diacritics stacking is also needed for Greek for papyrological work and this is also supported.

Version 4.0 added to the classic design of computer modern new shapes for Latin and Greek, in particular it adds families for Medieval Latin and Uncial Greek matching in style to the main family.

In terms of weights and sizes, all of its shapes come in Regular, Book weights at 10 and 8 point sizes and in Bold at 10 points.

Mathematics is also supported in Regular, Book, Bold and Sans-Regular weights/shapes, currently providing a full coverage of the Unicode Math blocks (with a few more glyphs needed for Mathematics that Unicode has forgotten to encode).

What follows is a sequence of commands and results so as to show how to access all features of the fonts. Character tables are also included.

IMPORTANT: If you want to provide patches for the fonts please contact me before you create them. The fonts evolve quickly and you may not have the latest development version and your patches may not apply if created for the published version.

¹from Claudio Beccari’s Greek.

2 License

All glyphs, *except* Latin, some basic Math symbols and Greek, are drawn by the author. The fonts are released under GUST license because the Latin Part, a derivative of LM fonts, is released with GUST license, *except* the fonts

NewCM10-Regular, all of NewCMUncial, and all of NewCM*Devanagari,

which are released under GPL3 or later, plus FE (Font Exception) plus DE (Distribution Exception).

Claudio Beccari, the designer of the Greek part, kindly agreed to release the Greek part under GPL3+FE+DE. I also thank Alexey Kuznetsov, who helped me with his knowledge on today's Russian needs, to completely redesign the cyrillic glyphs. This was a much needed work as the Cyrillic of previous releases were erratically copied from OFL fonts (and I apologise for this, as I overlooked OFL). OFL is incompatible with GUST and LPPL which were the licenses of components of NewCM. For diacritic stacking, NewCM needs to have these components in a single font file which is not possible if the components are incompatibly licensed. With this release, since Cyrillic is completely reworked, the fonts are distributed under the GUST license as it respects the conditions of the other component license, ie, LPPL. Any font file that doesn't have the Latin block, eg, Devanagari, is already switched to GPL.

The remaining fonts also are planned to move to GPL in the upcoming versions. Claudio Beccari, the author of Greek fonts, has kindly allowed me to combine his otherwise LPPL licensed fonts in my GPL licensed ones. The Latin block will be completely retraced with Knuthian CM and then the GUST component also will become GPL.

NewCM10-Regular released under this new license scheme does *not* use lm glyphs anymore. The Latin part has been retraced from public domain meta-font sources and hand adjusted glyph by glyph to match the original using fontforge (with its auto composing mechanism for accented glyphs) and the package `pdfpagediff`.

Important notice: The fonts contain thousands of glyphs. So I may have accidentally not removed some glyph from lm. This is not done on purpose and I will remove it and replace it immediately if I get notified for such a bug.

The details of Font Exception and Distribution Exception follow:

Font Exception If you create a document which uses this font, and embed this font or unaltered portions of this font into the document, this font does not by itself cause the resulting document to be covered by the GNU General Public License. This exception does not however invalidate any other reasons why the document might be covered by the GNU General Public License. If you modify this font, you may extend this exception to your version of the font, but you are not obligated to do so. If you do not wish to do so, delete this exception statement from your version.

Distribution Exception If you distribute the fonts as they are in your program, and your program follows a license compatible with GPL version 3

(or later) as described in <https://www.gnu.org/licenses/license-list.en.html> these fonts do not by themselves cause the resulting program to be covered by the GNU General Public License. This exception does not however invalidate any other reasons why the program might be covered by the GNU General Public License. If however you distribute a copy of the fonts that modifies either the glyphs (one or more) or the glyph-set by adding or removing glyphs, this exception is invalidated and your program has to follow GPL version 3 (or later).

3 Short history of ComputerModern type fonts

For historical reasons we provide a list of copyright holders of CM-type fonts up today. The licenses of these fonts do not apply to NewCM as no glyphs are copied from these except GUST licensed fonts.

- Copyright © Authors of original metafont fonts: Donald Ervin Knuth (cm & concrete fonts).
- 1995, 1996, 1997 Jörg Knappen, 1990, 1992 Norbert Schwarz (ec fonts).
- 1992–2006 A. Khodulev, O. Lapko, A. Berdnikov, V. Volovich (lh fonts).
- 1997–2005 Claudio Beccari (cb greek fonts and lx fonts).
- 2002 Fukui Rei (tipa fonts).
- 2003–2005 Hàn Thế Thành (Vietnamese fonts).
- 1996–2005 Walter Schmidt (cmbright fonts).
- 2003–2021 B. Jackowski and J. M. Nowacki (lm-modern fonts).
- Copyright © 2003–2009, Andrey V. Panov (panov@canopus.iacp.dvo.ru) with Reserved Font Family Name “Computer Modern Unicode fonts”.
- Copyright © 2012–2014, Andrey V. Panov (panov@canopus.iacp.dvo.ru) (cyrillic-modern).
- Copyright © 1997, 2009, American Mathematical Society (<http://www.ams.org>). All Rights Reserved.

4 How to load the fonts

The simpler way to load the fonts is through the `fontsetup` package. The command

```
\usepackage[default]{fontsetup}
```

will load the Book weight of the NewCM family,

```
\usepackage[olddefault]{fontsetup}
```

will load the Regular weight, and

```
\usepackage[sansdefault]{fontsetup}
```

will load the Sans Serif NewCM family.

Notice that the Mono family has Italic fonts only for Regular and Book weights but for the Bold it has Oblique. To avoid confusion fontspec files are provided that provide the Oblique shape for Regular and Book using the Fake-Bold fontspec parameter.

Also notice that the fonts support the microtype package for fine typographic tuning. See the documentation of microtype for this.

5 Classic spacing

Alexey Kuznetsov kindly contributed fontspec files with stretching factors that micro-adjust the fonts to be as close as possible to original metafont output. These are in the folder `tex/contrib/`. The files are marked with the string “cs” (for “classically stretched”). For example `NewCMcs.fontspec` micro-adjusts the Book Roman weight, and `NewCMcs-Regular.fontspec` for the Regular Roman fonts. So if you need to mimic as close as possible the original “Knuthian” output you can load the fonts with

```
\setmainfont{NewCMcs}  
\setsansfont{NewCMcsSans}  
\setmonofont{NewCMcsMono}
```

or

```
\setmainfont{NewCMcs-Regular}  
\setsansfont{NewCMcsSans-Regular}  
\setmonofont{NewCMcsMono-Regular}
```

6 The Latin alphabet

6.1 Ligatures and stylistic alternatives in Latin

The Serif font includes additional ligatures `fb ffb ffh ffj ffk fft fh fj ft fk` and the same with longs instead of `f` in the *default* liga table (in addition to the default `fi fl ffi ffl ff`). It also includes an alternative `k` (in the `cv01` table) and `$p ch ck ct st il` in the `dlig` table. Finally it also includes “end” versions for the letters `a`, `e`, `m`, `n` and `r` in the `cv02` table. To access the alternative `k` load the relative font (here the Book weight) with

```
\setmainfont[CharacterVariant=1]{NewCM10-Book.otf}
```

To load the same font with the `dlig` table enabled use

```
\setmainfont[RawFeature+=dlig]{NewCM10-Book.otf}
```

and to load the font with endings variations use

```
\setmainfont[CharacterVariant=2]{NewCM10-Regular.otf}
```


Of course the above can be mixed separating the optional arguments with comma, or one can define a custom font say by using

```
\newfontfamily\myfont[<options to enable>]{NewCM10-Book.otf}
```

Book	k	a e m n r	sp ch ck ct st il
cv01	k		
cv02		a e m n r	
dlig			sp ch ck ct st il

6.2 Oldstyle numbers

Typically oldstyle numbers are available in `onum` Lookup and with the `\textsc` if `fontsetup` is loaded. Also available they are with `\oldstylenums`. There are two series, one is with variable widths and one with fixed width for use in tables. The code

```
\oldstylenums{0123456789}\addfontfeatures{Numbers=Tabular}
\textsc{0123456789}
```

gives

```
0123456789
0123456789
```

An alternative design is also provided for the number 1 in `cv06`. The code

```
\oldstylenums{0123456789}\addfontfeatures{CharacterVariant=6}
\oldstylenums{0\textcolor{red}{1}23456789}
\addfontfeatures{CharacterVariant=6,Numbers=Tabular}
\oldstylenums{0\textcolor{red}{1}23456789}
```

gives

```
0123456789
0I23456789
0I23456789
```

6.3 Old Italic

The fonts also fully support the Old Italic Unicode block (U+10300–U+1032F) in the Sans font. For example, the slots U+10307, U+10310, U+10312, U+10314, U+1031F and U+1032F are 𐌱 𐌲 𐌳 𐌴 𐌵 𐌶.

6.4 Diacritics Stacking

Diacritics—the full block U+0300 to U+036F—and diacritics stacking is supported. In the margin you can see an example of stacking on the letter “x” in Roman, Sans and Mono. If you need to enter these accents you can use the `\char` command or just copy-paste from the following line (from this pdf file or the provided source `TEX` file):

→ (U+0300) (U+0301) (U+0302) (U+0303) (U+0304) (U+0305) (U+0306) (U+0307) (U+0308) (U+0309) (U+030A) (U+030B) (U+030C) (U+030D) (U+030E) (U+030F) (U+0310) (U+0311) (U+0312) (U+0313) (U+0314) (U+0315) (U+0316) (U+0317) (U+0318) (U+0319) (U+031A) (U+031B) (U+031C) (U+031D) (U+031E) (U+031F) (U+0320) (U+0321) (U+0322) (U+0323) (U+0324) (U+0325) (U+0326) (U+0327) (U+0328) (U+0329) (U+032A) (U+032B) (U+032C) (U+032D) (U+032E) (U+032F) (U+0330) (U+0331) (U+0332) (U+0333) (U+0334) (U+0335) (U+0336) (U+0337) (U+0338) (U+0339) (U+033A) (U+033B) (U+033C) (U+033D) (U+033E) (U+033F) (U+0340) (U+0341) (U+0342) (U+0343) (U+0344) (U+0345) (U+0346) (U+0347) (U+0348) (U+0349) (U+034A) (U+034B) (U+034C) (U+034D) (U+034E) (U+034F) (U+0350) (U+0351) (U+0352) (U+0353) (U+0354) (U+0355) (U+0356) (U+0357) (U+0358) (U+0359) (U+035A) (U+035B) (U+035C) (U+035D) (U+035E) (U+035F) (U+0360) (U+0361) (U+0362) (U+0363) (U+0364) (U+0365) (U+0366) (U+0367) (U+0368) (U+0369) (U+036A) (U+036B) (U+036C) (U+036D) (U+036E) (U+036F)

←

Some of the upper accents

\ / ^ ~ - — ˇ ˙ ˚ ˚ ˚ ˚

˘ ˙ ˚ ˚ ˚ ˚ ˚ ˚ ˚ ˚ ˚ ˚

Some of the lower accents

˘ ˙ ˚ ˚ ˚ ˚ ˚ ˚ ˚ ˚ ˚ ˚

˘ ˙ ˚ ˚ ˚ ˚ ˚ ˚ ˚ ˚ ˚ ˚

Please note that stacking is by default supported with xetex. With luatex you have to add the option `Renderer=HarfBuzz`, say by `\addfontfeatures{Renderer=HarfBuzz}` however this will probably affect hyphenation.

Also notice that your text editor may not support stacking. The editor may show the accents one after the other, but the pdf produced by xetex or luatex will have the accents stacked.

6.4.1 Coloring diacritics

If one wants to use color for diacritics, different from the color of the base character this does not work with $\text{Xe}^{\text{L}}\text{A}^{\text{T}}\text{E}^{\text{X}}$ (the commands of the `color` package break the stacking mechanism). It works though with $\text{Lua}^{\text{L}}\text{A}^{\text{T}}\text{E}^{\text{X}}$ using the `luacolor` package. However, there is a problem when the base glyph and the first diacritic above exist in the font as a precomposed character. For example, this is the case with aacute (á) (U+00E1). Such characters are treated as one by Lua and they can not be colorized with different colors. A work around is to place the empty character U+034F between “a” and acute (U+0301). So the following minimal example produces the result below:

```
\documentclass{article}
\usepackage[olddefault]{fontsetup}
\usepackage{luacolor}
\pagestyle{empty}
\newfontfamily{\ncmtest}[Renderer=Harfbuzz]{NewCM10-Regular.otf}
\definecolor{orange}{RGB}{255,191,0}
\definecolor{colorone}{RGB}{91,0,250}
\definecolor{colortwo}{RGB}{250,0,121}
\definecolor{colorthree}{RGB}{0,204,250}
\definecolor{colorfour}{RGB}{14,250,0}
\definecolor{colorfive}{RGB}{255,150,0}
\definecolor{colorgray}{gray}{0.8}
\newcommand{\emptydiacritic}{\char"034F}
\begin{document}
\Huge
{\ncmtest \color{colorgray}a\color{colorfour}\color{colortwo}%
\emptydiacritic\color{colorthree} \color{colorone} \color{colorfive} }
\end{document}
```



7 Greek

The full Unicode Greek block is supported, which is

- U+0370–U+03FF for monotonic, where missing glyphs, such as Heta (Ͱ), Pamphilian digamma (ͱ) etc, have been added. For example, it is now possible to write

βιβλίο instead of βιβλίο.

In order to auto select this conversion for middle beta and theta the StylisticSets `ss06` and `ss07` must be enabled with, say,

```
\addfontfeatures{StylisticSet=6,StylisticSet=7}
```

Important: if you add font features like the above in the preamble you need to set them with `\AddToHook{begindocument/end}`. So in the preamble you must have

```
\AddToHook{begindocument/end}{%
\addfontfeatures{StylisticSet=6,StylisticSet=7}}
```

To disable this feature you can do

```
\addfontfeature{RawFeature={-ss06},RawFeature={-ss07}}
```

Notice also that the original Beccari fonts for Greek used only the open theta ϑ. However, this violates the Unicode standard description for this glyph which proposes the closed theta θ. If you want the original behavior you need to enable `ss06`.

Source	βιβλίο	θυμήθηκα
<code>ss06</code> enabled	βιβλίο	θυμήθηκα
<code>ss06</code> and <code>ss07</code> enabled	βιβλίο	θυμήθηκα
<code>ss06</code> and <code>ss07</code> disabled	βιβλίο	θυμήθηκα

- U+1F00–U+1FFF for polytonic, and
- U+10140–U+1018F for ancient Greek numbers.

Θεώρημα 7.1 (Πυθαγόρειον) Ἐν τοῖς ὀρθογωνίοις τριγώνοις τὸ ἀπὸ τῆς τὴν ὀρθὴν γωνίαν ὑποτείνουσας πλευρᾶς τετράγωνον ἴσον ἐστὶ τοῖς ἀπὸ τῶν τὴν ὀρθὴν γωνίαν περιεχουσῶν πλευρῶν τετραγώνοις.

Small Caps is included (in Mono font too) and all polytonic accents of Greek. Ypogegrammeni is the default for all characters including Small Caps and pros-grammeni is offered as an alternative shape in the `ss01` lookup table:

	ypogegrammeni	prosgegrammeni
regular	Ᾱ ῥῖ ῖΩ ἈῤῖΩ	Ᾱ _ı ῥ _ı ῖ _ı Ω _ı Ἀ _ı ῤ _ı Ω _ı
sans	Ᾱ ῥῖ ῖΩ ἈῤῖΩ	Ᾱ _ı ῥ _ı ῖ _ı Ω _ı Ἀ _ı ῤ _ı Ω _ı
mono	Ᾱ ῥῖ ῖΩ ἈῤῖΩ	Ᾱ _ı ῥ _ı ῖ _ı Ω _ı Ἀ _ı ῤ _ı Ω _ı

The prosgegrammeni alternates can be accessed with

```
\textprosgegrammeni{<text>}
or the
{\prosgegrammeni <text>}
```

of the `fontsetup` package.

7.1 Other character variants

Guillemets (left and right) have a different shape for Greek. For this to work the fonts must be loaded with the `cv04` character variant.

Compare the default guillemets: «» with Greek guillemets: «».

There is a serious problem with Unicode and the Greek anoteleia (U+0387); the Greek semicolon. Unicode “thinks” that this character is the same with periodcentered (U+00B7). This influences the way keyboards are configured by several vendors such as xorg. Anoteleia is a dot written at x-height and not at 1/2 the x-height as the periodcentered. Although Unicode recognizes the problem², although they recognize that with their current standard you can not correctly write the Greek language, they refuse to fix it, justifying it by saying the magical words “backwards compatibility” (to a ...mistake, one could add).

NewComputerModern can not allow this, as it defies the purpose of its existence, which is to properly write every supported language. So enabling the CharacterVariant 04 (`cv04`) in addition to correct guillemets for Greek it maps periodcentered (produced by the keyboards (in Greek Linux keyboards by `AltGr+q`) to proper anoteleia.

It also fixes a long standing issue with the Greek apostrophe (') (U+1FBD) which is not the same with quoteright (') (U+2019). U+1FBD named as “Greek Koronis” by Unicode is the proper character.

Another problem that has to do with quotes inside quotes. The internal quotes in Greek should be written with the characters `quotedblleft` and `quotedblbase`³. For example, this is correct for Greek

«άλφα “βήτα»,»

But the keyboards only produce `quotesingle` which is already mapped to apostrophe and it is difficult to remember the names “`quotedblleft`” and “`quotedblbase`”. So when enabling `cv04` one can define the commands

²personal communication

³Μανόλης Τριανταφυλλίδης, *Νεοελληνική Γραμματική της δημοτικής*, Ανατύπωση της έκδοσης του ΟΕΣΒ 1941 με διορθώσεις, Θεσσαλονίκη 2002, σελ. 66, ενότητα 133.

`\newcommand\leftgrquotes{\char"201C}`
and
`\newcommand\rightgrquotes{\char"201D}`
for the rare case one needs quotes inside quotes. The `fontsetup` package does this automatically for Greek if the `xgreek` package has been loaded *before* the `fontsetup` package or when the language is set to Greek by, say, the Babel package. Otherwise, for non-Greek documents with small passages of Greek, the author may enable `cv04` by creating a custom command such as
`\newfontfamily\propergreek[CharacterVariant=4]{NewCM10-Book.otf}`
A phrase with Greek quotes inside quotes, proper anoteleia, and proper apostrophe is

«φώναζε: “ἀπ’ ἐξω τὴν προπαίδεια,,»· σαν ἐκδίχηση ἀκουγόταν...

7.2 Prosodic symbols

In Greek philology and in linguistics it is often needed to stack accent-type symbols above letters, even if they are not vowels. Although rare in writings, it is for example valid to place dieresis over the consonants π , τ and χ of the nasal complexes $\mu\pi$, $\nu\tau$, $\gamma\chi$ when it is necessary to show that these are pronounced, as written, voiceless, and not voiced. For example,

κομῖπρέσα, αντιανῖτανῖτικός, ελεγχῖτής

(see previous footnote). The fonts support this writing if a combining dieresis is placed after the letter to receive it. The combining dieresis is the character U+0308. On Linux desktops this is easily entered pressing Alt+Control+u, release them, and type the sequence 0308 and space.

More than that, in linguistics, they need to combine several accents above Greek letters. All this stacking of accents is supported by the fonts. For example, one can write

ᾱ̃ ῃ̃ Ἀ̃ Ἀ̃̃ Ὼ̃ ᾱ̂ Ḳ̂ ṭ̂

by placing the combining accents from the Unicode block U+0300–U+0362 plus the usual Greek accents *after* the letter. So the above was typed as

α^{-~} ε^{’~} Α[˘] Ἀ⁻ Ὠ^{’~} α^{-’} Ḳ^{˘’} ṭ^{-˘˘}

8 Greek Sans

The Greek Sans in previous releases of New Computer Modern was largely inherited from earlier designs and had not been systematically re-evaluated under modern usage conditions, in particular mixed Latin–Greek documents, dense mathematical text, and projected presentations (*beamer*). As a result, several long-standing issues were present: uneven paragraph color compared to Latin Sans, excessive local darkness in round letters (notably σ , θ , ω), inconsistent behavior of diacritics at small sizes, and reduced robustness under low-resolution

projection. While these issues were often tolerable in print, they became clearly visible in contemporary workflows that rely on screen reading and projection.

The redesigned Greek Sans included in this release is a ground-up revision that preserves the spirit and proportions of the original design while carefully rebalancing stroke weight, junctions, and counters to achieve uniform paragraph color and stable interaction with Latin Sans. Particular attention was paid to round and semi-round letters, final sigma, capital forms, and the full system of monotonic and polytonic diacritics, with extensive testing in mixed-script text, mathematical environments, and real projector conditions. The result is a Greek Sans that integrates naturally with Latin Sans, remains calm and legible in dense academic material, and is robust in presentation settings, without sacrificing typographic discipline or historical continuity.

8.1 Archaic Greek writing

The Sans Serif Regular font provides access to 6th century bce and 4th century bce Greek capitals in ss04 and ss03 lookups. The `fontsetup` package provides commands such as

`\textivbce{}`, `\ivbce`, `\textvibce{}` and `\vibce`

6th century bce:

~ΒΔΕΙΣ ΑΓΕΩ~ΕΤϐΒΤΟΣ ΕΙΣΙΤΩ

4th century bce:

ΜΗΔΕΙΞ ΑΓΕΩΜΕΤΡΗΤΟΞ ΕΙΞΙΤΩ

Moreover, all fonts (except Mono & Math) support Ancient Greek Numerals (the full Unicode block of Greek digits U+10140–U+1018E is supported), with most symbols designed from scratch (and did not exist in C. Beccari’s original fonts). A few of the new symbols:

ΗϞϘϙϒ^ΜϜ

The four numerals that already existed in this range (that is U+10144–U+10147) in Beccari’s fonts have been altered to a new design matching the style of cm but also provide some Ancient Greek flair. The new designs in Serifed and SansSerifed are:

ϠϡϢϣ ϡϢϣϣ

The `fontsetup` package provides commands for all of the above symbols. The commands follow the Unicode name of each slot (minus the “Greek Acrophonic”). So the Unicode slot U+1014F named “Greek Acrophonic Attic Five Staters” can be accessed with the command `\atticfivestaters` and it gives ϡ; and the slot U+10182 named “Greek Kyathos Base Sign” can be accessed with the command `\greekkyathosbasesign` and it gives Ϟ.

8.2 Aegean Numbers

Aegean numbers are supported in the Sans fonts and their slots are defined in `fontsetup` package using commands of the form `\aegeanXXXX` where `XXXX` is the Unicode name of the character (without spaces). A few examples are:

𐀀 𐀁 𐀂 𐀃 𐀄 𐀅 𐀆 𐀇 𐀈 𐀉 𐀊 𐀋 𐀌 𐀍 𐀎

and the whole table of Aegean Numbers with the commands to access the glyphs is shown on page 44.

8.3 Support for Papyrology

Papyrology needs to declare that a glyph is missing from the papyrus or the papyrus is worn out at this point and the papyrologist adds the missing glyph but it is not clear from the papyrus. This is done by adding a dot below the glyph and it is supported for all Greek glyphs in the upright fonts monotonic or polytonic:

Α̣ Α̣̣ Η̣ Ξ̣ Ω̣ ρ̣

where in the source we just typed the dot below (char U+0323) after the glyph. This feature is supported for the 4th bce and 6th bce Greek in Sans:

ΓΕΩΜΕΤΡΙΑ ΓΕΩΜΕΤΡΙΑ

More support for papyrology is available. Check Section 22.

8.4 Support for Chemistry

It happens often that Greek upright characters are needed in Chemistry. People often have trouble with this (and this is why packages such as `chemgreek` exist). If Greek keyboard is available then it is easy; you just type in Greek, say β -glucan to get “ β -glucan”. But many writers do not have the Greek keyboard enabled, and they do not need to. Usually they type `$\beta $-glucan` but the result “ β -glucan” is not satisfying. One can use the “up” versions typing `$ \upbeta $-glucan` but still the result “ β -glucan” looks more Math than Chemistry. To help with this, the `fontsetup` package provides commands such as `\chemAlpha`, `\chemalpha`, `\chemBeta`, `\chembeta`, etc. So this information essentially would only belong to the `fontsetup` documentation if it was not for kappa and rho. If we type in Greek κ -compound we get “ κ -compound” which is not satisfying, as kappa is too cursive for this use. So the NewCM family provides an alternative kappa for this reason and this is how `\chemkappa` is defined in `fontsetup`:

```
\newcommand{\chemkappa}{\textrm{\char"03F0}}:
```

We write `\chemkappa-compound` and now get “ κ -compound”.

(The `\textrm` command in the above definition is there to make the command work in math mode too.) Similar is the situation for `\chemrho` (ρ) and `\chemrhoalt` (ϱ).

Especially for Chemistry notice that the MathFonts contain the “reaction arrows”, that is U+1F8D0–U+1F8D4 (\rightleftharpoons , \rightleftharpoons , \rightleftharpoons , \rightleftharpoons , \rightleftharpoons), unsuccessful reaction arrows U+1F8D6 (\nrightarrow) and U+1F8D7 (\nrightarrow) and the isolobal arrow U+1F8D8 (\leftrightarrow). You can directly type these symbols in MathMode (consult your editor manual on how to do this). or create commands such as

```
\Umathchardef\isolobalarrow = "0 "0 "1F8D8
and then use  $\isolobalarrow$ 
```

9 Cyrillic

The NewComputerModern font family includes full support for the modern Cyrillic alphabet. The implementation is intended for academic typesetting and therefore aims at typographic compatibility with the Latin and Greek components of the family. Particular care has been taken to ensure consistent stroke weight, proportions, and text color across scripts, so that multilingual documents remain visually harmonious.

The Cyrillic repertoire covers the standard alphabets used in Russian, Bulgarian, Ukrainian, Serbian, and North Macedonian.

9.1 Bulgarian Localized Forms

Bulgarian typographic tradition uses distinct shapes for several lowercase letters that differ from the forms commonly used in Russian typography. NewCM provides these forms through the OpenType `locl` feature. When the Bulgarian language is selected via a language-aware package (for example `polyglossia` or `babel`), the appropriate glyph substitutions are applied automatically.

The following five letters have Bulgarian-specific designs:

Default Cyrillic	Bulgarian forms
б г д п т	б г д п т

These alternate forms affect only the lowercase alphabet. Uppercase letters are common across Cyrillic typographic traditions and therefore remain unchanged.

9.2 Small Capitals

The NewCM Cyrillic implementation also includes small capitals. These are designed to match the proportions and weight of the Latin small capitals in the family, allowing consistent typographic hierarchy in multilingual documents.

Small caps are accessed in the usual way via font features. For example:

`\addfontfeatures{Letters=SmallCaps}`

or with the standard L^AT_EX command

`\textsc{текст на кирилица}`

The Cyrillic small capitals follow the same design principles as the Latin ones: slightly increased letterspacing and carefully adjusted stroke weight ensure that they remain legible and visually balanced in continuous text.

9.3 Example

The following example illustrates Cyrillic text with Bulgarian localization and small capitals:

Това е примерен текст на кирилица.

БЪЛГАРСКИ ТЕКСТ СЪС МАЛКИ ГЛАВНИ БУКВИ.

Това е примерен текст на кирилица.

БЪЛГАРСКИ ТЕКСТ СЪС МАЛКИ ГЛАВНИ БУКВИ.

The result demonstrates the integration of Cyrillic glyphs within the typographic system of the NewCM family.

Я помню чудное мгновенье:
Передо мной явилась ты,
Как мимолетное виденье,
Как гений чистой красоты.
(Пушкинъ)

Again, as in Greek there is a different kind of guillemets for Cyrillic which are available in CharacterVariant 3 (cv03). Compare:

Defaults guillemets: «» Cyrillic guillemets: «» Greek guillemets: «»

Same is the situation with Cyrillic emdash which is shorter than the default:

Default emdash: —

Cyrillic emdash: —

9.4 Design Principles

The Cyrillic component of NewCM has been designed with the same typographic philosophy that guides the Latin and Greek parts of the family. The goal is not only to provide Unicode coverage, but to ensure that Cyrillic text integrates naturally within documents that mix multiple scripts, as is common in academic publications.

Several aspects were considered during the design process. First, the overall proportions and x-height were aligned with the Latin alphabet so that line rhythm remains consistent when the scripts appear together. Second, the stroke

modulation and contrast were calibrated to match the visual weight of the surrounding Latin and Greek glyphs. This ensures a uniform typographic “color” across multilingual paragraphs.

Particular attention was given to letters that typically dominate the texture of Cyrillic text, such as *u*, *u*, *ж*, and *ф*. Their internal counters and stroke distribution were carefully balanced so that dense sequences remain readable even at smaller sizes.

The Bulgarian localized forms were implemented following modern typographic practice and are derived from the cursive structures traditionally used in Bulgarian typesetting. Their design was adapted to the stylistic character of the NewCM family so that the localized forms remain compatible with the rest of the typeface while preserving their distinctive identity.

Finally, the Cyrillic small capitals were developed to match the Latin small-cap system of NewCM. Their proportions, stroke thickness, and spacing were adjusted to produce a consistent appearance in headings, running text, and scholarly apparatus such as abbreviations or bibliographic references.

These design choices aim to make the NewCM Cyrillic extension suitable for demanding academic applications, including multilingual scientific publications, lecture notes, and typographically complex mathematical documents.

10 Hebrew

The Hebrew blocks U+0590–U+05FF and Hebrew Presentation forms U+FB1D–U+FB4F are fully covered. A few letters from Hebrew:

אבגדהוזטז

11 Coptic and Epact Numbers

The Coptic language is fully supported. This covers the Coptic blocks in the Greek and Coptic Unicode block (U+03E2–U+03EF), the full Coptic Unicode block (U+2C80–U+2CFF) and the Coptic Epact Numbers (U+102E0–U+102FF). A few letters from Coptic and Epact numbers follow:

ΛΟΓΟΣ ΝΑΙΓΥΠΤΙΟΣ Ϸ ω υ Ϸ ε

12 Cherokee

Both Unicode blocks U+13A0–U+13FF and U+AB70–U+ABBF for Cherokee are supported. The samples below were kindly provided by Sedi Eastwood:

Sample 1:

EGPCAG66A6MA TTP6MA
ECPCAC66A6MA TTP6MA

Live/exist in a manner that there is
never a reason or purpose to let go of
one another.
(gvjalijvdijadayohisdi ijehesdi
gvtsalitsvditsadayohisdi itsehesdi)

Sample 2:

SG6FG46MA
SC6FG46MA

Value the existence of one another.
(dejadageyusesdi
detsadageyusesdi)

Sample 3:

iP6FMA SG6β946MA
iP6FMA SC6β946MA

Think of one another as sacred or hold
the existence of others sacred.
(vlisgedi dejadayelvsesdi)

13 Canadian Aboriginal Syllabics

Canadian Aboriginal Syllabics are supported in the sans font. The full Unicode blocks are covered, which are U+1400–U+167F, U+18B0–U+18F5, and U+11AB0–U+11ABF.

Some examples in Blackfoot Siksiká:

Blackfoot-speaking real people: ᓄᓴᓴᓴ Siksiká: ᓴᓴᓴ

Siksikáí'powahsin: ᓴᓴᓴᓴᓴᓴᓴ In bold: ᓴᓴᓴᓴᓴᓴᓴᓴ

Niitsipowahsin: ᓴᓴᓴᓴᓴᓴ In oblique: ᓴᓴᓴᓴᓴᓴ

Some more in Cree:

ᐃᐃᐃᐃ ᐃᐃ ᐃᐃᐃᐃᐃᐃᐃᐃ ᐃᐃᐃᐃᐃᐃ
ᐃᐃᐃᐃᐃᐃ ᐃᐃᐃᐃᐃᐃᐃᐃᐃ ᐃᐃᐃᐃᐃᐃᐃ ᐃᐃ
ᐃᐃᐃᐃᐃᐃᐃᐃ ᐃᐃᐃᐃᐃᐃᐃᐃᐃᐃᐃᐃᐃᐃᐃᐃᐃ
ᐃᐃᐃᐃᐃᐃᐃᐃ ᐃᐃᐃᐃᐃᐃᐃᐃᐃᐃᐃᐃᐃᐃᐃᐃᐃ
ᐃᐃᐃᐃᐃᐃᐃᐃᐃ ᐃᐃᐃᐃᐃᐃᐃᐃᐃᐃᐃᐃᐃᐃᐃᐃᐃᐃ

The masonry for the wall through
time has been given to people
of Manitoba by the international
union of bricklayers and allied
craftsmen local one Manitoba to
commemorate 100 years of ser-
vice, by Manitoba masonry con-
tractors and by Manitoba ma-
sonry suppliers.

14 Devanagari

Devanagari script is supported for the serifed font in Regular (10pt/8pt), Book (10pt/8pt), and Bold (10pt). The fonts support Hindi (as the default), Sanskrit, Marathi and Nepali Languages. The optional arguments for the **fontspec** font-selection mechanism must include

Script=Devanagari, Language=XXXX where XXXX must be replaced with one of Hindi, Sanskrit, Marathi, Nepali. If the Language parameter is not set

then the default is Hindi. For Lua^AT_EX the parameter `Renderer=Harfbuzz` must also be included.

So if say Marathi is needed as the default font document then one can use the following:

```
\usepackage{fontspec}
\setmainfont[Script=Devanagari, Language=Marathi,%
Renderer=Harfbuzz]{NewCM10Devanagari-Book.otf}
```

The Devanagari fonts were developed with the help of निरंजन (Niranjan) whose name appears in the copyright section of the fonts and I also thank him for providing the samples below. It should also be noted that the design is original and based on old handwritten books. In old civilizations, such as the Indian one, it is only natural that the design be affected by how the handwritten books look. Ink creates “drop”-like serifs on paper that absorbs it, and the tool used to write also affects the look of the script. All these were taken into account. A Sanskrit sample from बृहदारण्यकोपनिषद् (brhadāraṇyakopaniṣad) follows:

ॐ पूर्णमदः पूर्णमिदं, पूर्णात्पूर्णमुदच्यते।
पूर्णस्य पूर्णमादाय पूर्णमेवावशिष्यते॥

That^a is complete;
this^b too is complete.
From one complete comes the other. Taking out
one complete from the other too results in a complete.

^athe outer world
^bthe inner world

Next is a beautiful part of a poem in Marathi by तुकाराम (Tukaram) and its translation:

जें कां रंजलें गांजलें।
त्यांसि म्हणे जो आपुलें॥ १ ॥
तो चि साधु ओळखावा।
देव तेथें चि जाणावा॥

Only the one who treats the downtrodden people equally is a sage^a.
One may sense the essence of god there.

^a “The wise” of course, not the plant.

Devanagari Unicode letters (range U+0900–U+097F) are also available as variables (letters) and numbers in the Regular and Book Math fonts. They are available as usually in three weights in the Math fonts so that the color is balanced when in script size (eg in exponents or indices). For this to work a version of `fontsetup` package greater or equal to 1.8 with options `default` or `olddefault` loaded is needed. This is because Devanagari letters are not Math variables in Unicode standard and hence not supported currently as such by the unicode-math package. To show this possibility next is a theorem in Hindi (mixing with Greek):

प्रमेय (Πυθαγόρας (पिथागोरास)) अगर समकोण त्रिभुज के कर्ण की लंबाई को ‘अ’ और अन्य दो भुजाओं की लंबाई को ‘क’ और ‘ख’ कहते हैं, तो भुजाओं की लम्बाई के वर्गों की जोड़, कर्ण के वर्ग जितनी होती है, अर्थात् $अ^२ = क^२ + ख^२$ ।

However, if only Devanagari numbers are needed with the source using arabic numerals then one can use the Stylistic Set 04 of the Math font. So the command `\setmathfont[StylisticSet=4]{NewCMMath-Regular.otf}` with source:

```
\[\sum_{n=0}^{\infty} \frac{1}{n!}x^n=1+x+\frac{1}{2!}x^2
+\frac{1}{3!}x^3+\frac{1}{4!}x^4+\ldots=e^x.\]
\[9!=1\cdot2\cdot3\cdot4\cdot5\cdot6\cdot7\cdot8\cdot9.\]
\[2^{\{2^2\}}\]
```

will have the following effect:

$$\sum_{n=0}^{\infty} \frac{१}{n!} x^n = १ + x + \frac{१}{२!} x^2 + \frac{१}{३!} x^3 + \frac{१}{४!} x^४ + \dots = e^x.$$

$$१! = १ \cdot २ \cdot ३ \cdot ४ \cdot ५ \cdot ६ \cdot ७ \cdot ८ \cdot ९.$$

$$2^{2^2}$$

which when the math font is reset to use Arabic numbers with `\setmathfont{NewCMMath-Book.otf}` it gives:

$$\sum_{n=0}^{\infty} \frac{1}{n!} x^n = 1 + x + \frac{1}{2!} x^2 + \frac{1}{3!} x^3 + \frac{1}{4!} x^4 + \dots = e^x.$$

$$9! = 1 \cdot 2 \cdot 3 \cdot 4 \cdot 5 \cdot 6 \cdot 7 \cdot 8 \cdot 9.$$

$$2^{2^2}$$

15 Armenian Script Support

New Computer Modern, as of version 8.0.0 provides full, native support for the Armenian script, designed with the same typographic principles that govern the Latin, Greek, and Cyrillic components of the family. Armenian is treated as a first-class script across all major styles, weights, and optical sizes. A sample follows.

Պյութագորասի թեորեմը մաթեմատիկայի ամենակարևոր թեորեմներից մեկն է:

Թեորեմ (Πυθαγόρας) Մեն ուղղանկյուն եռանկյան մեջ, կառուցելի վրա կառուցված երկու քառակուսիների մակերեսների գումարը հավասար է հիպոտենուզայի վրա կառուցված քառակուսու մակերեսին, այսինքն $\alpha^2 = \beta^2 + \gamma^2$:

Armenian is available in:

- Families: Serif (NewCM), Sans (NewCMSans), Mono (NewCMMono)

- Series: Regular, Book, Bold
- Shapes: Roman, Italic (Serif), Oblique (Sans), Bold Italic / Bold Oblique
- Optical sizes: 10 pt and 8 pt designs (with independent drawings)

The Armenian design in NewCM follows three guiding principles:

- **Typographic neutrality:** Armenian glyphs are designed to blend seamlessly with Latin and Greek text in academic and technical documents, without drawing undue attention or altering line color.
- **Script-specific structure** Armenian joins, counters, and internal rhythm are handled according to the structural logic of the script, not by mechanical reuse of Latin shapes. Particular care is taken with join-heavy letters and complex counters to preserve clarity at text sizes.
- **Consistency across families** Serif, Sans, and Mono Armenian share the same underlying proportions and rhythm, while respecting the constraints of each family (contrast in Serif, openness in Sans, grid discipline in Mono).

Serif Armenian is designed in a Didot-influenced style, consistent with NewCM’s high contrast and vertical stress.

Italic Armenian is a true italic, not a slanted Roman, with moderated slant to respect Armenian letter structure.

Armenian ligatures defined by Unicode (U+FB13–U+FB17) are supported via standard OpenType liga features.

Separate 8 pt designs increase stroke robustness and counter openness for small-size text.

Sans Armenian is drawn independently, not derived from the Serif outlines.

Italic is implemented as an oblique, with script-sensitive slant values (distinct for uppercase and lowercase).

Stroke weights are adjusted to account for Armenian’s higher internal density, ensuring even color in mixed-script text.

Suitable for modern academic layouts, captions, headings, and presentations.

Mono Armenian is fully monospaced, with all glyphs sharing a fixed advance width.

The design prioritizes grid predictability and alignment, making it suitable for code, tables, and technical material.

Joins and counters are optimized for clarity at small sizes without compromising alignment.

Mathematics and Armenian integrates cleanly with NewCM’s math fonts:

Mixed Armenian–Latin–Greek text maintains consistent color and hierarchy.

Bold and italic Greek symbols used in mathematics harmonize visually with Armenian Bold and Italic text.

This makes NewCM suitable for scholarly publications involving Armenian text alongside formal mathematics.

15.1 Intended use

NewCM’s Armenian support is intended for:

- Academic books and journals
- Critical editions and linguistic work
- Mathematical and scientific texts
- Multilingual documents combining Armenian with Latin and Greek
- Technical material requiring monospaced layout

With full Armenian coverage across Serif, Sans, and Mono, all major weights and shapes, and dedicated optical sizes, New Computer Modern offers one of the most complete and typographically coherent Armenian implementations available in a general-purpose academic type family.

16 Georgian (Mkhedruli) script support

NewCM includes full support for Georgian Mkhedruli (Unicode U+10D0–U+10FF) across the Roman, Sans, and Mono families, in Regular, Bold, Oblique/Italic, and Bold Oblique styles. The design aims at seamless integration with Computer Modern–derived Latin, Greek, and mathematical typography, with particular emphasis on paragraph color, rhythm, and mixed-script stability.

16.1 Design principles

The Georgian design in NewCM is based exclusively on **Mkhedruli**, the contemporary Georgian script used in all modern contexts. No uppercase/lowercase distinction exists in Mkhedruli, and the font follows Unicode conventions without introducing artificial capitalization.

Key design goals were:

Text color compatibility with NewCM Latin and Greek at typical academic sizes (9–12 pt).

Controlled joins and counters, especially in Bold and Mono styles, to avoid local darkening.

Script neutrality: Georgian text should neither dominate nor recede in mixed-script paragraphs.

Mechanical italics (obliques) rather than cursive reinterpretations, in line with NewCM practice.

The Sans and Mono variants are true companions of their Latin counterparts, not condensed or simplified derivatives. In Mono, spacing discipline was prioritized over calligraphic expressiveness to ensure usability in code listings and technical material.

16.2 Italics and emphasis

In Georgian typography, emphasis in academic and editorial contexts is commonly expressed through **italic (oblique) forms**, similarly to Latin practice. NewCM therefore provides full Georgian support in oblique styles for Roman, Sans, and Mono families. Italic Georgian is constructed as a consistent slant of the upright forms, ensuring harmony with Latin and Greek italics in theorem statements and emphasized text.

16.3 Example: the Pythagorean theorem

Below is a short Georgian text introducing the Pythagorean theorem, followed by the theorem statement itself, set in italics to demonstrate typical mathematical typography.

ჰითაგორას თეორემა არის ევკლიდური გეომეტრიის ერთ-ერთი ყველაზე ცნობილი და ფუნდამენტური შედეგი. იგი ამყარებს კავშირს მართკუთხა სამკუთხედის გვერდებს შორის და ფართოდ გამოიყენება მათემატიკაში, ფიზიკაში და ინჟინერიაში.

თეორემა (Πυθαγόρας) მართკუთხა სამკუთხედში ჰიპოტენუზის კვადრატი ტოლია კატეტების კვადრატების ჯამისა. $c^2 = a^2 + b^2$.

This example illustrates the behavior of Georgian text in italic theorem statements, its interaction with Latin mathematical symbols, and the balance between text and displayed mathematics.

16.4 Scope

NewCM Georgian covers:

- Unicode block U+10D0–U+10FF (Georgian)
- Roman, Sans, and Mono families
- Regular, Bold, Oblique/Italic, and Bold Oblique styles

The Georgian support is intended for *academic, scientific, and technical documents*, where Georgian text appears alongside Latin, Greek, and mathematics in a unified typographic system.

17 Medieval Latin and Uncial Greek

The family includes new shapes for medieval Latin and uncial Greek. The `fontsetup` package provides `\textuncial{<text>}` and `{\uncial text}` to use this shape. Let us write a sentence in this shape:

Τῆς ΠΥΘΑΓΟΡΕΑΝ ΤῆΟΡΕΜ ΙΣ ΟΝΕ ΟΥ Τῆ ΜΟΣΤ ΙΜΠΟΡΤΑΝΤ
 ΤῆΟΡΕΜΣ ΙΝ ΜΑΤῆΜΑΤΙΣ.
 Το ΠῤῆΑΓΟΡΕΙΟ ΘΕΩΡῤῤΑ ΕΙΝΑΙ ΑΠΟ ΤΑ ΣΗΜΑΝΤΙΚΟΤΕΡΑ ΘΕ-
 ΩΡῤῤΑΤΑ ΤΩΝ ΜΑΘΗΜΑΤΙΚΩΝ.

Medieval Latin and Uncial Greek fonts will give a lot of “missing slot” warnings if the microtype package is loaded. To suppress these warnings use

```
\DeclareMicrotypeAlias{NewCMUncial10-Book.otf}{TU-empty}
```

and similarly for the other NewCMUncial fonts.

18 Braille

Braille, both 6dot (uni2801–uni283F) as well as 8dot (uni2840–uni28FF) patterns are included in two versions. The Regular font provides the characters for sighted persons (such as teachers) so they can easily see which dots are on and which off. The Sans font contains the true Braille characters. I decided to have the sighted version in the Regular font since a blind person does not need the real Braille pattern, as those are produced by embossers. The Braille patterns here are meant as fonts to typeset text mainly for sighted persons.

	6dot	8dot
Regular version	⠠⠠⠠⠠⠠⠠	⠠⠠⠠⠠⠠⠠⠠⠠
Sans version	⠠⠠⠠⠠⠠⠠	⠠⠠⠠⠠⠠⠠

19 IPA symbols

IPA symbols are included and following a suggestion of Huanyu Liu the kerning found in `tipa` package has been added here and further improved. Moreover the letters eth, eng, beta, theta and chi exists in IPA-style in the fonts and are accessible in the ss05 lookup table since they are in a different design from the Latin and Greek letters. You can access this lookup table using the `\textipa` command of the `fontsetup` package.

	Non-IPA	IPA
Regular	ð ɳ β θ χ	ð ɳ β θ χ
Sans	ð ɳ β θ χ	ð ɳ β θ χ

I am grateful to निरंजन (Niranjan) for suggesting and testing all the IPA improvements that follow:

The joining of two characters such as ts , d_3 , kp , tf etc is also supported. The low tie is the character U+035C and the upper tie is U+0361. The ts is produced by typing the sequence t then the low tie and then s. Similarly kp is produced by typing the sequence k then the upper tie and then p. For tf and d_3 the fonts have a contextual chaining substitution table that uses a tie which prints lower so it does not touch f and 3. In cases that one wants to show the tie as xyz then one needs to enable the cv5 character variant since the tie characters (U+035C and U+0361) are marks and not base glyphs. One can do that with a command such as

```
\newfontfamily{\showtie}[CharacterVariant=5]{NewCM10-Book.otf}
```

There is also an older practice that such sequences are joint into a ligature. This is not the modern way of writing but it seems than many people in the linguistics still prefer it. The fonts support this, if one enables the `lipa` table (local ipa) of the fonts. This can be done by adding the `RawFeature+=lipa` to the font specs when loading it and it is done automatically with the commands `\textoldipa{arg}` and `\oldipatext arg` of the `fontsetup` package.

For example, `\textoldipa{ts, tf}` produces ts , tf .

All other symbols of the `tipa` package are supported. Some examples are:

$\text{p}^{\text{̣}}$, ' , t , l , d , j , s , e , k , m , t , t , etc.

It is worth noting that all of the above is also available in the Mono family:

ts tf ts $\text{p}^{\text{̣}}$ ' t l d j s e k m t t

Another issue is that IPA used to use the Italic alternation of “g” even in the upright design. This restriction was lifted in 1949 as can be seen on page 3 in [IPAreview].

Even though the normal upright shape is accepted as an alternation, there are linguists who prefer the older (Italic) shape “g” and hence we are making it the default in the stylistic set dedicated for IPA (ss05).

Another such case is with the “a” shape. It has a different visual form in Italics “a” which resembles to a distinct symbol of IPA leading to a wrong transcription. In order to avoid this confusion, we have used the slanted version “a” of “a” in ss05. The other vowel symbol (ie, a (U+0251)) is also available in the fonts.

Next we give a real example of using IPA symbols. The quote below is from *A course in phonetics*, Ladefoged, Peter and Johnson, Keith, (2014), Cengage learning, pp. 285–286, and the `StylisticSet=05` has been enabled:

Other examples of this phenomenon have also been mentioned. We saw that before [ɹ], English does not distinguish between tense and lax vowels. Consequently, the vowel that occurs in, for example, sing has a quality between that of [i] and [ɪ]. Similarly, there is no distinction between tense and lax vowels before [ɹ]. The vowel in here in most forms of American English is also intermediate between [i] and [ɪ]. The principle of maximum perceptual separation also accounts for some of the differences between languages. French has two high rounded vowels, [u] as in *tout* [tu] “all,” and [y] as in *tu* [ty] “you.”

20 All Unicode combining marks

As NewComputerModern has top priority the support of Academic work, all combining marks are supported in all fonts. Please notice that not all slots in these blocks are filled in with glyphs by Unicode standard. Some slots are empty and reserved for the future. All glyphs described by Unicode are in NewCM.

These are the Unicode blocks:

- Combining Diacritical Marks U+0300–U+036F
- Combining Diacritical Marks Extended U+1AB0–U+1AFF
- Combining Diacritical Marks Supplement U+1DC0–U+1DFF
- Combining Diacritical Marks for Symbols U+20D0–U+20FF
- Combining Half Marks U+FE20–U+FE2F

All blocks support stacking. Examples, other than the main diacritical block U+0300–U+036F, follow in the next subsections.

20.1 Extended and Supplement Combining Diacritical Marks

Both blocks are supported:

Extended U+1AB0–U+1AFF. These are used to represent fine-grained phonetic, prosodic, and tone features, primarily in linguistics, dialectology, and musicology. They are combining characters, so they attach to a base letter. Example:

mǎ mà mǎ mǎ mǎ mǎ
mǎ mà mǎ

Used for advanced tone marks and diacritic combinations in phonetics and linguistics (eg, tonal contour notation in African and East Asian languages). For example, Yoruba, Zulu, Mandarin, or reconstructed Proto-Bantu tone patterns.

contractions, sacred names, and numeric values. The FE2E/FE2F pair lets a single titlo be encoded as two combining halves that attach to the leftmost and rightmost letters of the span and visually join into one continuous overline. However, when more than two characters are used, or characters with different heights, there is no provision by Unicode how these characters should join. Necessarily the titlo will break. However, a new command `\titlo` has been introduced in `fontsetup` package (that applies to `newcomputermodern.sty` as well) that produces proper titlos on more than two letters and of unequal heights. I thank David Carlisle for his help with this.

Use case: when an editor wants a single titlo covering two or more adjacent Cyrillic letters, they insert the left-half after the first letter and the right-half after the last letter of the span.

- Two-letter titlo; Characters: U+043A U+FE2E U+043D U+FE2F: $\overline{\text{KH}}$
- Three-letter titlo with different heights; Characters: U+043A U+FE2E U+0430 U+0431 U+FE2F: $\overline{\text{Ka6}}$
- Titlo used with a numeral (titlo often marks numeric use of letters); Characters: U+0432 U+FE2E U+0438 U+FE2F: $\overline{\text{BII}}$
- Corrected output using the `\titlo` command with `\titlo{\Ka6}`: $\overline{\text{Ka6}}$.

21 Editorial Marks

NewComputerModern supports a range of combining editorial marks intended for use in scholarly text, critical editions, manuscript transcription, and linguistic annotation. These marks function as textual modifiers applied to arbitrary base characters and are distinct from precomposed mathematical symbols.

21.1 Scope and Intended Use

Editorial marks are used to indicate:

1. manuscript witnesses and scribal hands in critical apparatus,
2. editorial interventions and conjectural readings,
3. layered annotation in palaeography and codicology,
4. feature marking and classification in linguistics.

In these contexts, the mark conveys editorial status, not mathematical meaning.

21.2 Combining Enclosing Marks

The font provides full support for combining enclosing marks, including:

U+20DD COMBINING ENCLOSING CIRCLE

U+20DE COMBINING ENCLOSING SQUARE

U+20DF COMBINING ENCLOSING DIAMOND

These characters attach to a preceding base glyph. The enclosing shape adapts to the base character’s size and style (e.g. italic, small caps, Greek, Cyrillic), enabling consistent use in running text.

Example (editorial sigla):

Ⓐ Ⓟ Ⓕ Ⓚ Ⓢ Ⓝ

(the last example is in small caps). Editorial marks are positioned using dedicated anchors in the text font. The enclosing shapes are centered optically relative to the base glyph rather than aligned to the mathematical axis. Letters with ascenders or descenders are adjusted to preserve visual balance inside the enclosure.

Editorial combining marks should not be confused with precomposed circled letters used in mathematical notation (eg automata theory). Mathematical circled symbols are atomic glyphs provided by math fonts and are accessed via Unicode or math alphabets. Combining enclosing marks in NewComputerModern are intended exclusively for textual and editorial use.

22 Phonetic Extensions and Supplement

The block U+1D00–U+1D7F contains letters and small capitals used in the IPA and related phonetic alphabets, and the block U+1D80–U+1DBF adds further small capitals and modifier letters for phonetic and prosodic notation. The small letters here are *not* to be confused with typographical small capitals. The designs here are at the x-height of the font which is smaller than small capitals. Small capitals should still be called with commands such as `\textsc` or similar. Examples:

- (i) [b̥aɾɪ] [b̥aɾɪ] [b̥aɾɪ] [b̥aɾɪ]
- (ii) [m̥aɪ] [m̥aɪ] [m̥aɪ] [m̥aɪ] meaning “man” pronounced with a bilabial trill
- (iii) [ɖ̥aɪ] [ɖ̥aɪ] [ɖ̥aɪ] [ɖ̥aɪ] could represent a voiced implosive /ɖaɪ/ in a phonetic study.
- (iv) tʰa kʷa n̥i l̥a tʰa kʷa n̥i l̥a tʰa kʷa n̥i l̥a aspirated t, labialized k, palatalized n, velarized l
- (v) tʰ d̥ n̥ tʰ d̥ n̥ tʰ d̥ n̥ detailed articulatory descriptions and experimental phonetics.

gives

100 ₹

100 ₹

24 Bold Sans

lm fonts and cmu fonts do not contain a properly made BoldSans. Their BoldSans is a stroke-extension of the Sans with rounded corners. NewCM fixes that and provides a true BoldSans:

LM	NewCM
XΞ	XΞЯДЛ

25 LX fonts for presentations

This is a very interesting variation especially designed for beamer projections. The original design belongs to Claudio Beccari who suggested me to incorporate this into NewCM. It is compatible with the Knuthian design and looks very clean on projections. The fonts have been reworked to stabilize the color and provide Oblique, Bold and BoldOblique.

Here is a sample with Pythagorean theorem.

The Pythagorean Theorem is one of the most important theorems in Mathematics.

Το Πυθαγόρειο Θεώρημα είναι από τα σημαντικότερα θεωρήματα των Μαθηματικών.

Theorem (Πυθαγόρειον) In a right angle triangle with sides α , β , γ with α being the hypotenuse, we have

$$\alpha^2 = \beta^2 + \gamma^2.$$















Ἐν τοῖς ὀρθογωνίοις τριγώνοις τὸ ἀπὸ τῆς τὴν ὀρθὴν γωνίαν ὑποτείνουσας πλευρᾶς τετράγωνον ἴσον ἐστὶ τοῖς ἀπὸ τῶν τὴν ὀρθὴν γωνίαν περιεχουσῶν πλευρῶν τετραγώνοις.

You can select this variation when in Sans family by enabling StylisticSet 8 for both text and Mathematics, although one must make adjustments of the height of letters as their x-height is bigger. It is simpler to load the `fontsetup` package with option `lx` which will take care of proper scaling factors for text and Mathematics.

Please notice that you have to use `lualatex` if you need vertical constructions like the extensible parenthesis, extensible braces etc. Unfortunately `xelatex` fails to use the font properly for these glyphs.

26 Food Allergies

Food allergy symbols have long been proposed to be accepted to Unicode standard but there has not been any progress up to now. The Sans 10 Regular and Book include standard allergy symbols in the Private Use Area. Each glyph is named after what it represents. For example, the soya symbol is named “soya” so copying the symbol from a pdf file and pasting elsewhere you will get its name, that is “soya”. The symbols are in positions U+E033 to U+E040 and can be accessed using the commands of the next table.

<code>\char"E033</code>	 CRUSTACEANS	<code>\char"E034</code>	 EGGS
<code>\char"E035</code>	 GLUTEN	<code>\char"E036</code>	 FISH
<code>\char"E037</code>	 LUPIN	<code>\char"E038</code>	 MILK
<code>\char"E039</code>	 MOLLUSCS	<code>\char"E03A</code>	 MUSTARD
<code>\char"E03B</code>	 PEANUT	<code>\char"E03C</code>	 SESAME
<code>\char"E03D</code>	 SOYA	<code>\char"E03E</code>	 TREENUTS
<code>\char"E03F</code>	 CELERY	<code>\char"E040</code>	 SO ₂

27 Unicode Math coverage and options

NewCM provides full Unicode math support, that is all Mathematics Unicode Slots presented in <http://www.unicode.org/charts/> in the Math weights, Regular, Book and Bold. These blocks are:

Mathematical Symbols

- Arrows (U+2190–U+21FF)
- Supplemental Arrows-A (U+27F0–U+27FF)
- Supplemental Arrows-B (U+2900–U+297F)
- Supplemental Arrows-C (U+1F800–U+1F8FF)
- Additional Arrows (U+2B00–U+2BFF)
- Miscellaneous Symbols and Arrows (U+2B00–U+2BFF)

Mathematical Alphanumeric Symbols

- (U+1D400–U+1D7FF)
- Arabic Mathematical Alphabetic Symbols (U+1EE00–U+1EEFF)
- Letterlike Symbols (U+2100–U+214F)

Mathematical Operators

- (U+2200–U+22FF)
- Basic operators: Plus, Factorial (U+0000–U+007F)
- Division, Multiplication (U+0080–U+00FF)
- Supplemental Mathematical Operators (U+2A00–U+2AFF)
- Miscellaneous Mathematical Symbols-A (U+27C0–U+27EF)
- Miscellaneous Mathematical Symbols-B (U+2980–U+29FF)
- Floors and Ceilings (U+2308–U+230B)
- Invisible Operators (U+2061–U+2064)

Geometric Shapes (U+25A0–U+25FF)

- Additional Shapes (U+2B00–U+2BFF)
- Box Drawing (U+2500–U+257F)
- Block Elements (U+2580–U+259F)
- Geometric Shapes Extended (U+1F780–U+1F7FF)

Unfortunately, the `unicode-math` package does not currently provide commands for the hundreds of extra glyphs that have been added in order to fully cover the above Unicode ranges. The user can consult the Unicode charts at the above link and access the required glyph with `\char"#` where `#` is the Unicode number of the slot the glyph belongs to.



For example, `\char"2BDA` will give the Hygeia symbol (U+2BDA) the Rod of Asclepius as shown above (grayed and scaled $\times 8$). The glyph that appeared in TUGboat (see [AT]), being more realistic will be moved to a new font in the future with ornaments.

27.1 Bold Math

A complete math font, such as NewCM, contains all alphabetic characters in bold too. These characters are typically accessed using the `\mathbf` command. However, this is not true for bold versions of symbols. This creates difficulties when a user has some math in chapter or section titles, or when a user wants to create a poster with colored background. In the later case the Regular and Book weights look too light (especially with dark backgrounds) and one is in need of a real Bold Math font, that has everything in Bold.

NewCM, starting from version 6.0.0 provides an independent Bold Math font for the first time for a Computer Modern font family. Let us compare:

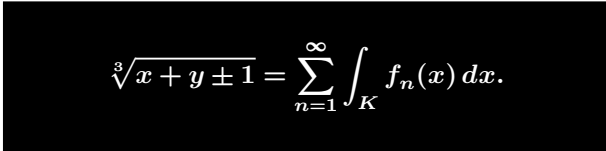
Book inline: $\sqrt[3]{x+y\pm 1} = \sum_{n=1}^{\infty} \int_K f_n(x) dx$ and the same in display

$$\sqrt[3]{x+y\pm 1} = \sum_{n=1}^{\infty} \int_K f_n(x) dx.$$

Bold inline: $\sqrt[3]{x+y\pm 1} = \sum_{n=1}^{\infty} \int_K f_n(x) dx$ and the same in display

$$\sqrt[3]{x+y\pm 1} = \sum_{n=1}^{\infty} \int_K f_n(x) dx.$$

Bold Math with colors inverted:



$$\sqrt[3]{x+y\pm 1} = \sum_{n=1}^{\infty} \int_K f_n(x) dx.$$

In order to use the Bold font for chapter and section titles, when using the `default` or `olddefault` options of the `fontsetup` package, change the math version to bold with `\mathversion{bold}` *before* the commands for chapter and section and switch back to normal with `\mathversion{normal}` *afterwards*.

The `unicode-math` package, according to its documentation has still some troubles with “versions” and the `range` options. These seem to affect at least the calligraphic and script math alphabets. In normal version for example the commands `\symcal` and `\symscr` work as expected, but when one switches to the bold version the `\symscr` fails. In this case one can use `\sympfscr`: \mathcal{A} \mathscr{A} (which was `\symcal A$ $ \sympfscr A$`).

27.2 Sans Serif Math

As of version 7.0.0, the family includes a full-featured Math Sans font. Up to now such a font did not exist in the CM family (although partial solutions

existed), and it posed a serious problem for scientific writing especially in the preparation of presentations. The font supplied with NewCM covers all Unicode math slots but it also provides some new features. The lowercase Latin alphabet has been completely re-worked so that it really looks as it should when used for Math variables. The letters are

|a|b|c|d|e|f|g|h|i|j|k|l|m|n|o|p|q|r|s|t|u|v|w|x|y|z|

In Large

|a|b|c|d|e|f|g|h|i|j|k|l|m|n|o|p|q|r|s|t|u|v|w|x|y|z|

Moreover the calligraphic capitals have been re-worked to match better with the sans serif style (see subsection 27.4 for how to select them)

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

The same holds true for the `\mathbb`/`\symsf` capital letters. They have been adjusted to match the Sans design, both in style and weight:

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

It is also worth noticing that the SansSerifed and the Serifed letters have swapped slots in the font. So if using the MathSans font and you write for example `\symsf{ABCDabcd}` you will get the Serifed version(!): ABCDabcd. This choice facilitates converting a document with Serifed fonts using Sans for emphasis or differentiation to keep these characteristics when changed to Sans.

One can see the Sans Serif Math font in action in the provided file `testmath-newcm.pdf` which comes from the \mathcal{MS} - \LaTeX bundle (here the logo is written in NewCMSans).

27.3 Optical sizes for more glyphs

So far the fonts provided optical sizes for 1st and 2nd order exponents for letters. This was not true though for binary operators and some symbols commonly used in mathematics. For example, the `\perp` symbol (\perp) often appears in 2nd order exponents and then it appeared very thin. Some printers could even hardly print its thin lines. Now, such glyphs plus several binary operators, such as $+$, $-$, \pm , $\dot{+}$ etc are now provided in optical sizes, so that expressions such as

$$+^{++} \quad \perp^{\perp\perp} \quad *^{**} \quad \frac{|P_{F^\perp}(K)|}{e^{(x+y)^*}}$$

appear on screen and print properly on printers. Zoom or print and compare with `latinmodern-math` font:

$$+^{++} \quad \perp^{\perp\perp} \quad *^{**} \quad \frac{|P_{F^\perp}(K)|}{e^{(x+y)^*}}$$

27.4 Math Script

Calligraphic letters are accessed as usual with `\mathcal` or `\symcal`, producing

ABCDEFGHIJKLMNOPQRSTUVWXYZ,

which is a re-traced and glyph by glyph adjustment of the original Knuth design. Prior to version 8.0.0 of the fonts the `\mathcal` was providing the Euler chancery style calligraphics inherited from `lmodern-math`. This version is now saved in `StylisticSet 6`.

However, mathematicians often need a second level of “scriptness”. The fonts provide an alternative calligraphic, a script design at `StylisticSet 1`. For this to work one has to re-set the math font using `\setmathfont [range={\mathscr,\mathbfscr},StylisticSet=1]{NewCMMath-Book.otf}`

(or the Regular version). The `fontsetup` package from version 2.4.0 and on, supports automatically all three variants (thanks to David Carlisle, see [web2]) introducing a new command, namely `\mathchancery/\symchancery` to access the chancery version.

The next code

```

 $\mathcal{ABCDEFGHIJKLMNOPQRSTUVWXYZ}$ 
 $\mathchancery{ABCDEFGHIJKLMNOPQRSTUVWXYZ}$ 
 $\mathscr{ABCDEFGHIJKLMNOPQRSTUVWXYZ}$ 
 $\mscra\mscrb\mscrc\mscrd\mscre\mscrf\mscrg\mscrh\mscri\mscrj$ 
 $\mscrk\mscrl\mscrm\mscrn\mscro\mscrp\mscrq\mscrr\mscrs\mscrt$ 
 $\mscru\mscrv\mscrw\mscrx\mscry\mscrz$ 

```

produces

ABCDEFGHIJKLMNOPQRSTUVWXYZ
ABCDEFGHIJKLMNOPQRSTUVWXYZ
ABCDEFGHIJKLMNOPQRSTUVWXYZ
abcdefghijklmnopqrstuvwxyz

27.5 Math Kerning

Math kerning has been added to all NewCM Math fonts. This feature greatly improves Math typesetting, especially for the calligraphic letters but for regular letters as well, such as, Y or Γ .

Latin Modern	NewCM-Regular
$F_x Y_x \mathcal{N}_x \mathcal{J}_x \mathcal{G}_x \Gamma_x$	$F_x Y_x \mathcal{N}_x \mathcal{J}_x \mathcal{G}_x \Gamma_x$

Regular kerning in math mode works with Lua^AT_EX but not with X_YL^AT_EX. The next table shows the effect of kerning in math mode with calligraphic capital letters and capital Latin italic letters with period and comma (look closely). This table is compiled with Lua^AT_EX.

Before	Current (with Lua)
$\mathcal{H}.\mathcal{H},\mathcal{H}, H.H.H,H,H,H,$	$\mathcal{H}.\mathcal{H},\mathcal{H}, H.H.H,H,H,H,$

Regular kerning in math mode does not work with X_YL^AT_EX as said above. With Lua^AT_EX we notice that it works with calligraphic capitals but not with Latin italic capitals. The reason looks to be the fact that these letters have italic correction enabled in the fonts (as they should). Italic correction breaks the application of the kern. Now if italic correction is removed, by say $\mathcal{H}\backslash/. \$$, then the engine does not see the characters \mathcal{H} and period as consecutive characters to apply the kern, and the kern is lost.

The code of the table above is as follows:

```
\begin{tabular}{c|c}
Before & Current (with Lua)\\ \hline
{\setmathfont{latinmodern-math.otf}}\Large
${\symcal H.}^{\{ {\symcal H,}^{\{ {\symcal H,}^{\{ }$
${\mathcal H}\backslash/.}^{\{ {\mathcal H.}^{\{ {\mathcal H}\backslash/,}^{\{ {\mathcal H}\backslash/,}^{\{ }$
& {\setmathfont{NewCMMath-Regular.otf}}\Large
${\symcal H.}^{\{ {\symcal H,}^{\{ {\symcal H,}^{\{ }$
${\mathcal H}\backslash/.}^{\{ {\mathcal H.}^{\{ {\mathcal H}\backslash/,}^{\{ {\mathcal H}\backslash/,}^{\{ }$
\end{tabular}
```

We also notice that Lua^AT_EX has a parameter set by the command `\mathitalicsmode`

with default value 0. If set to 2 with `\mathitalicsmode=2` then Lua^AT_EX uses regular letter kerning in math mode but breaks other things like the end-points of integrals (see [web1]). We conclude this subsection realizing that there is no single engine (among X_YL^AT_EX and Lua^AT_EX) that supports all features.

27.6 Blackboard Bold

The fonts contain as default the AMS blackboard bold. These are:

ABCDEFGHIJKLMNOPQRSTUVWXYZ
 abcdefghijklmnopqrstuvwxyz
 0123456789 π δ Γ Π Σ D d e i j

They also contain a blackboard bold that matches the design of Computer Modern but respecting the fact that most users have been used for a long period to the AMS bb design. Compare the default

$$\mathbb{R} \in \mathbb{R} \quad \mathbb{Q} \in \mathbb{Q}$$

with the new design

$$\mathbb{R} \in \mathbb{R} \quad \mathbb{Q} \in \mathbb{Q}$$

To access this design one needs to load the math font enabling the `ss03` stylistic set using for example
`\setmathfont[StylisticSet=3]{NewCMMath-Book.otf}` Then the above blackboard bold design changes to

ABCDEFGHIJKLMNOPQRSTUVWXYZ

abcdefghijklmnopqrstuvwxyz

0123456789 π γ Γ Π Σ *Ddeifj*

If using the latest `fontsetup` then you can choose the NewCM blackboard bold with the option `newcmbb`.

27.6.1 Small Caps for Blackboard Bold

A user might need to use Blackboard Bold in Small Caps. For example, this may arise when running heads are in small caps and contain a Blackboard Bold symbol. Consider for example the case when a chapter is named “The Lebesgue measure on \mathbb{R} ”. Then the running head in small caps will look awkward:

THE LEBESGUE MEASURE ON \mathbb{R}

In such cases, a small caps version of the number sets is needed. NewCM provides the needed glyphs in `ss05`. So setting
`\setmathfont[StylisticSet=5]{NewCMMath-Book.otf}`
the above running head becomes:

THE LEBESGUE MEASURE ON \mathbb{R}

These glyphs are provided in the CM style (instead of the AMS bb) by enabling `ss03` too. So the command
`\setmathfont[StylisticSet=3,StylisticSet=5]{NewCMMath-Book.otf}`
will produce:

THE LEBESGUE MEASURE ON \mathbb{R}

The same is true for the Regular and Bold Math fonts.

27.7 Upright and extensible integrals

The Math fonts (both Regular and Book weights) include upright integrals in the ss02 StylisticSet. Use with

```
\setmathfont[StylisticSet=2]{NewCMMath-Book.otf}
or
\setmathfont[StylisticSet=2]{NewCMMath-Regular.otf}
```

or use the `upint` option of the `fontsetup` package with

```
\usepackage[upint,default]{fontsetup}
```

for the Book weight, or

```
\usepackage[upint,olddefault]{fontsetup}
```

for the regular weight.

Moreover, extensible integrals are supported by the fonts but *NOT* by the Unicode TeX engines. The following code is a trick so that extensible integrals can be constructed using Lua^ATeX. The result is shown at the end of the article. What the code below does, is that it defines the slot `uni222B` (integral) as a delimiter. And then this is extended as a delimiter with the mechanism that the font provides.

```
\documentclass{article}
\usepackage[default]{fontsetup}
\begin{document}
$
\Uleft\Udelimiter 0 0 "222B
\begin{pmatrix}
1\\2\\3\\4\\5\\6\\7\\8\\9
\end{pmatrix}
\Uright.
$
\end{document}
```

$$\int \begin{pmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \end{pmatrix}$$

27.8 Additional and alternative characters in Math

27.8.1 Alternative symbol for the empty set

There is a mismatch between the TeX world for the empty set and the Unicode standard. For TeX the default empty set is \emptyset , that is a slashed zero. Unicode suggests that the proper character is \varnothing , and suggests the switch to the slashed zero if \varnothing is followed by the Variation Selector U+FE00. However, this method is not yet supported by Unicode TeX engines. Since the fonts must comply to the Unicode standard the math fonts follow the Unicode suggestion. To make things work as expected in TeX, the user must load the Math fonts with `CharacterVariant=2`. For example, with

```
\setmathfont[CharacterVariant=2]{NewCMMath-Regular.otf}
```


This will map `\emptyset` to \emptyset . If `CharacterVariant=2` is not loaded the `\emptyset` will give the proper Unicode character, that is \varnothing .

If users want to have access to both symbols, \emptyset with `\emptyset` and \varnothing with `\varnothing` in the same document then this can be done with the next four lines.

```
\setmathfont[CharacterVariant=2]{NewCMMath-Regular.otf}
\AtBeginDocument{%
\renewcommand{\emptyset}{\char"2205}%
\renewcommand{\varnothing}{\char"2300}}
```

The `fontsetup` package defaults to the \TeX -default empty set (\emptyset) and provides the option ‘`varnothing`’ to easily switch to the alternative character.

27.8.2 Other alternative symbols

The Unicode standard defines more alternative symbols as for the `emptyset` above using variation selector characters U+FE00 and U+FE01. NewCM Math fonts support all variations suggested by Unicode (except U+FF10 as this is to be used with CJK fonts, not with math). The Unicode method is that we get a variation of the original glyph if it is followed by U+FE00 or U+FE01. However, this method is not supported yet by the Unicode \TeX engines. One block of such characters are the script capitals, chancery or roundhand. \TeX access to those is described before, in Section 27.4. The rest are the following:

$\emptyset, \cap, \cup, \subseteq, \supseteq, \subset, \supset, \sqsubset, \sqsupset, \sqcap, \sqcup, \oplus, \otimes, \ominus, \lesssim, \gtrsim, \mathbb{O}, -, \lhd, \rhd, \lessgtr, \gtrless, \leq, \geq, \leqslant, \geqslant$

As we did for the empty set, we have included these variations in `cv01`. When enabled it will produce the varied characters:

$\emptyset, \cap, \cup, \subseteq, \supseteq, \subset, \supset, \sqsubset, \sqsupset, \sqcap, \sqcup, \oplus, \otimes, \ominus, \lesssim, \gtrsim, \mathbb{O}, \mathbb{J}, \mathbb{L}, \lessgtr, \gtrless, \leq, \geq, \leqslant, \geqslant$

27.8.3 Additional symbols

It also provides four more arrows that correspond to the commands

```
\nrightrightarrows (\rightharpoonright) \nleftleftarrows (\leftharpoonleft)
\twoheadhookrightarrow (\twoheadrightarrow) \twoheadhookleftarrow (\twoheadleftarrow)
```

and

```
\increasingconcave (\nearrow) \increasingconvex (\nearrow)
\decreasingconcave (\searrow) \decreasingconvex (\searrow)
```

and supported by the `default`, `olddefault` and `sansdefault` options of the `fontsetup` package. These symbols are not in the Unicode Standard and so they are added in the Private Area of the fonts. The later four arrows are needed when making tables describing the behavior of functions when they are studied in Calculus.

27.8.4 Additional operators

An operator for convolution of functions seems to have long been forgotten from both Unicode and the \TeX world. The `default` and `olddefault` options of the `fontsetup` package define a new command `\convolution` which behaves just like the `\sum` and `\int` operators. The convolution of N functions in inline mode: $\bigstar_{n=1}^N f_n$ and the same in display mode:

$$\bigstar_{n=1}^N f_n.$$

27.8.5 “Smoother” changing radicals

One more radical size has been added that improves the way radical sizes change. Compare the previous state

$$\sqrt{q} \quad \sqrt{q_1^2} \quad \sqrt{q_1^{2^3}} \quad \sqrt{\frac{1}{2}}$$

with the new one

$$\sqrt{q} \quad \sqrt{q_1^2} \quad \sqrt{q_1^{2^3}} \quad \sqrt{\frac{1}{2}}.$$

27.9 Notes on Mathematics

Extensible tildes and hats produce different results with \XeTeX and \LuaTeX because these unicode engines treat differently the width of characters. In particular, \XeTeX handles italic correction as part of the character width but \LuaTeX does this only if a character follows. This affects how extensible accents like `\widehat` or `\widetilde` select the proper size. With \XeTeX , `\widetilde{Y}` `\widetilde{X}` will give the expected result $\widetilde{Y}\widetilde{X}$; but with \LuaTeX the letter Y is narrower (and its italic correction is not taken into account if it stands alone) and gets the plain tilde size, as in \tilde{Y} . To bypass this with \LuaTeX one has to use a zero width character after a letter such as Y so that \LuaTeX takes into account the italic correction of Y . For example with

`\widetilde{Y\Uchar"200D}`

the result will be correct (U+200D is the “Zero Width Joiner” in Unicode).

I thank Ulrike Fischer for this solution.

28 The Medieval Latin and Uncial Greek glyph complement

Table 1: NewCMUncial10-Book.otf (features: ExternalLocation=../otf/)

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
Basic Latin																
U+0020–002F		!	"	#	\$	%	&	'	()	*	+	,	-	.	-
U+0030–003F	o	1	2	3	4	5	6	7	8	9	:	;	<	=	>	-
U+0040–004F	Q	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
U+0050–005F	P	Q	R	S	T	U	V	W	X	Y	Z	-	-	-	-	-
U+0060–006F	-	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
U+0070–007F	p	q	r	s	t	u	v	w	x	y	z	-	-	-	-	-
Latin-1 Supplement																
U+00A0–00AF	-	-	-	-	-	-	-	-	ˆ	-	-	-	-	-	-	-
U+00B0–00BF	-	-	-	-	-	-	-	˙	-	-	-	-	-	-	-	-
Greek and Coptic																
U+0370–037F	-	-	-	-	´	,	-	-	-	-	-	-	-	-	;	-
U+0380–038F	-	-	-	-	ˆ	ˆ	λ	·	€	H	I	-	O	-	Υ	Ω
U+0390–039F	ı	λ	β	Γ	Δ	€	Z	H	Θ	I	K	Λ	M	N	Ξ	O
U+03A0–03AF	Π	P	-	C	T	Υ	Φ	X	Ψ	Ω	İ	Ŧ	λ	ε	Η	Ι
U+03B0–03BF	Ŧ	λ	β	Γ	Δ	ε	ζ	H	θ	I	K	λ	μ	N	ξ	O
U+03C0–03CF	Π	P	C	C	T	Υ	Φ	X	Ψ	ω	İ	Ŧ	O	Υ	ω	-
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F

Total number of glyphs shown from NewCMUncial10-Book.otf: 160

29 The Aegean Numbers glyph complement

<code>\aegeanseparator</code>	∣	<code>\aegeaneighthundred</code>	⋈
<code>\aegeanseparatordot</code>	·	<code>\aegeanninehundred</code>	⋈
<code>\aegeancheckmark</code>	×	<code>\aegeanonethousand</code>	÷
<code>\aegeanone</code>	∣	<code>\aegeantwothousand</code>	÷÷
<code>\aegeantwo</code>	∣∣	<code>\aegeanthreethousand</code>	÷÷
<code>\aegeanthree</code>	∣∣∣	<code>\aegeanfourthousand</code>	÷÷
<code>\aegeanfour</code>	∣∣∣	<code>\aegeanfivethousand</code>	÷÷
<code>\aegeanfive</code>	∣∣∣	<code>\aegeansixthousand</code>	÷÷
<code>\aegeansix</code>	∣∣∣	<code>\aegeanseventhousand</code>	÷÷
<code>\aegeanseven</code>	∣∣∣	<code>\aegeaneightthousand</code>	÷÷
<code>\aegeaneight</code>	∣∣∣	<code>\aegeanninethousand</code>	÷÷
<code>\aegeanine</code>	∣∣∣	<code>\aegeantenthousand</code>	+
<code>\aegeanten</code>	-	<code>\aegeantwentythousand</code>	÷÷
<code>\aegeantwenty</code>	=	<code>\aegeanthirtythousand</code>	÷÷
<code>\aegeanthirty</code>	≡	<code>\aegeanfourtythousand</code>	÷÷
<code>\aegeanfourty</code>	==	<code>\aegeanfiftythousand</code>	÷÷
<code>\aegeanfifty</code>		<code>\aegeansixtythousand</code>	÷÷
<code>\aegeansixty</code>	≡≡	<code>\aegeanseventythousand</code>	÷÷
<code>\aegeanseventy</code>	≡≡	<code>\aegeaneightythousand</code>	÷÷
<code>\aegeaneighty</code>	≡≡	<code>\aegeanninetythousand</code>	÷÷
<code>\aegeanninety</code>	≡≡	<code>\aegeanweightbaseunit</code>	⌘
<code>\aegeanonehundred</code>	°	<code>\aegeanweightfirstsubunit</code>	℥
<code>\aegeantwohundred</code>	°°	<code>\aegeanweightsecondsubunit</code>	℥
<code>\aegeanthreehundred</code>	°°°	<code>\aegeanweightthirdsubunit</code>	℥
<code>\aegeanfourhundred</code>	°°°	<code>\aegeanweightfourthsubunit</code>	℥
<code>\aegeanfivehundred</code>	°°°	<code>\aegeandrymeasurefirstsubunit</code>	℥
<code>\aegeansixhundred</code>	°°°	<code>\aegeanliquidmeasurefirstsubunit</code>	℥
<code>\aegeansevenhundred</code>	°°°	<code>\aegeansecondsubunit</code>	℥
		<code>\aegeanthirdsubunit</code>	℥

30 Extra Math Relation symbols

Supported by Unicode and unicode-math and NewComputerModern	Unsupported by Unicode Added in NewComputerModern
\in <code>\disin</code>	\notin <code>\ndisin</code>
\circ <code>\closure</code>	\oslash <code>\nclosure</code>
\uparrow <code>\uparrowarrow</code>	\Uparrow <code>\nuparrowarrow</code>
\downarrow <code>\downarrowarrow</code>	\Downarrow <code>\ndownarrowarrow</code>
\updownarrow <code>\updownarrowarrow</code>	\Updownarrow <code>\nupdownarrowarrow</code>
\curvearrowleft <code>\leftwvearrow</code>	\nleftwvearrow <code>\nleftwvearrow</code>
\curvearrowright <code>\rightwvearrow</code>	\nrightwvearrow <code>\nrightwvearrow</code>
\Uparrow <code>\twoheaduparrow</code>	\nUparrow <code>\ntwoheaduparrow</code>
\Downarrow <code>\twoheaddownarrow</code>	\nDownarrow <code>\ntwoheaddownarrow</code>
\mapsto <code>\mapsfrom</code>	\nmapsto <code>\nmapsfrom</code>
\Uparrow <code>\mapsup</code>	\nUparrow <code>\nmapsup</code>
\mapsto <code>\mapsto</code>	\nmapsto <code>\nmapsto</code>
\Downarrow <code>\mapsdown</code>	\nDownarrow <code>\nmapsdown</code>
\updownarrow <code>\updownarrowarrowbar</code>	\nupdownarrow <code>\nupdownarrowarrowbar</code>
\hookleftarrow <code>\hookleftarrow</code>	\nhookleftarrow <code>\nhookleftarrow</code>
\hookrightarrow <code>\hookrightarrow</code>	\nhookrightarrow <code>\nhookrightarrow</code>
\looparrowleft <code>\looparrowleft</code>	\nlooparrowleft <code>\nlooparrowleft</code>
\looparrowright <code>\looparrowright</code>	\nlooparrowright <code>\nlooparrowright</code>
\leftrightsquigarrow <code>\leftrightsquigarrow</code>	\nleftrightsquigarrow <code>\nleftrightsquigarrow</code>
\downarrow <code>\downzigzagarrow</code>	\ndownzigzagarrow <code>\ndownzigzagarrow</code>
\lsh <code>\Lsh</code>	\nLsh <code>\nLsh</code>
\rsh <code>\Rsh</code>	\nRsh <code>\nRsh</code>
\ldsh <code>\Ldsh</code>	\nLdsh <code>\nLdsh</code>
\rdsh <code>\Rdsh</code>	\nRdsh <code>\nRdsh</code>
\linefeed <code>\linefeed</code>	\nlinefeed <code>\nlinefeed</code>
\carriagereturn <code>\carriagereturn</code>	\ncarriagereturn <code>\ncarriagereturn</code>
\curvearrowleft <code>\curvearrowleft</code>	\ncurvearrowleft <code>\ncurvearrowleft</code>
\curvearrowright <code>\curvearrowright</code>	\ncurvearrowright <code>\ncurvearrowright</code>

$\overleftarrow{\nwarrow}$	<code>\barovernorthwestarrow</code>	\nrightarrow	<code>\nbarovernorthwestarrow</code>
$\overleftrightarrow{\leftarrow}$	<code>\barleftarrowrightarrowbar</code>	\nleftrightarrow	<code>\nbarleftarrowrightarrowbar</code>
\circlearrowleft	<code>\acwopencirculararrow</code>	\circlearrowright	<code>\nacwopencirculararrow</code>
\circlearrowright	<code>\cwopencirculararrow</code>	\circlearrowleft	<code>\ncwopencirculararrow</code>
\leftharpoonup	<code>\leftharpoonup</code>	\nleftharpoonup	<code>\nleftharpoonup</code>
\leftharpoondown	<code>\leftharpoondown</code>	\nleftharpoondown	<code>\nleftharpoondown</code>
\upharpoonright	<code>\upharpoonright</code>	\nupharpoonright	<code>\nupharpoonright</code>
\upharpoonleft	<code>\upharpoonleft</code>	\nupharpoonleft	<code>\nupharpoonleft</code>
\rightharpoonup	<code>\rightharpoonup</code>	\nrightharpoonup	<code>\nrightharpoonup</code>
\rightharpoondown	<code>\rightharpoondown</code>	\nrightharpoondown	<code>\nrightharpoondown</code>
\downharpoonright	<code>\downharpoonright</code>	\ndownharpoonright	<code>\ndownharpoonright</code>
\downharpoonleft	<code>\downharpoonleft</code>	\ndownharpoonleft	<code>\ndownharpoonleft</code>
\rightleftarrows	<code>\rightleftarrows</code>	\nrightleftarrows	<code>\nrightleftarrows</code>
\updownarrows	<code>\updownarrows</code>	\nupdownarrows	<code>\nupdownarrows</code>
\leftrightarrows	<code>\leftrightarrows</code>	\nleftrightarrows	<code>\nleftrightarrows</code>
\upuparrows	<code>\upuparrows</code>	\nupuparrows	<code>\nupuparrows</code>
\downdownarrows	<code>\downdownarrows</code>	\ndowndownarrows	<code>\ndowndownarrows</code>
\leftrightharpoons	<code>\leftrightharpoons</code>	\nleftrightharpoons	<code>\nleftrightharpoons</code>
\rightleftharpoons	<code>\rightleftharpoons</code>	\nrightleftharpoons	<code>\nrightleftharpoons</code>
\Uparrow	<code>\Uparrow</code>	\nUparrow	<code>\nUparrow</code>
\Downarrow	<code>\Downarrow</code>	\nDownarrow	<code>\nDownarrow</code>
\Updownarrow	<code>\Updownarrow</code>	\nUpdownarrow	<code>\nUpdownarrow</code>
\Nwarrow	<code>\Nwarrow</code>	\nNwarrow	<code>\nNwarrow</code>
\Nearrow	<code>\Nearrow</code>	\nNearrow	<code>\nNearrow</code>
\Searrow	<code>\Searrow</code>	\nSearrow	<code>\nSearrow</code>
\Swarrow	<code>\Swarrow</code>	\nSwarrow	<code>\nSwarrow</code>
\Lleftarrow	<code>\Lleftarrow</code>	\nLleftarrow	<code>\nLleftarrow</code>
\Rrightarrow	<code>\Rrightarrow</code>	\nRrightarrow	<code>\nRrightarrow</code>
\leftsquigarrow	<code>\leftsquigarrow</code>	\nleftsquigarrow	<code>\nleftsquigarrow</code>

\rightsquigarrow	<code>\rightsquigarrow</code>	\nrightarrow	<code>\nrightarrow</code>
\Updownarrow	<code>\nHuparrow</code>	\nleftrightarrow	<code>\nnHuparrow</code>
\Downarrow	<code>\nHdownarrow</code>	\nleftrightarrow	<code>\nnHdownarrow</code>
\dashleftarrow	<code>\leftdasharrow</code>	\dashleftarrow	<code>\nleftdasharrow</code>
\dashedrightarrow	<code>\updasharrow</code>	\dashrightarrow	<code>\nupdasharrow</code>
\dashrightarrow	<code>\righdasharrow</code>	\dashrightarrow	<code>\nrighdasharrow</code>
\dashedrightarrow	<code>\downdasharrow</code>	\dashrightarrow	<code>\ndowndasharrow</code>
\barleftarrow	<code>\barleftarrow</code>	\barleftarrow	<code>\nbarleftarrow</code>
\rightarrowbar	<code>\rightarrowbar</code>	\rightarrowbar	<code>\nrightarrowbar</code>
\leftarrowbar	<code>\leftwhitearrow</code>	\leftarrowbar	<code>\nleftwhitearrow</code>
\uparrowbar	<code>\upwhitearrow</code>	\uparrowbar	<code>\nupwhitearrow</code>
\Rightarrow	<code>\rightwhitearrow</code>	\Rightarrow	<code>\nrighwhitearrow</code>
\Downarrow	<code>\downwhitearrow</code>	\Downarrow	<code>\ndownwhitearrow</code>
\Updownarrow	<code>\whitearrowupfrombar</code>	\Updownarrow	<code>\nwhitearrowupfrombar</code>
\rightarrow	<code>\circleonrightarrow</code>	\rightarrow	<code>\ncircleonrightarrow</code>
\Updownarrow	<code>\downuparrows</code>	\Updownarrow	<code>\ndownuparrows</code>
\Rightarrow	<code>\rightthreearrows</code>	\Rightarrow	<code>\nrighthreearrows</code>
\leftarrow	<code>\leftarrowtriangle</code>	\leftarrow	<code>\nleftarrowtriangle</code>
\rightarrow	<code>\rightarrowtriangle</code>	\rightarrow	<code>\nrightarrowtriangle</code>
\leftrightarrow	<code>\leftrightarrowtriangle</code>	\leftrightarrow	<code>\nleftrightarrowtriangle</code>
\in	<code>\smallin</code>	\notin	<code>\nsmallin</code>
\ni	<code>\smallni</code>	\nexists	<code>\nsmallni</code>
\propto	<code>\propto</code>	\npropto	<code>\npropto</code>
\colon	<code>\dashcolon</code>	\ncolon	<code>\ndashcolon</code>
\dotso	<code>\dotsminusdots</code>	\nminus	<code>\ndotsminusdots</code>
\sim	<code>\kernelcontraction</code>	\ncong	<code>\nkernelcontraction</code>
\backsim	<code>\backsim</code>	\nbacksim	<code>\nbacksim</code>
\eqsim	<code>\eqsim</code>	\neqsim	<code>\neqsim</code>
\approx	<code>\approx</code>	\napprox	<code>\napprox</code>

\approx	<code>\approxident</code>	\napprox	<code>\napproxident</code>
\cong	<code>\backcong</code>	\ncong	<code>\nbackcong</code>
\bumpeq	<code>\Bumpeq</code>	\nbumpeq	<code>\nBumpeq</code>
\bumpeq	<code>\bumpeq</code>	\nbumpeq	<code>\nbumpeq</code>
\doteq	<code>\doteq</code>	\ndoteq	<code>\ndoteq</code>
\doteq	<code>\Doteq</code>	\nDoteq	<code>\nDoteq</code>
\fallingdotseq	<code>\fallingdotseq</code>	\nfallingdotseq	<code>\nfallingdotseq</code>
\risingdotseq	<code>\risingdotseq</code>	\nrisingdotseq	<code>\nrisingdotseq</code>
\coloneq	<code>\coloneq</code>	\ncoloneq	<code>\ncoloneq</code>
\eqcolon	<code>\eqcolon</code>	\neqcolon	<code>\neqcolon</code>
\eqcirc	<code>\eqcirc</code>	\neqcirc	<code>\neqcirc</code>
\circeq	<code>\circeq</code>	\ncirceq	<code>\ncirceq</code>
\arceq	<code>\arceq</code>	\narceq	<code>\narceq</code>
\wedgeeq	<code>\wedgeeq</code>	\nwedgeeq	<code>\nwedgeeq</code>
\veeeq	<code>\veeeq</code>	\nveeeq	<code>\nveeeq</code>
\stareq	<code>\stareq</code>	\nstareq	<code>\nstareq</code>
\triangleq	<code>\triangleq</code>	\ntriangleq	<code>\ntriangleq</code>
$\stackrel{\text{def}}{=}$	<code>\eqdef</code>	$\nstackrel{\text{def}}{=}$	<code>\neqdef</code>
\measeq	<code>\measeq</code>	\nmeaseq	<code>\nmeaseq</code>
\questeq	<code>\questeq</code>	\nquesteq	<code>\nquesteq</code>
\equiv	<code>\Equiv</code>	\nEquiv	<code>\nEquiv</code>
\leq	<code>\leqq</code>	\nleqq	<code>\nleqq</code>
\geq	<code>\geqq</code>	\ngeqq	<code>\ngeqq</code>
\ll	<code>\ll</code>	\nll	<code>\nll</code>
\gg	<code>\gg</code>	\ngg	<code>\ngg</code>
\between	<code>\between</code>	\nbetween	<code>\nbetween</code>
\prec	<code>\precsim</code>	\nprec	<code>\nprecsim</code>
\succ	<code>\succsim</code>	\nsucc	<code>\nsuccsim</code>
\sqsubset	<code>\sqsubset</code>	\nsqsubset	<code>\nsqsubset</code>

\sqsubset	<code>\sqsubset</code>	\nmid	<code>\nsqsubset</code>
\dashv	<code>\dashv</code>	\nmid	<code>\ndashv</code>
\vdash	<code>\assert</code>	\nmid	<code>\nassert</code>
\vDash	<code>\vDash</code>	\nmid	<code>\nvDash</code>
\rhd	<code>\prurel</code>	\nmid	<code>\nprurel</code>
\rhd	<code>\scurel</code>	\nmid	<code>\nscurel</code>
$\circ\bullet$	<code>\origof</code>	\circ/\bullet	<code>\norigof</code>
$\bullet\circ$	<code>\imageof</code>	\bullet/\circ	<code>\nimageof</code>
\multimap	<code>\multimap</code>	\nmid	<code>\nmultimap</code>
\backsimeq	<code>\backsimeq</code>	\nmid	<code>\nbacksimeq</code>
\curlyvee	<code>\curlyvee</code>	\nmid	<code>\ncurlyvee</code>
\curlywedge	<code>\curlywedge</code>	\nmid	<code>\ncurlywedge</code>
\subseteq	<code>\Subset</code>	\nsubseteq	<code>\nSubset</code>
\supseteq	<code>\Supset</code>	\nsubseteq	<code>\nSupset</code>
\cap	<code>\Cap</code>	\nmid	<code>\nCap</code>
\cup	<code>\Cup</code>	\nmid	<code>\nCup</code>
\pitchfork	<code>\pitchfork</code>	\nmid	<code>\npitchfork</code>
\lessdot	<code>\lessdot</code>	\nmid	<code>\nlessdot</code>
\gtrdot	<code>\gtrdot</code>	\nmid	<code>\ngtrdot</code>
\lll	<code>\lll</code>	\lll	<code>\nlll</code>
\ggg	<code>\ggg</code>	\ggg	<code>\nggg</code>
\lesseqgtr	<code>\lesseqgtr</code>	\nmid	<code>\nlesseqgtr</code>
\gtreqless	<code>\gtreqless</code>	\nmid	<code>\ngtreqless</code>
\leqless	<code>\leqless</code>	\nmid	<code>\neqless</code>
\geqgtr	<code>\geqgtr</code>	\nmid	<code>\neqgtr</code>
\curlyeqprec	<code>\curlyeqprec</code>	\nmid	<code>\ncurlyeqprec</code>
\curlyeqsucc	<code>\curlyeqsucc</code>	\nmid	<code>\ncurlyeqsucc</code>
\varisins	<code>\varisins</code>	\nmid	<code>\nvarisins</code>
\isins	<code>\isins</code>	\nmid	<code>\nisins</code>

$\acute{\epsilon}$	<code>\isindot</code>	$\cancel{\epsilon}$	<code>\nisindot</code>
$\bar{\epsilon}$	<code>\varisinobar</code>	$\cancel{\bar{\epsilon}}$	<code>\nvarisinobar</code>
$\bar{\epsilon}$	<code>\isinobar</code>	$\cancel{\bar{\epsilon}}$	<code>\nisinobar</code>
\in	<code>\isinvb</code>	$\cancel{\in}$	<code>\nisinvb</code>
\in	<code>\isinE</code>	$\cancel{\in}$	<code>\nisinE</code>
\ni	<code>\nisd</code>	$\cancel{\ni}$	<code>\nnisd</code>
\ni	<code>\varnis</code>	$\cancel{\ni}$	<code>\nvarnis</code>
\ni	<code>\nis</code>	$\cancel{\ni}$	<code>\nnis</code>
$\bar{\ni}$	<code>\varniobar</code>	$\cancel{\bar{\ni}}$	<code>\nvarniobar</code>
$\bar{\ni}$	<code>\niobar</code>	$\cancel{\bar{\ni}}$	<code>\nniobar</code>
\in	<code>\bagmember</code>	$\cancel{\in}$	<code>\nbagmember</code>
\frown	<code>\frown</code>	$\cancel{\frown}$	<code>\nfrown</code>
\smile	<code>\smile</code>	$\cancel{\smile}$	<code>\nsmile</code>
\triangle	<code>\vartriangle</code>	$\cancel{\triangle}$	<code>\nvartriangle</code>
\perp	<code>\perp</code>	$\cancel{\perp}$	<code>\nperp</code>
\subset	<code>\subsetcirc</code>	$\cancel{\subset}$	<code>\nsubsetcirc</code>
\supset	<code>\supsetcirc</code>	$\cancel{\supset}$	<code>\nsupsetcirc</code>
\vee	<code>\veedot</code>	$\cancel{\vee}$	<code>\nveedot</code>
\wedge	<code>\wedgedot</code>	$\cancel{\wedge}$	<code>\nwedgedot</code>
ψ	<code>\upin</code>	$\cancel{\psi}$	<code>\nupin</code>
\dashv	<code>\DashVDash</code>	$\cancel{\dashv}$	<code>\nDashVDash</code>
\dashv	<code>\dashVdash</code>	$\cancel{\dashv}$	<code>\ndashVdash</code>
\multimap	<code>\multimapinv</code>	$\cancel{\multimap}$	<code>\nmultimapinv</code>
\longdash	<code>\vlongdash</code>	$\cancel{\longdash}$	<code>\nvlongdash</code>
\longdashv	<code>\longdashv</code>	$\cancel{\longdashv}$	<code>\nlongdashv</code>
\circlearrowleft	<code>\cirbot</code>	$\cancel{\circlearrowleft}$	<code>\ncirbot</code>
\lozenge	<code>\lozengeminus</code>	$\cancel{\lozenge}$	<code>\nlozengeminus</code>
\diamond	<code>\concavediamond</code>	$\cancel{\diamond}$	<code>\nconcavediamond</code>
\diamond	<code>\concavediamondtickleft</code>	$\cancel{\diamond}$	<code>\nconcavediamondtickleft</code>

	<code>\downarrowbar</code>		<code>\ndownarrowbar</code>
	<code>\nvrightarrowtail</code>		<code>\nnvrightarrowtail</code>
	<code>\nVrightarrowtail</code>		<code>\nnVrightarrowtail</code>
	<code>\twoheadrightarrowtail</code>		<code>\ntwoheadrightarrowtail</code>
	<code>\nvtwoheadrightarrowtail</code>		<code>\nnvtwoheadrightarrowtail</code>
	<code>\nVtwoheadrightarrowtail</code>		<code>\nnVtwoheadrightarrowtail</code>
	<code>\lefttail</code>		<code>\nlefttail</code>
	<code>\righttail</code>		<code>\nrighttail</code>
	<code>\leftdbltail</code>		<code>\nleftdbltail</code>
	<code>\rightdbltail</code>		<code>\nrightdbltail</code>
	<code>\diamondleftarrow</code>		<code>\ndiamondleftarrow</code>
	<code>\rightarrowtaildiamond</code>		<code>\nrightarrowtaildiamond</code>
	<code>\diamondleftarrowbar</code>		<code>\ndiamondleftarrowbar</code>
	<code>\barrightarrowtaildiamond</code>		<code>\nbarrightarrowtaildiamond</code>
	<code>\nwsearrow</code>		<code>\nnwsearrow</code>
	<code>\neswarrow</code>		<code>\nneswarrow</code>
	<code>\hknwarrow</code>		<code>\nhknwarrow</code>
	<code>\hknearrow</code>		<code>\nhknearrow</code>
	<code>\hksearrow</code>		<code>\nhksearrow</code>
	<code>\hkswarrow</code>		<code>\nhkswarrow</code>
	<code>\rightarrowcurvedarrow</code>		<code>\nrightarrowcurvedarrow</code>
	<code>\uprightcurvearrow</code>		<code>\nuprightcurvearrow</code>
	<code>\downrightcurvedarrow</code>		<code>\ndownrightcurvedarrow</code>
	<code>\leftdowncurvedarrow</code>		<code>\nleftdowncurvedarrow</code>
	<code>\rightdowncurvedarrow</code>		<code>\nrightdowncurvedarrow</code>
	<code>\cwrightarcarrow</code>		<code>\ncwrightarcarrow</code>
	<code>\acwleftarcarrow</code>		<code>\nacwleftarcarrow</code>
	<code>\acwoverarcarrow</code>		<code>\nacwoverarcarrow</code>
	<code>\acwunderarcarrow</code>		<code>\nacwunderarcarrow</code>

	<code>\curvearrowrightminus</code>		<code>\ncurvearrowrightminus</code>
	<code>\curvearrowleftplus</code>		<code>\ncurvearrowleftplus</code>
	<code>\cwundercurvearrow</code>		<code>\ncwundercurvearrow</code>
	<code>\ccwundercurvearrow</code>		<code>\nccwundercurvearrow</code>
	<code>\acwcirculararrow</code>		<code>\nacwcirculararrow</code>
	<code>\cwcirculararrow</code>		<code>\ncwcirculararrow</code>
	<code>\rightarrowshortleftarrow</code>		<code>\nrightarrowshortleftarrow</code>
	<code>\leftarrowshortrightarrow</code>		<code>\nleftarrowshortrightarrow</code>
	<code>\shortrightarrowleftarrow</code>		<code>\nshortrightarrowleftarrow</code>
	<code>\rightarrowplus</code>		<code>\nrightarrowplus</code>
	<code>\leftarrowplus</code>		<code>\nleftarrowplus</code>
	<code>\rightarrowx</code>		<code>\nrightarrowx</code>
	<code>\leftrightarrowcircle</code>		<code>\nleftrightarrowcircle</code>
	<code>\twoheaduparrowcircle</code>		<code>\ntwoheaduparrowcircle</code>
	<code>\leftrightharpoonupdown</code>		<code>\nleftrightharpoonupdown</code>
	<code>\leftrightharpoondownup</code>		<code>\nleftrightharpoondownup</code>
	<code>\updownharpoonrightleft</code>		<code>\nupdownharpoonrightleft</code>
	<code>\updownharpoonleftright</code>		<code>\nupdownharpoonleftright</code>
	<code>\leftrightharpoonupup</code>		<code>\nleftrightharpoonupup</code>
	<code>\updownharpoonrightright</code>		<code>\nupdownharpoonrightright</code>
	<code>\leftrightharpoondowndown</code>		<code>\nleftrightharpoondowndown</code>
	<code>\updownharpoonleftleft</code>		<code>\nupdownharpoonleftleft</code>
	<code>\barleftharpoonup</code>		<code>\nbarleftharpoonup</code>
	<code>\rightharpoonupbar</code>		<code>\nrightharpoonupbar</code>
	<code>\barupharpoonright</code>		<code>\nbarupharpoonright</code>
	<code>\downharpoonrightbar</code>		<code>\ndownharpoonrightbar</code>
	<code>\barleftharpoondown</code>		<code>\nbarleftharpoondown</code>
	<code>\rightharpoondownbar</code>		<code>\nrightharpoondownbar</code>
	<code>\barupharpoonleft</code>		<code>\nbarupharpoonleft</code>

\downarrow	<code>\downharpoonleftbar</code>	\nmid	<code>\ndownharpoonleftbar</code>
\leftharpoonup	<code>\leftharpoonupbar</code>	\nleftarrow	<code>\nleftharpoonupbar</code>
$\bar{\rightarrow}$	<code>\barrightharpoonup</code>	\nrightarrow	<code>\nbarrightharpoonup</code>
\upharpoonright	<code>\upharpoonrightbar</code>	\nmid	<code>\nupharpoonrightbar</code>
$\bar{\downarrow}$	<code>\bardownharpoonright</code>	\nmid	<code>\nbardownharpoonright</code>
\leftharpoonup	<code>\leftharpoonupdownbar</code>	\nleftarrow	<code>\nleftharpoonupdownbar</code>
$\bar{\rightarrow}$	<code>\barrightharpoonupdown</code>	\nrightarrow	<code>\nbarrightharpoonupdown</code>
\upharpoonleft	<code>\upharpoonleftbar</code>	\nmid	<code>\nupharpoonleftbar</code>
$\bar{\downarrow}$	<code>\bardownharpoonleft</code>	\nmid	<code>\nbardownharpoonleft</code>
\Leftarrow	<code>\leftharpoonsupdown</code>	\nLeftarrow	<code>\nleftharpoonsupdown</code>
\Uparrow	<code>\upharpoonsleftrightarrow</code>	\nUparrow	<code>\nupharpoonsleftrightarrow</code>
\Rightarrow	<code>\rightharpoonsupdown</code>	\nRightarrow	<code>\nrightharpoonsupdown</code>
\Downarrow	<code>\downharpoonsleftrightarrow</code>	\nDownarrow	<code>\ndownharpoonsleftrightarrow</code>
\Leftarrow	<code>\leftrightharpoonsup</code>	\nLeftarrow	<code>\nleftrightharpoonsup</code>
\Rightarrow	<code>\leftrightharpoonsdown</code>	\nRightarrow	<code>\nleftrightharpoonsdown</code>
\Leftarrow	<code>\rightleftharpoonsup</code>	\nLeftarrow	<code>\nrightleftharpoonsup</code>
\Rightarrow	<code>\rightleftharpoonsdown</code>	\nRightarrow	<code>\nrightleftharpoonsdown</code>
\Leftarrow	<code>\leftharpoonupdash</code>	\nLeftarrow	<code>\nleftharpoonupdash</code>
\Rightarrow	<code>\dashleftharpoonupdown</code>	\nRightarrow	<code>\ndashleftharpoonupdown</code>
\Rightarrow	<code>\rightharpoonupdash</code>	\nRightarrow	<code>\nrightharpoonupdash</code>
\Rightarrow	<code>\dashrightharpoonupdown</code>	\nRightarrow	<code>\ndashrightharpoonupdown</code>
\Updownarrow	<code>\updownharpoonsleftrightarrow</code>	\nUpdownarrow	<code>\nupdownharpoonsleftrightarrow</code>
\Downuparrow	<code>\downupharpoonsleftrightarrow</code>	\nDownuparrow	<code>\ndownupharpoonsleftrightarrow</code>
\Rightarrow	<code>\rightimply</code>	\nRightarrow	<code>\nrightimply</code>
\Rightarrow	<code>\equalrightarrow</code>	\nRightarrow	<code>\nequalrightarrow</code>
\Rightarrow	<code>\similarrightarrow</code>	\nRightarrow	<code>\nsimilarrightarrow</code>
\Leftarrow	<code>\leftarrowssimilar</code>	\nLeftarrow	<code>\nleftarrowssimilar</code>
\Rightarrow	<code>\rightarrowssimilar</code>	\nRightarrow	<code>\nrightarrowssimilar</code>
\Rightarrow	<code>\rightarrowapprox</code>	\nRightarrow	<code>\nrightarrowapprox</code>

\leftrightsquigarrow	<code>\ltlarr</code>	\nleftarrow	<code>\nltlarr</code>
\leftarrowleftarrow	<code>\leftarrowless</code>	\nleftarrowleftarrow	<code>\nleftarrowless</code>
\rightarrowrightarrow	<code>\gtrarr</code>	\nrightarrowrightarrow	<code>\ngtrarr</code>
\hookrightarrow	<code>\subrarr</code>	\nhookrightarrow	<code>\nsubrarr</code>
\leftarrowsubset	<code>\leftarrowsubset</code>	\nleftarrowsubset	<code>\nleftarrowsubset</code>
\rightarrowsupset	<code>\suplarr</code>	\nrightarrowsupset	<code>\nsuplarr</code>
\frownleft	<code>\leftfishtail</code>	\nleftfishtail	<code>\nleftfishtail</code>
\frownright	<code>\rightfishtail</code>	\nrightfishtail	<code>\nrightfishtail</code>
\upfrown	<code>\upfishtail</code>	\nupfrown	<code>\nupfishtail</code>
\downfrown	<code>\downfishtail</code>	\ndownfrown	<code>\ndownfishtail</code>
\Vdash	<code>\Vvert</code>	\nVdash	<code>\nVvert</code>
\lparenless	<code>\lparenless</code>	\nlparenless	<code>\nlparenless</code>
\rparenengtr	<code>\rparenengtr</code>	\nrparenengtr	<code>\nrparenengtr</code>
\Lparenengtr	<code>\Lparenengtr</code>	\nLparenengtr	<code>\nLparenengtr</code>
\Rparenless	<code>\Rparenless</code>	\nRparenless	<code>\nRparenless</code>
\Rrightarrow	<code>\rtriltri</code>	\nRrightarrow	<code>\nrtriltri</code>
\triangleleft	<code>\ltrivb</code>	\ntriangleleft	<code>\nltrivb</code>
\triangleright	<code>\vbrtri</code>	\nvbrtri	<code>\nvbrtri</code>
$\circ\circ$	<code>\dualmap</code>	$\circ\circ$	<code>\ndualmap</code>
\trianglelefteq	<code>\lrtriangleeq</code>	\ntrianglelefteq	<code>\nlrtriangleeq</code>
\sqcup	<code>\shuffle</code>	\nshuffle	<code>\nshuffle</code>
\models	<code>\gleichstark</code>	\nmodels	<code>\ngleichstark</code>
\rightarrowtail	<code>\ruledelayed</code>	\nrightarrowtail	<code>\nruledelayed</code>
\triangleleftsub	<code>\dsub</code>	\ntriangleleftsub	<code>\ndsub</code>
\trianglerightsub	<code>\rsub</code>	\ntrianglerightsub	<code>\nrsub</code>
\doteq	<code>\eqdot</code>	\neqdot	<code>\neqdot</code>
\doteqsim	<code>\dotequiv</code>	\ndoteqsim	<code>\ndotequiv</code>
\approx	<code>\simminussim</code>	\napprox	<code>\nsimminussim</code>
\cong	<code>\congdot</code>	\ncong	<code>\ncongdot</code>

$\stackrel{*}{=}$	<code>\asteq</code>	\neq	<code>\nasteq</code>
$\hat{\approx}$	<code>\hatapprox</code>	$\nhat{\approx}$	<code>\nhatapprox</code>
\approx	<code>\approxeqq</code>	\napprox	<code>\napproxeqq</code>
\approx	<code>\eqqsim</code>	\neqqsim	<code>\neqqsim</code>
\coloneqq	<code>\Coloneq</code>	\ncoloneqq	<code>\nColoneq</code>
$=$	<code>\eqeq</code>	\neqeq	<code>\neqeq</code>
$==$	<code>\eqeqeq</code>	\neqeqeq	<code>\neqeqeq</code>
$\ddot{=}$	<code>\ddotseq</code>	$\n\ddot{=}$	<code>\n\ddotseq</code>
\equiv	<code>\equivDD</code>	$\n\equiv$	<code>\nequivDD</code>
\ltcurlyeq	<code>\ltcir</code>	\nltcurlyeq	<code>\nltcir</code>
\gtcurlyeq	<code>\gtcir</code>	\ngtcurlyeq	<code>\ngtcir</code>
\ltquest	<code>\ltquest</code>	\nltquest	<code>\nltquest</code>
\gtquest	<code>\gtquest</code>	\ngtquest	<code>\ngtquest</code>
\leqslant	<code>\leqslant</code>	\nleqslant	<code>\nleqslant</code>
\geqslant	<code>\geqslant</code>	\ngeqslant	<code>\ngeqslant</code>
\lesdot	<code>\lesdot</code>	\nlesdot	<code>\nlesdot</code>
\gesdot	<code>\gesdot</code>	\ngesdot	<code>\ngesdot</code>
\lesdoto	<code>\lesdoto</code>	\nlesdoto	<code>\nlesdoto</code>
\gesdoto	<code>\gesdoto</code>	\ngesdoto	<code>\ngesdoto</code>
\lesdotor	<code>\lesdotor</code>	\nlesdotor	<code>\nlesdotor</code>
$\gesdoto1$	<code>\gesdoto1</code>	$\ngesdoto1$	<code>\ngesdoto1</code>
\lessapprox	<code>\lessapprox</code>	\nlessapprox	<code>\nlessapprox</code>
\gtrapprox	<code>\gtrapprox</code>	\ngtrapprox	<code>\ngtrapprox</code>
\lesseqgtr	<code>\lesseqgtr</code>	\nlesseqgtr	<code>\nlesseqgtr</code>
\gtreqless	<code>\gtreqless</code>	\ngtreqless	<code>\ngtreqless</code>
\lsime	<code>\lsime</code>	\nlsime	<code>\nlsime</code>
\gsime	<code>\gsime</code>	\ngsime	<code>\ngsime</code>
\lsimg	<code>\lsimg</code>	\nlsimg	<code>\nlsimg</code>
\gsiml	<code>\gsiml</code>	\ngsiml	<code>\ngsiml</code>

\lll	<code>\lgE</code>	\lll	<code>\nlgE</code>
\lll	<code>\glE</code>	\lll	<code>\nglE</code>
\lll	<code>\lesges</code>	\lll	<code>\nlesges</code>
\lll	<code>\gesles</code>	\lll	<code>\ngesles</code>
\ll	<code>\eqslantless</code>	\ll	<code>\neqslantless</code>
\gg	<code>\eqslantgtr</code>	\gg	<code>\neqslantgtr</code>
\leq	<code>\elsdot</code>	\leq	<code>\nelsdot</code>
\geq	<code>\egsdot</code>	\geq	<code>\negsdot</code>
\leq	<code>\eqqless</code>	\leq	<code>\neqqless</code>
\gg	<code>\eqqgtr</code>	\gg	<code>\neqqgtr</code>
\ll	<code>\eqqslantless</code>	\ll	<code>\neqqslantless</code>
\gg	<code>\eqqslantgtr</code>	\gg	<code>\neqqslantgtr</code>
\sim	<code>\simless</code>	\sim	<code>\nsimless</code>
\sim	<code>\simgtr</code>	\sim	<code>\nsimgtr</code>
\sim	<code>\simlE</code>	\sim	<code>\nsimlE</code>
\sim	<code>\simgE</code>	\sim	<code>\nsimgE</code>
\ll	<code>\Lt</code>	\ll	<code>\nLt</code>
\gg	<code>\Gt</code>	\gg	<code>\nGt</code>
\ll	<code>\partialmeetcontraction</code>	\ll	<code>\npartialmeetcontraction</code>
\times	<code>\glj</code>	\times	<code>\nglj</code>
\times	<code>\gla</code>	\times	<code>\ngla</code>
\triangleleft	<code>\ltcc</code>	\triangleleft	<code>\nltcc</code>
\triangleright	<code>\gtcc</code>	\triangleright	<code>\ngtcc</code>
\triangleleft	<code>\lescc</code>	\triangleleft	<code>\nlescc</code>
\triangleright	<code>\gescc</code>	\triangleright	<code>\ngescc</code>
\ll	<code>\smt</code>	\ll	<code>\nsmt</code>
\gg	<code>\lat</code>	\gg	<code>\nlat</code>
\ll	<code>\smtE</code>	\ll	<code>\nsmtE</code>
\gg	<code>\late</code>	\gg	<code>\nlate</code>

\equiv	<code>\bumpeq</code>	\nbumpeq	<code>\nbumpeq</code>
\preceq	<code>\preceq</code>	\npreceq	<code>\npreceq</code>
\succeq	<code>\succeq</code>	\nsucceq	<code>\nsucceq</code>
\preceqq	<code>\preceqq</code>	\npreceqq	<code>\npreceqq</code>
\succeqq	<code>\succeqq</code>	\nsucceqq	<code>\nsucceqq</code>
\precapprox	<code>\precapprox</code>	\nprecapprox	<code>\nprecapprox</code>
\succapprox	<code>\succapprox</code>	\nsuccapprox	<code>\nsuccapprox</code>
\prec	<code>\Prec</code>	\nprec	<code>\nPrec</code>
\succ	<code>\Succ</code>	\nsucc	<code>\nSucc</code>
\subsetdot	<code>\subsetdot</code>	\nsubsetdot	<code>\nsubsetdot</code>
\supsetdot	<code>\supsetdot</code>	\nsupsetdot	<code>\nsupsetdot</code>
\subsetplus	<code>\subsetplus</code>	\nsubsetplus	<code>\nsubsetplus</code>
\supsetplus	<code>\supsetplus</code>	\nsupsetplus	<code>\nsupsetplus</code>
\subsetmult	<code>\subsetmult</code>	\nsubsetmult	<code>\nsubsetmult</code>
\supmult	<code>\supmult</code>	\nsupmult	<code>\nsupmult</code>
\subedot	<code>\subedot</code>	\nsubedot	<code>\nsubedot</code>
\supedot	<code>\supedot</code>	\nsupedot	<code>\nsupedot</code>
\subseteqq	<code>\subseteqq</code>	\nsubseteqq	<code>\nsubseteqq</code>
\supseteqq	<code>\supseteqq</code>	\nsupseteqq	<code>\nsupseteqq</code>
$\subset\sim$	<code>\subset\sim</code>	$\nsubset\sim$	<code>\nsubset\sim</code>
$\sup\sim$	<code>\sup\sim</code>	$\nsup\sim$	<code>\nsup\sim</code>
\subsetapprox	<code>\subsetapprox</code>	\nsubsetapprox	<code>\nsubsetapprox</code>
\supsetapprox	<code>\supsetapprox</code>	\nsupsetapprox	<code>\nsupsetapprox</code>
\lshook	<code>\lshook</code>	\nlsqhook	<code>\nlsqhook</code>
\rsqhook	<code>\rsqhook</code>	\nrsqhook	<code>\nrsqhook</code>
\subset	<code>\subset</code>	\nsubset	<code>\nsubset</code>
\supset	<code>\supset</code>	\nsupset	<code>\nsupset</code>
\subseteq	<code>\subseteq</code>	\nsubseteq	<code>\nsubseteq</code>
\supseteq	<code>\supseteq</code>	\nsupseteq	<code>\nsupseteq</code>
























































\supset	<code>\supsup</code>	$\not\supset$	<code>\nsupsup</code>
\supseteq	<code>\supsub</code>	$\not\supseteq$	<code>\nsupsub</code>
\subseteq	<code>\subsub</code>	$\not\subseteq$	<code>\nsubsub</code>
\supsetneq	<code>\supsup</code>	$\not\supsetneq$	<code>\nsupsup</code>
\supsetsubset	<code>\suphsub</code>	$\not\supsetsubset$	<code>\nsuphsub</code>
\supseteqneq	<code>\supdsub</code>	$\not\supseteqneq$	<code>\nsupdsub</code>
\pitchfork	<code>\forkv</code>	$\not\pitchfork$	<code>\nforkv</code>
\topfork	<code>\topfork</code>	$\not\topfork$	<code>\ntopfork</code>
\pitchfork	<code>\mlcp</code>	$\not\pitchfork$	<code>\nmlcp</code>
\dashleftarrow	<code>\shortlefttack</code>	$\not\dashleftarrow$	<code>\nshortlefttack</code>
\dashrightarrow	<code>\shortdowntack</code>	$\not\dashrightarrow$	<code>\nshortdowntack</code>
\dashv	<code>\shortuptack</code>	$\not\dashv$	<code>\nshortuptack</code>
\perp	<code>\perps</code>	$\not\perp$	<code>\nperps</code>
\Vdash	<code>\vDdash</code>	$\not\Vdash$	<code>\nvDdash</code>
\dashv	<code>\dashV</code>	$\not\dashv$	<code>\ndashV</code>
\Dashv	<code>\Dashv</code>	$\not\Dashv$	<code>\nDashv</code>
\DashV	<code>\DashV</code>	$\not\DashV$	<code>\nDashV</code>
\Vdash	<code>\varVdash</code>	$\not\Vdash$	<code>\nvarVdash</code>
\Barv	<code>\Barv</code>	$\not\Barv$	<code>\nBarv</code>
\vBar	<code>\vBar</code>	$\not\vBar$	<code>\nvBar</code>
\vBarv	<code>\vBarv</code>	$\not\vBarv$	<code>\nvBarv</code>
\barV	<code>\barV</code>	$\not\barV$	<code>\nbarV</code>
\Vbar	<code>\Vbar</code>	$\not\Vbar$	<code>\nVbar</code>
\Not	<code>\Not</code>	$\not\Not$	<code>\nNot</code>
\bNot	<code>\bNot</code>	$\not\bNot$	<code>\nbNot</code>
\circ	<code>\cirmid</code>	$\not\circ$	<code>\ncirmid</code>
\circ	<code>\midcir</code>	$\not\circ$	<code>\nmidcir</code>
\circ	<code>\topcir</code>	$\not\circ$	<code>\ntopcir</code>
\lll	<code>\lllnest</code>	$\not\lll$	<code>\nlllnest</code>

\ggg	<code>\gggnest</code>	\nggg	<code>\ngggnest</code>
\leqq	<code>\leqqslant</code>	\nleqq	<code>\nleqqslant</code>
\geqq	<code>\geqqslant</code>	\ngeqq	<code>\ngeqqslant</code>
\circleftarrow	<code>\circleonleftarrow</code>	\ncircleonleftarrow	<code>\ncircleonleftarrow</code>
\leftthreetimes	<code>\leftthreetimes</code>	\nleftthreetimes	<code>\nleftthreetimes</code>
\leftarrowplus	<code>\leftarrowonoplus</code>	\nleftarrowplus	<code>\nleftarrowonoplus</code>
\longleftsquigarrow	<code>\longleftsquigarrow</code>	\nlongleftsquigarrow	<code>\nlongleftsquigarrow</code>
\twoheadmapsfrom	<code>\twoheadmapsfrom</code>	\ntwoheadmapsfrom	<code>\ntwoheadmapsfrom</code>
\twoheadleftdbkarrow	<code>\twoheadleftdbkarrow</code>	\ntwoheadleftdbkarrow	<code>\ntwoheadleftdbkarrow</code>
\leftarrowdot	<code>\leftarrowdotarrow</code>	\nleftarrowdot	<code>\nleftarrowdotarrow</code>
\twoheadleftarrowtail	<code>\twoheadleftarrowtail</code>	\ntwoheadleftarrowtail	<code>\ntwoheadleftarrowtail</code>
\leftarrowx	<code>\leftarrowx</code>	\nleftarrowx	<code>\nleftarrowx</code>
\leftcurvedarrow	<code>\leftcurvedarrow</code>	\nleftcurvedarrow	<code>\nleftcurvedarrow</code>
\equalleftarrow	<code>\equalleftarrow</code>	\nequalleftarrow	<code>\nequalleftarrow</code>
\bsimilarleftarrow	<code>\bsimilarleftarrow</code>	\nbsimilarleftarrow	<code>\nbsimilarleftarrow</code>
\leftarrowbackapprox	<code>\leftarrowbackapprox</code>	\nleftarrowbackapprox	<code>\nleftarrowbackapprox</code>
\rightarrowgtr	<code>\rightarrowgtr</code>	\nrightarrowgtr	<code>\nrightarrowgtr</code>
\rightarrowsupset	<code>\rightarrowsupset</code>	\nrightarrowsupset	<code>\nrightarrowsupset</code>
\Lleftarrow	<code>\Lleftarrow</code>	\NLleftarrow	<code>\NLleftarrow</code>
\Rrightarrow	<code>\Rrightarrow</code>	\NRrightarrow	<code>\NRrightarrow</code>
\bsimilarrightarrow	<code>\bsimilarrightarrow</code>	\nbsimilarrightarrow	<code>\nbsimilarrightarrow</code>
\rightarrowbackapprox	<code>\rightarrowbackapprox</code>	\nrightarrowbackapprox	<code>\nrightarrowbackapprox</code>
\simleftarrow	<code>\simleftarrow</code>	\nsimilarleftarrow	<code>\nsimilarleftarrow</code>
\leftarrowapprox	<code>\leftarrowapprox</code>	\nleftarrowapprox	<code>\nleftarrowapprox</code>
\leftarrowbsimilar	<code>\leftarrowbsimilar</code>	\nleftarrowbsimilar	<code>\nleftarrowbsimilar</code>
\rightarrowbsimilar	<code>\rightarrowbsimilar</code>	\nrightarrowbsimilar	<code>\nrightarrowbsimilar</code>
\twoheadmapsto	<code>\twoheadmapsto</code>	\ntwoheadmapsto	<code>\ntwoheadmapsto</code>

Supported by Unicode	Unsupported by Unicode
Unsupported by unicode-math	Added in NewComputerModern
Supported by NewComputerModern	

\uparrow	<code>\whitemapsup</code>	\nuparrow	<code>\nwhitemapsup</code>
------------	---------------------------	-------------	----------------------------

	<code>\whitemapsupbar</code>		<code>\nwhitemapsupbar</code>
	<code>\whitemapsupvbar</code>		<code>\nwhitemapsupvbar</code>
	<code>\whitetwoheaduparrow</code>		<code>\nwhitetwoheaduparrow</code>
	<code>\whitetwoheadmapsup</code>		<code>\nwhitetwoheadmapsup</code>
	<code>\whitemapsto</code>		<code>\nwhitemapsto</code>
	<code>\narrowcorner</code>		<code>\nnarrowcorner</code>
	<code>\searrowcorner</code>		<code>\nsearrowcorner</code>
	<code>\whiteupdownarrow</code>		<code>\nwhiteupdownarrow</code>
	<code>\newwhitearrow</code>		<code>\nnewwhitearrow</code>
	<code>\nnwwhitearrow</code>		<code>\nnnwwhitearrow</code>
	<code>\sewwhitearrow</code>		<code>\nsewwhitearrow</code>
	<code>\swwhitearrow</code>		<code>\nswwhitearrow</code>
	<code>\leftrightwhitearrow</code>		<code>\nleftrightwhitearrow</code>
	<code>\leftblackarrow</code>		<code>\nleftblackarrow</code>
	<code>\upblackarrow</code>		<code>\nupblackarrow</code>
	<code>\downblackarrow</code>		<code>\ndownblackarrow</code>
	<code>\neblackarrow</code>		<code>\nneblackarrow</code>
	<code>\nwblackarrow</code>		<code>\nnwblackarrow</code>
	<code>\seblackarrow</code>		<code>\nseblackarrow</code>
	<code>\swblackarrow</code>		<code>\nswblackarrow</code>
	<code>\leftrightblackarrow</code>		<code>\nleftrightblackarrow</code>
	<code>\updownblackarrow</code>		<code>\nupdownblackarrow</code>
	<code>\rightarrowdown</code>		<code>\nrightarrowdown</code>
	<code>\rightarrowup</code>		<code>\nrightarrowup</code>
	<code>\leftarrowdown</code>		<code>\nleftarrowdown</code>
	<code>\leftarrowup</code>		<code>\nleftarrowup</code>
	<code>\lefttriangleheadarrow</code>		<code>\nlefttriangleheadarrow</code>
	<code>\uptriangleheadarrow</code>		<code>\nuptriangleheadarrow</code>
	<code>\righttriangleheadarrow</code>		<code>\nrighttriangleheadarrow</code>

	<code>\downtriangleheadarrow</code>		<code>\ndowntriangleheadarrow</code>
	<code>\leftrighthttriangleheadarrow</code>		<code>\nleftrighthttriangleheadarrow</code>
	<code>\updowntriangleheadarrow</code>		<code>\nupdowntriangleheadarrow</code>
	<code>\nwtriangleheadarrow</code>		<code>\nnwtriangleheadarrow</code>
	<code>\netriangleheadarrow</code>		<code>\nnetriangleheadarrow</code>
	<code>\setriangleheadarrow</code>		<code>\nsetriangleheadarrow</code>
	<code>\swtriangleheadarrow</code>		<code>\nswtriangleheadarrow</code>
	<code>\lefttriangleheaddasharrow</code>		<code>\nlefttriangleheaddasharrow</code>
	<code>\uptriangleheaddasharrow</code>		<code>\nuptriangleheaddasharrow</code>
	<code>\righttriangleheaddasharrow</code>		<code>\nrighttriangleheaddasharrow</code>
	<code>\downtriangleheaddasharrow</code>		<code>\ndowntriangleheaddasharrow</code>
	<code>\cwtriangleheadopencirclearrow</code>		<code>\ncwtriangleheadopencirclearrow</code>
	<code>\acwtriangleheadopencirclearrow</code>		<code>\nacwtriangleheadopencirclearrow</code>
	<code>\lefttriangleheadarrowbar</code>		<code>\nlefttriangleheadarrowbar</code>
	<code>\uptriangleheadarrowbar</code>		<code>\nuptriangleheadarrowbar</code>
	<code>\righttriangleheadarrowbar</code>		<code>\nrighttriangleheadarrowbar</code>
	<code>\downtriangleheadarrowbar</code>		<code>\ndowntriangleheadarrowbar</code>
	<code>\nwtriangleheadarrowbar</code>		<code>\nnwtriangleheadarrowbar</code>
	<code>\netriangleheadarrowbar</code>		<code>\nnetriangleheadarrowbar</code>
	<code>\setriangleheadarrowbar</code>		<code>\nsetriangleheadarrowbar</code>
	<code>\swtriangleheadarrowbar</code>		<code>\nswtriangleheadarrowbar</code>
	<code>\htabkey</code>		<code>\nhtabkey</code>
	<code>\vtabkey</code>		<code>\nvtabkey</code>
	<code>\ltrharrowrtrharrow</code>		<code>\nltrharrowrtrharrow</code>
	<code>\uptrharrowdowntrharrow</code>		<code>\nuptrharrowdowntrharrow</code>
	<code>\rtrharrowltrharrow</code>		<code>\nrtrharrowltrharrow</code>
	<code>\downtrharrowuptrharrow</code>		<code>\ndowntrharrowuptrharrow</code>
	<code>\ltrharrowrltrharrow</code>		<code>\nltrharrowrltrharrow</code>
	<code>\uptrharrowuptrharrow</code>		<code>\nuptrharrowuptrharrow</code>

\Rightarrow	<code>\rtrharrowrtrharrow</code>	\nRightarrow	<code>\nrtrharrowrtrharrow</code>
\Downarrow	<code>\downtrharrowdowntrharrow</code>	\nDownarrow	<code>\ndowntrharrowdowntrharrow</code>
\curvearrowright	<code>\acwtrhruarrow</code>	\ncurvearrowright	<code>\nacwtrhruarrow</code>
\Uparrow	<code>\acwtrhbuarrow</code>	\nUparrow	<code>\nacwtrhbuarrow</code>
\Crightarrow	<code>\acwtrhluarrow</code>	\nCrightarrow	<code>\nacwtrhluarrow</code>
\hookrightarrow	<code>\acwtrhtuarrow</code>	\nhookrightarrow	<code>\nacwtrhtuarrow</code>
\leftarrow	<code>\returnleft</code>	\nleftarrow	<code>\nreturnleft</code>
\rightarrow	<code>\returnright</code>	\nrightarrow	<code>\nreturnright</code>
\leftarrow	<code>\newlineleft</code>	\nleftarrow	<code>\nnewlineleft</code>
\rightarrow	<code>\newlinerright</code>	\nrightarrow	<code>\nnewlinerright</code>
\boxrightarrow	<code>\fourcornerarrowsacw</code>	\nboxrightarrow	<code>\nfourcornerarrowsacw</code>
\rightarrow	<code>\rightwardsblackarrow</code>	\nrightarrow	<code>\nrightwardsblackarrow</code>

References

- [AT] Antonis Tsolomitis, *The NewComputerModern font family*, TUGboat Vol. 42, No. 1, 2021.
- [IPAre] Council actions on revisions of the IPA, Phonetic Representation: b) Revision of the IPA, Journal of the International Phonetic Association, Volume 23, Issue 1, June 1993, pp. 32–34.
- [web1] <https://tex.stackexchange.com/a/697498>
- [web2] <https://tex.stackexchange.com/a/762017>