

Experimental unicode mathematical typesetting: The unicode-math package

Will Robertson

2009/10/11 v0.4

Abstract

Warning! This package is experimental and subject to change without regard for backwards compatibility. Performance issues may be encountered until algorithms are refined.

This package is intended to be a complete implementation of unicode maths for \LaTeX using the \XeTeX (and later, \LuaTeX) typesetting engines. With this package, changing maths fonts will be as easy as changing text fonts — not that there are many unicode maths fonts yet.

Maths input is simplified with unicode since literal glyphs may be entered instead of control sequences.

Contents

1	Introduction	3	7.3	The main <code>\setmathfont</code> macro	28
2	Acknowledgements	3	7.4	(Big) operators	36
3	Getting started	3	7.5	Radicals	39
3.1	Package options	3	7.6	Delimiters	39
4	Unicode maths font setup	4	7.7	Maths accents	42
4.1	Using multiple fonts	5	8	Font features	43
4.2	Script and scriptscript fonts/features	6	8.1	OpenType maths font features	43
5	Maths input	6	8.2	Script and scriptscript font options	43
5.1	Math ‘style’	6	8.3	Range processing	43
5.2	Bold style	7	8.4	Resolving Greek symbol name control sequences	48
5.3	Sans serif style	8	8.5	Setting up the mappings	50
5.4	All (the rest) of the mathematical alphabets	9	9	Maths alphabets mapping definitions	54
5.5	Miscellanea	10	9.1	Non-bold math alphabets	58
I	The unicode-math package	15	9.2	Bold math alphabets	62
6	Things we need	15	9.3	Definitions of the math symbols	69
6.1	Options	19	10	Epilogue	70
6.2	Overcoming <code>\@on-</code> lypreamble	24	II	stix table data extraction	78
6.3	Other things	25	A	Documenting maths support in the NFSS	80
7	Fundamentals	26	A.1	Overview	80
7.1	Enlarging the number of maths families	26	III	X_YTeX math font dimensions	81
7.2	<code>\DeclareMathSymbol</code> for unicode ranges	26			

1 Introduction

This document describes the unicode-math package, which is an *experimental* implementation of a macro to unicode glyph encoding for mathematical characters. Its intended use is for \XeTeX , although it is conjectured that some effort could be spent to create a cross-format package that would also work with $\text{Lua}\TeX$.

Users who desire to specify maths alphabets only (Greek and Latin letters) may wish to use Andrew Moschou’s mathspec package instead.

2 Acknowledgements

Many thanks to Microsoft for developing OpenType math as part of Office 2007; Jonathan Kew for implementing unicode math support in $\text{Xe}\TeX$; Barbara Beeton for her prodigious effort compiling the definitive list of unicode math glyphs and their $\text{L}\text{A}\text{T}\text{E}\text{X}$ names (inventing them where necessary), and also for her thoughtful replies to my sometimes incessant questions. Ross Moore and Chris Rowley have provided moral and technical support from the very early days with great insight into the issues we face trying to extend and use TEX in the future. Apostolos Syropoulos, Joel Salomon, and Khaled Hosny have been fantastic beta testers.

3 Getting started

Load unicode-math as a regular $\text{L}\text{A}\text{T}\text{E}\text{X}$ package. It should be loaded after any other maths or font-related package in case it needs to overwrite their definitions. Here’s an example:

```
\usepackage{amsmath} % if desired
\usepackage{unicode-math}
\setmathfont{Cambria Math}
```

3.1 Package options

Package options may be set when the package is loaded or at any later stage with the `\unimathsetup` command. Therefore, the following two examples are equivalent:

```
\usepackage[math-style=TeX]{unicode-math}
% OR
\usepackage{unicode-math}
\unimathsetup{math-style=TeX}
```

Table 1: Package options.

Option	Description	See...
<code>math-style</code>	Style of letters	section §5.1
<code>bold-style</code>	Style of bold letters	section §5.2
<code>sans-style</code>	Style of sans serif letters	section §5.3
<code>nabla</code>	Style of the nabla symbol	section §5.5.1
<code>partial</code>	Style of the partial symbol	section §5.5.2
<code>vargreek-shape</code>	Style of phi and epsilon	section §5.5.3
<code>colon</code>	Behaviour of <code>\colon</code>	section §5.5.6
<code>slash-delimiter</code>	Glyph to use for ‘stretchy’ slash	section §5.5.7

Note, however, that some package options affects how maths is initialised and changing an option such as `math-style` will not take effect until a new maths font is set up.

Package options may *also* be used when declaring new maths fonts, passed via options to the `\setmathfont` command. Therefore, the following two examples are equivalent:

```
\unimathsetup{math-style=TeX}
\setmathfont{Cambria Math}
% OR
\setmathfont[math-style=TeX]{Cambria Math}
```

A short list of package options is shown in table 1. See following sections for more information.

4 Unicode maths font setup

In the ideal case, a single unicode font will contain all maths glyphs we need. The file `unicode-math-table.tex` (based on Barbara Beeton’s `stix` table) provides the mapping between unicode maths glyphs and macro names (all 3298 — or however many — of them!). A single command

```
\setmathfont[<font features>]{<font name>}
```

implements this for every every symbol and alphabetic variant. That means `x` to x , `\xi` to ξ , `\leq` to \leq , etc., `\mathcal{H}` to \mathcal{H} and so on, all for unicode glyphs within a single font.

This package deals well with unicode characters for maths input. This includes using literal Greek letters in formulae, resolving to upright or italic depending on preference.

Font features specific to `unicode-math` are shown in table 2. Package options (see table 1) may also be used. Other `fontspec` features are also valid.

Table 2: Maths font options.

Option	Description	See...
<code>range</code>	Style of letters	section §4.1
<code>script-font</code>	Font to use for sub- and super-scripts	section §4.2
<code>script-features</code>	Font features for sub- and super-scripts	section §4.2
<code>sscript-font</code>	Font to use for nested sub- and super-scripts	section §4.2
<code>sscript-features</code>	Font features for nested sub- and super-scripts	section §4.2

4.1 Using multiple fonts

There will probably be few cases where a single unicode maths font suffices (simply due to glyph coverage). The upcoming `srix` font comes to mind as a possible exception. It will therefore be necessary to delegate specific unicode ranges of glyphs to separate fonts:

```
\setmathfont[range=<unicode range>,<font features>]{<font name>}
```

where *<unicode range>* is a comma-separated list of unicode slots and ranges such as `{"27D0-"27EB,"27FF,"295B-"297F}`. You may also use the macro for accessing the glyph, such as `\int`, or whole collection of symbols with the same math type, such as `\mathopen`, or complete math alphabets such as `\mathbb`. (Only numerical slots, however, can be used in ranged declarations.)

4.1.1 Control over maths alphabets

Exact control over maths alphabets can be somewhat involved. Here is the current plan.

- `[range=\mathbb]` to use the font for ‘bb’ letters only.
- `[range=\mathbfssfit/{greek,Greek}]` for Greek lowercase and uppercase only (with `latin`, `Latin`, `num` as well for Latin lower-/upper-case and numbers).
- `[range=\mathsf->\mathbfssfit]` to map to different output alphabet(s) (which is rather useless right now but will become less useless in the future).

And now the trick. If a particular math alphabet is not defined in the font, fall back onto the lower-base plane (i.e., upright) glyphs. Therefore, to use an ASCII-encoded fractur font, for example, write

```
\setmathfont[range=\mathfrak]{SomeFrakturFont}
```

and because the math plane fractur glyphs will be missing, unicode-math will know to use the ASCII ones instead. If necessary (but why?) this behaviour can be forced with `[range=\mathfrac->\mathup]`.

4.2 Script and scriptscript fonts/features

Cambria Math uses OpenType font features to activate smaller optical sizes for scriptsize and scriptscriptsize symbols (the B and C , respectively, in A_{B_C}). Other fonts will possibly use entirely separate fonts.

Not yet implemented: Both of these options must be taken into account. I hope this will be mostly automatic from the users' points of view. The `+ssty` feature can be detected and applied automatically, and appropriate optical size information embedded in the fonts will ensure this latter case. Fine tuning should be possible automatically with `fontspec` options. We might have to wait until MnMath, for example, before we really know.

5 Maths input

X_YTeX's unicode support allows maths input through two methods. Like classical TeX, macros such as `\alpha`, `\sum`, `\pm`, `\leq`, and so on, provide verbose access to the entire repertoire of characters defined by unicode. The literal characters themselves may be used instead, for more readable input files.

5.1 Math 'style'

Classically, TeX uses italic lowercase Greek letters and *upright* uppercase Greek letters for variables in mathematics. This is contrary to the ISO standards of using italic forms for both upper- and lowercase. Furthermore, the French (contrary again, *quelle surprise*) have been known to use upright uppercase *Latin* letters as well as upright upper- and lowercase Greek. Finally, it is not unknown to use upright letters for all characters, as seen in the Euler fonts.

The unicode-math package accommodates these possibilities with an interface heavily inspired by Walter Schmidt's `lucimatx` package: a package option `math-style` that takes one of four arguments: `TeX`, `ISO`, `French`, or `upright` (case insensitive).

The philosophy behind the interface to the mathematical alphabet symbols lies in L^ATeX's attempt of separating content and formatting. Because input source text may come from a variety of places, the upright and 'mathematical' italic Latin and Greek alphabets are *unified* from the point of view of having a specified meaning in the source text. That is, to get a mathematical ' x ', either the ascii ('keyboard') letter `x` may be typed, or the actual unicode character may be used. Similarly for Greek letters. The upright or italic forms are then chosen based on the `math-style` package option.

If glyphs are desired that do not map as per the package option (for example, an upright ' g ' is desired but typing `g` yields ' g '), *markup* is required to specify this; to follow from the example: `\mathup{g}`. Maths alphabets commands such as `\mathup` are detailed later.

Table 3: Effects of the `math-style` package option.

Package option	Example	
	Latin	Greek
<code>math-style=ISO</code>	(a, z, B, X)	$(\alpha, \beta, \Gamma, \Xi)$
<code>math-style=TeX</code>	(a, z, B, X)	$(\alpha, \beta, \Gamma, \Xi)$
<code>math-style=French</code>	(a, z, B, X)	$(\alpha, \beta, \Gamma, \Xi)$
<code>math-style=upright</code>	(a, z, B, X)	$(\alpha, \beta, \Gamma, \Xi)$

Alternative interface However, some users may not like this convention of normalising their input. For them, an upright x is an upright ‘ x ’ and that’s that. (This will be the case when obtaining source text from copy/pasting PDF or Microsoft Word documents, for example.) For these users, the `literal` option to `math-style` will effect this behaviour.

The `math-style` options’ effects are shown in brief in table 3.

5.2 Bold style

Similar as in the previous section, ISO standards differ somewhat to \TeX ’s conventions (and classical typesetting) for ‘boldness’ in mathematics. In the past, it has been customary to use bold *upright* letters to denote things like vectors and matrices. For example, $\mathbf{M} = (M_x, M_y, M_z)$. Presumably, this was due to the relatively scarcity of bold italic fonts in the pre-digital typesetting era. It has been suggested that *italic* bold symbols are used nowadays instead.

Bold Greek letters have simply been bold variant glyphs of their regular weight, as in $\xi = (\xi_r, \xi_\varphi, \xi_\theta)$. Confusingly, the syntax in \LaTeX has been different for these two examples: `\mathbf{f}` in the former (‘ \mathbf{M} ’), and `\bm` (or `\boldsymbol`, deprecated) in the latter (‘ ξ ’).

In `unicode-math`, the `\mathbf{f}` command works directly with both Greek and Latin maths alphabet characters and depending on package option either switches to upright for Latin letters (`bold-style=TeX`) as well or keeps them italic (`bold-style=ISO`).

To match the package options for non-bold characters, for `bold-style=upright` all bold characters are upright, and `bold-style=literal` does not change the upright/italic shape of the letter.

Upright and italic bold mathematical letters input as direct unicode characters are normalised with the same rules. For example, with `bold-style=TeX`, a literal bold italic latin character will be typeset upright.

Note that `bold-style` is independent of `math-style`, although if the former is not specified then sensible defaults are chosen based on the latter.

The `bold-style` options’ effects are shown in brief in table 4.

Table 4: Effects of the `bold-style` package option.

Package option	Example	
	Latin	Greek
<code>bold-style=ISO</code>	$(\boldsymbol{a}, \boldsymbol{z}, \boldsymbol{B}, \boldsymbol{X})$	$(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\Gamma}, \boldsymbol{\Xi})$
<code>bold-style=TeX</code>	$(\boldsymbol{a}, \boldsymbol{z}, \boldsymbol{B}, \boldsymbol{X})$	$(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\Gamma}, \boldsymbol{\Xi})$
<code>bold-style=upright</code>	$(\boldsymbol{a}, \boldsymbol{z}, \boldsymbol{B}, \boldsymbol{X})$	$(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\Gamma}, \boldsymbol{\Xi})$

5.3 Sans serif style

Unicode contains upright and italic, medium and bold mathematical alphabet characters. These may be explicitly selected with the `\mathsfup`, `\mathsfit`, `\mathbfsup`, and `\mathbfsfit` commands discussed in section §5.4.

How should the generic `\mathsf` behave? Unlike bold, sans serif is used much more sparingly in mathematics. I’ve seen recommendations to typeset tensors in sans serif italic or sans serif italic bold (e.g., examples in the `isomath` and `mattens` packages). But L^AT_EX’s `\mathsf` is *upright* sans serif.

Therefore I reluctantly add the package options `[sans-style=upright]` and `[sans-style=italic]` to control the behaviour of `\mathsf`. The `upright` style sets up the command to use the seemingly-useless upright sans serif, including Greek; the `italic` style switches to using italic in both Latin and Greek alphabets. In other words, this option simply changes the meaning of `\mathsf` to either `\mathsfup` or `\mathsfit`, respectively. Please let me know if more granular control is necessary here.

There is also a `[sans-style=literal]` setting, set automatically with `[math-style=literal]`, which retains the uprightness of the input characters used when selecting the sans serif output.

5.3.1 What about bold sans serif?

While you might want your bold upright and your sans serif italic, I don’t believe you’d also want your bold sans serif upright (or all vice versa, if that’s even conceivable). Therefore, bold sans serif follows from the setting for sans serif; it is completely independent of the setting for bold.

In other words, `\mathbfsf` is `\mathbfsup` or `\mathbfsfit` based on `[sans-style=upright]` or `[sans-style=italic]`, respectively. And `[sans-style=literal]` causes `\mathbfsf` to retain the same italic or upright shape as the input, and turns it bold sans serif.

Note well! There is no medium-weight sans serif Greek alphabet in unicode; therefore, `\mathsf{\alpha}` does not make sense (simply produces ‘ α ’) while `\mathbfsf{\alpha}` gives ‘ α ’.

Table 5: Mathematical alphabets defined in unicode. Black dots indicate an alphabet exists in the font specified; grey dots indicate shapes that should always be taken from the upright font even in the italic style. See main text for description of `\mathbbit`.

Font				Alphabet		
Style	Shape	Series	Switch	Latin	Greek	Numerals
Serif	Upright	Normal	<code>\mathup</code>	•	•	•
		Bold	<code>\mathbfup</code>	•	•	•
	Italic	Normal	<code>\mathit</code>	•	•	•
		Bold	<code>\mathbfit</code>	•	•	•
Sans serif	Upright	Normal	<code>\mathsfup</code>	•		•
	Italic	Normal	<code>\mathsfit</code>	•		•
	Upright	Bold	<code>\mathsfbfup</code>	•	•	•
	Italic	Bold	<code>\mathsfbfit</code>	•	•	•
Typewriter	Upright	Normal	<code>\mathtt</code>	•		•
Double-struck	Upright	Normal	<code>\mathbb</code>	•		•
	Italic	Normal	<code>\mathbbit</code>	•		
Script	Upright	Normal	<code>\mathscr</code>	•		
		Bold	<code>\matbfscr</code>	•		
Fraktur	Upright	Normal	<code>\mathfrak</code>	•		
		Bold	<code>\mathbffrac</code>	•		

5.4 All (the rest) of the mathematical alphabets

Unicode contains separate codepoints for most if not all variations of alphabet shape one may wish to use in mathematical notation. The complete list is shown in table 5. Some of these have been covered in the previous sections.

At present, the math font switching commands do not nest; therefore if you want sans serif bold, you must write `\mathsfbf{...}` rather than `\mathbf{\mathsf{...}}`. This may change in the future.

5.4.1 Double-struck

The double-struck alphabet (also known as ‘blackboard bold’) consists of upright Latin letters $\{a-z, A-Z\}$, numerals $\mathbb{0}-\mathbb{9}$, summation symbol $\mathbb{\Sigma}$, and four Greek letters only: $\{\mathbb{V}, \mathbb{W}, \mathbb{F}, \mathbb{M}\}$.

While `\mathbb{\sum}` does produce a double-struck summation symbol, its limits aren’t properly aligned (see section §??). Therefore, either the literal character or the control sequence `\Bbbsum` are recommended instead.

There are also five Latin *italic* double-struck letters: $\mathbb{D}, \mathbb{d}, \mathbb{I}, \mathbb{i}, \mathbb{J}$. These can be accessed (if not with their literal characters or control sequences) with the `\mathbbit`

Table 6: The various forms of nabla.

Description		Glyph
Upright	Serif	∇
	Bold serif	∇
	Bold sans	∇
Italic	Serif	∇
	Bold serif	∇
	Bold sans	∇

alphabet switch, but note that only those five letters will give the expected output.

5.5 Miscellanea

5.5.1 Nabla

The symbol ∇ comes in the six forms shown in table 6. We want an individual option to specify whether we want upright or italic nabla by default (when either upright or italic nabla is used in the source). \TeX classically uses an upright nabla, but iso standards differ (I think). The package options `nabla=upright` and `nabla=italic` switch between the two choices. This is then inherited through `\mathbf`; `\mathit` and `\mathup` can be used to force one way or the other.

`nabla=italic` is implicit when using `math-style=ISO` and `nabla=upright` follows both `math-style=TeX` and `math-style=French`.

5.5.2 Partial

The same applies to the symbols U+2202: PARTIAL DIFFERENTIAL and U+1D715: MATH ITALIC PARTIAL DIFFERENTIAL.

At time of writing, both the Cambria Math and STIX fonts display these two glyphs in the same italic style, but this is hopefully a bug that will be corrected in the future — the ‘plain’ partial differential should really have an upright shape.

Use the `partial=upright` or `partial=italic` package options to specify which one you would like. The default is (always, unless someone requests and argues otherwise) `partial=italic`.¹

See table 7 for the variations on the partial differential symbol.

¹A good argument would revolve around some international standards body recommending upright over italic. I just don’t have the time right now to look it up.

Table 7: The various forms of the partial differential. Note that in the fonts used to display these glyphs, the first upright partial is incorrectly shown in an italic style.

Description		Glyph
Regular	Upright	∂
	Italic	∂
Bold	Upright	∂
	Italic	∂
Sans bold	Upright	∂
	Italic	∂

5.5.3 Epsilon and phi: ε vs. ϵ and φ vs. ϕ

\TeX defines `\epsilon` to look like ϵ and `\varepsilon` to look like ε . The Unicode glyph directly after delta and before zeta is ‘epsilon’ and looks like ε ; there is a subsequent variant of epsilon that looks like ϵ . This creates a problem. People who use unicode input won’t want their glyphs transforming; \TeX users will be confused that what they think as ‘normal epsilon’ is actual the ‘variant epsilon’. And the same problem exists for ‘phi’.

We have a package option to control this behaviour. With `\vargreek-shape=TeX`, `\phi` and `\epsilon` produce ϕ and ε and `\varphi` and `\varepsilon` produce ϕ and ϵ . With `\vargreek-shape=unicode`, these symbols are swapped. Note, however, that unicode characters are not affected by this option. That is, no remapping occurs of the characters/glyphs, only the control sequences.

The package default is to use `\vargreek-shape=TeX`.

U+3B5: GREEK SMALL LETTER EPSILON

U+3F5: GREEK LUNATE EPSILON SYMBOL

U+3C6: GREEK SMALL LETTER PHI

U+3D5: GREEK SMALL LETTER SCRIPT PHI

5.5.4 Primes

Primes (x') may be input in several ways. You may use any combination of `ascii` straight quote (`'`), `unicode` prime (`'`), and `\prime`; when multiple primes occur next to each other, they chain together to form double, triple, or quadruple primes if the font contains pre-drawn glyphs. These may also be accessed with `\primedouble`, `\primetriple`, and `\primequadruple`.

If the font does not contain the pre-drawn glyphs or more than four primes are used, the single prime glyph is used multiple times with a negative kern to get the spacing right. There is no user interface to adjust this negative kern yet (because I haven’t decided what it should look like); if you need to, write something

A 0 1 2 3 4 5 6 7 8 9 + - = () i n Z

Figure 1: The unicode superscripts supported as input characters. These are the literal glyphs from Charis SIL, not the output seen when used for maths input. The ‘A’ and ‘Z’ are to provide context for the size and location of the superscript glyphs.

A 0 1 2 3 4 5 6 7 8 9 + - = () a e i o r u v x β γ ρ ϕ χ Z

Figure 2: The unicode subscripts supported as input characters. See note from figure 1.

like this:

```
\ExplSyntaxOn
\muskip_gset:Nn \g_um_primekern_muskip { -\thinmuskip/2 }
\ExplSyntaxOff
```

5.5.5 Unicode subscripts and superscripts

You may, if you wish, use unicode subscripts and superscripts in your source document. For basic expressions, the use of these characters can make the input more readable. Adjacent sub- or super-scripts will be concatenated into a single expression.

The range of subscripts and superscripts supported by this package are shown in figures 1 and 2. Please request more if you think it is appropriate.

5.5.6 Colon

The colon is one of the few confusing characters of unicode maths. In \TeX , `:` is defined as a colon with relation spacing: ‘ $a : b$ ’. While `\colon` is defined as a colon with punctuation spacing: ‘ $a:b$ ’.

In unicode, `U+003A: COLON` is defined as a punctuation symbol, while `U+2236: RATIO` is the colon-like symbol used in mathematics to denote ratios and other things.

This breaks the usual straightforward mapping from control sequence to unicode input character to (the same) unicode glyph.

To preserve input compatibility, we remap the `ASCII` input character ‘`:`’ to `U+2236: RATIO`. Typing a literal `U+2236: RATIO` char will result in the same output. If `amsmath` is loaded, then the definition of `\colon` is inherited from there (it looks like a punctuation colon with additional space around it). Otherwise, `\colon` is made to output a colon with `\mathpunct` spacing.

Table 8: Slashes and backslashes.

Slot	Name	Glyph	Command
U+002F	SOLIDUS	/	<code>\solidus</code>
U+2044	FRACTION SLASH	/	<code>\fracslash</code>
U+2215	DIVISION SLASH	/	<code>\slash</code>
U+29F8	BIG SOLIDUS	/	<code>\xsol</code>
U+005C	REVERSE SOLIDUS	\	<code>\backslash</code>
U+2216	SET MINUS	\	<code>\smallsetminus</code>
U+29F5	REVERSE SOLIDUS OPERATOR	\	<code>\setminus</code>
U+29F9	BIG REVERSE SOLIDUS	\	<code>\xbsol</code>

The package option `[colon=literal]` forces ASCII input ‘:’ to be printed as `\mathcolon` instead.

5.5.7 Slashes and backslashes

There are several slash-like symbols defined in unicode. The complete list is shown in table 8.

In regular L^AT_EX we can write `\left\slash...\right\backslash` and so on and obtain extensible delimiter-like symbols. Not all of the unicode slashes are suitable for this (and do not have the font support to do it).

Slash Of U+2044: FRACTION SLASH, TR25 says that it is:

...used to build up simple fractions in running text...however parsers of mathematical texts should be prepared to handle fraction slash when it is received from other sources.

U+2215: DIVISION SLASH should be used when division is represented without a built-up fraction; $\pi \approx 22/7$, for example.

U+29F8: BIG SOLIDUS is a ‘big operator’ (like Σ).

Backslash The U+005C: REVERSE SOLIDUS character `\backslash` is used for denoting double cosets: $A \backslash B$. (So I’m led to believe.) It may be used as a ‘stretchy’ delimiter if supported by the font.

MathML uses U+2216: SET MINUS like this: $A \setminus B$.² The L^AT_EX command name `\smallsetminus` is used for backwards compatibility.

²§4.4.5.11 <http://www.w3.org/TR/2000/REC-MATHML-20001023/>

Presumably, U+29F5: REVERSE SOLIDUS OPERATOR is intended to be used in a similar way, but it could also (perhaps?) be used to represent ‘inverse division’: $\pi \approx 7 \setminus 22$.³ The L^AT_EX name for this character is `\setminus`.

Finally, U+29F9: BIG REVERSE SOLIDUS is a ‘big operator’ (like Σ).

How to use all of these things Unfortunately, font support for the above characters/glyphs is rather inconsistent. In Cambria Math, the only slash that grows (say when writing

$$\left[\begin{array}{cc} a & b \\ c & d \end{array} \right] / \left[\begin{array}{cc} 1 & 1 \\ 1 & 0 \end{array} \right])$$

is the FRACTION SLASH, which we just established above is sort of only supposed to be used in text.

Of the above characters, the following are allowed to be used after `\left`, `\middle`, and `\right`:

- `\solidus`;
- `\fracslash`;
- `\slash`; and,
- `\backslash` (the only reverse slash).

However, we assume that there is only *one* stretchy slash in the font; this is assumed by default to be U+002F: SOLIDUS. Writing `\left/` or `\left\slash` or `\leftfracslash` will all result in the same stretchy delimiter being used.

The delimiter used can be changed with the `slash-delimiter` package option. Allowed values are `ascii`, `frac`, and `div`, corresponding to the respective unicode slots.

For example: as mentioned above, Cambria Math’s stretchy slash is U+2044: FRACTION SLASH. When using Cambria Math, then `unicode-math` should be loaded with the `[slash-delimiter=frac]` option. (This should be a font option rather than a package option, but it will change soon.)

5.5.8 Normalising some input characters

I believe all variant forms should be used as legal input that is normalised to a consistent output glyph, because we want to be fault-tolerant in the input. Here are the duplicates:

U+2511: LATIN SMALL LETTER ALPHA

U+25B: LATIN SMALL LETTER EPSILON

³This is valid syntax in the Octave and Matlab programming languages, in which it means matrix inverse pre-multiplication. I.e., $A \setminus B \equiv A^{-1}B$.

U+263: LATIN SMALL LETTER GAMMA
 U+269: LATIN SMALL LETTER IOTA
 U+278: LATIN SMALL LETTER PHI
 U+28A: LATIN SMALL LETTER UPSILON
 U+190: LATIN CAPITAL LETTER EPSILON
 U+194: LATIN CAPITAL LETTER GAMMA
 U+196: LATIN CAPITAL LETTER IOTA
 U+1B1: LATIN CAPITAL LETTER UPSILON

(Not yet implemented.)

File I

The unicode-math package

This is the package.

```
1 \ProvidesPackage{unicode-math}
2 [2009/10/11 v0.4 Unicode maths in XeLaTeX]
```

6 Things we need

Packages

```
3 \RequirePackage{expl3}[2009/08/12]
4 \RequirePackage{xparse}[2009/08/31]
5 \RequirePackage{fontspec}
   Start using LATEX3 — finally!
6 \ExplSyntaxOn
```

Counters and conditionals

```
7 \newcounter{um@fam}
8 \newif\if@um@fontspec@feature
9 \newif\if@um@ot@math@
10 \bool_new:N \l_um_init_bool
```

For math-style:

```
11 \newif\if@um@literal
12 \newif\if@um@upGreek
13 \newif\if@um@upgreek
14 \newif\if@um@upLatin
15 \newif\if@um@uplatin
```

For bold-style:

```
16 \newif\if@um@bfliteral
17 \newif\if@um@bfupGreek
18 \newif\if@um@bfupgreek
19 \newif\if@um@bfupLatin
20 \newif\if@um@bfuplatin
```

For nabla:

```
21 \newif\if@um@upNabla
22 \newif\if@um@uppartial
23 \bool_new:N \g_um_texgreek_bool
```

6.0.9 Alphabet unicode positions

Before we begin, let's define the positions of the various unicode alphabets so that our code is a little more readable.⁴

```
24 \def\um@usv@upnum{48}
25 \def\um@usv@upLatin{65}
26 \def\um@usv@uplatin{97}
27 \def\um@usv@upGreek{"391}
28 \def\um@usv@upgreek{"3B1}
29 \def\um@usv@itLatin{"1D434}
30 \def\um@usv@itlatin{"1D44E}
31 \def\um@usv@itGreek{"1D6E2}
32 \def\um@usv@itgreek{"1D6FC}
33 \def\um@usv@bbnum{"1D7D8}
34 \def\um@usv@bbLatin{"1D538}
35 \def\um@usv@bblatin{"1D552}
36 \def\um@usv@scrLatin{"1D49C}
37 \def\um@usv@scrlatin{"1D4B6}
38 \def\um@usv@frakLatin{"1D504}
39 \def\um@usv@fraklatin{"1D51E}
40 \def\um@usv@sfnum{"1D7E2}
41 \def\um@usv@sfupnum{"1D7E2}
42 \def\um@usv@sfitnum{"1D7E2}
43 \def\um@usv@sfupLatin{"1D5A0}
44 \def\um@usv@sflatin {"1D5A0}
45 \def\um@usv@sfuplatin{"1D5BA}
46 \def\um@usv@sflatin{"1D5BA}
47 \def\um@usv@sfitLatin{"1D608}
48 \def\um@usv@sfitlatin{"1D622}
49 \def\um@usv@ttnum{"1D7F6}
50 \def\um@usv@ttLatin{"1D670}
51 \def\um@usv@ttlLatin{"1D68A}
```

⁴'u.s.v.' stands for 'unicode scalar value'.

Bold:

```
52 \def\um@usv@bfnum {"1D7CE}
53 \def\um@usv@bfupnum{"1D7CE}
54 \def\um@usv@bfitnum{"1D7CE}
55 \def\um@usv@bfLatin {"1D400}
56 \def\um@usv@bflatin {"1D41A}
57 \def\um@usv@bfGreek {"1D6A8}
58 \def\um@usv@bfgreek {"1D6C2}
59 \def\um@usv@bfupLatin{"1D400}
60 \def\um@usv@bfuplatin{"1D41A}
61 \def\um@usv@bfupGreek{"1D6A8}
62 \def\um@usv@bfupgreek{"1D6C2}
63 \def\um@usv@bfitLatin{"1D468}
64 \def\um@usv@bfitlatin{"1D482}
65 \def\um@usv@bfitGreek{"1D71C}
66 \def\um@usv@bfitgreek{"1D736}
67 \def\um@usv@bffrakLatin{"1D56C}
68 \def\um@usv@bffraklatin{"1D586}
69 \def\um@usv@bfscrLatin{"1D4D0}
70 \def\um@usv@bfscrlatin{"1D4EA}
71 \def\um@usv@bfsfnum {"1D7EC}
72 \def\um@usv@bfsfupnum{"1D7EC}
73 \def\um@usv@bfsfitnum{"1D7EC}
74 \def\um@usv@bfsfupLatin{"1D5D4}
75 \def\um@usv@bfsfuplatin{"1D5EE}
76 \def\um@usv@bfsfupGreek{"1D756}
77 \def\um@usv@bfsfupgreek{"1D770}
78 \def\um@usv@bfsfitLatin{"1D63C}
79 \def\um@usv@bfsfitlatin{"1D656}
80 \def\um@usv@bfsfitGreek{"1D790}
81 \def\um@usv@bfsfitgreek{"1D7AA}
82 \def\um@usv@bfsfLatin {\if@um@upLatin\um@usv@bfsfupLatin\else\um@usv@bfsfitLatin\fi}
83 \def\um@usv@bfsflatin {\if@um@uplatin\um@usv@bfsfuplatin\else\um@usv@bfsfitlatin\fi}
84 \def\um@usv@bfsfGreek {\if@um@upGreek\um@usv@bfsfupGreek\else\um@usv@bfsfitGreek\fi}
85 \def\um@usv@bfsfgreek {\if@um@upgreek\um@usv@bfsfupgreek\else\um@usv@bfsfitgreek\fi}
```

Greek variants:

```
86 \def\um@usv@varTheta{"3F4}
87 \def\um@usv@Digamma{"3DC}
88 \def\um@usv@varepsilon{"3F5}
89 \def\um@usv@vartheta{"3D1}
90 \def\um@usv@varkappa{"3F0}
91 \def\um@usv@varphi{"3D5}
92 \def\um@usv@varrho{"3F1}
93 \def\um@usv@varpi{"3D6}
94 \def\um@usv@digamma{"3DD}
```

Bold:

```
95 \def\um@usv@bfvarTheta{"1D6B9}  
96 \def\um@usv@bfDigamma{"1D7CA}  
97 \def\um@usv@bfvarepsilon{"1D6DC}  
98 \def\um@usv@bfvartheta{"1D6DD}  
99 \def\um@usv@bfvarkappa{"1D6DE}  
100 \def\um@usv@bfvarphi{"1D6DF}  
101 \def\um@usv@bfvarrho{"1D6E0}  
102 \def\um@usv@bfvarpi{"1D6E1}  
103 \def\um@usv@bfdigamma{"1D7CB}
```

Italic Greek variants:

```
104 \def\um@usv@ith{"210E}  
105 \def\um@usv@itvarTheta{"1D6F3}  
106 \def\um@usv@itvarepsilon{"1D716}  
107 \def\um@usv@itvartheta{"1D717}  
108 \def\um@usv@itvarkappa{"1D718}  
109 \def\um@usv@itvarphi{"1D719}  
110 \def\um@usv@itvarrho{"1D71A}  
111 \def\um@usv@itvarpi{"1D71B}
```

Bold italic:

```
112 \def\um@usv@bfuph{"1D421}  
113 \def\um@usv@bfith{"1D489}  
114 \def\um@usv@bfitvarTheta{"1D72D}  
115 \def\um@usv@bfitvarepsilon{"1D750}  
116 \def\um@usv@bfitvartheta{"1D751}  
117 \def\um@usv@bfitvarkappa{"1D752}  
118 \def\um@usv@bfitvarphi{"1D753}  
119 \def\um@usv@bfitvarrho{"1D754}  
120 \def\um@usv@bfitvarpi{"1D755}
```

Bold sans:

```
121 \def\um@usv@bfsfupvarTheta{"1D767}  
122 \def\um@usv@bfsfupvarepsilon{"1D78A}  
123 \def\um@usv@bfsfupvartheta{"1D78B}  
124 \def\um@usv@bfsfupvarkappa{"1D78C}  
125 \def\um@usv@bfsfupvarphi{"1D78D}  
126 \def\um@usv@bfsfupvarrho{"1D78E}  
127 \def\um@usv@bfsfupvarpi{"1D78F}
```

Bold sans italic:

```
128 \def\um@usv@bfsfitvarTheta{"1D7A1}  
129 \def\um@usv@bfsfitvarepsilon{"1D7C4}  
130 \def\um@usv@bfsfitvartheta{"1D7C5}  
131 \def\um@usv@bfsfitvarkappa{"1D7C6}  
132 \def\um@usv@bfsfitvarphi{"1D7C7}  
133 \def\um@usv@bfsfitvarrho{"1D7C8}
```

```
134 \def\um@usv@bfsfitvarpi{"1D7C9}
```

Nabla:

```
135 \def\um@usv@Nabla{"2207}
136 \def\um@usv@itNabla{"1D6FB}
137 \def\um@usv@bfNabla{"1D6C1}
138 \def\um@usv@bfitNabla{"1D735}
139 \def\um@usv@bfsfNabla{"1D76F}
140 \def\um@usv@bfsfitNabla{"1D7A9}
```

Partial:

```
141 \def\um@usv@partial{"2202}
142 \def\um@usv@itpartial{"1D715}
143 \def\um@usv@bfpartial{"1D6DB}
144 \def\um@usv@bfitpartial{"1D74F}
145 \def\um@usv@bfsfpartial{"1D789}
146 \def\um@usv@bfsfitpartial{"1D7C3}
```

6.1 Options

xkeyval's package support is used here. I'll switch over to l3keys2e at some stage.

`\unimathsetup` This macro can be used in lieu of or later to override options declared when the package is loaded.

```
147 \DeclareDocumentCommand \unimathsetup {m} {
148   \setkeys{unicode-math.sty}{#1}
149 }
```

math-style

```
150 \define@choicekey*{unicode-math.sty}
151   {math-style}[\@tempa\@tempb]{iso,tex,french,upright,literal}{
152   \ifcase\@tempb\relax
153     \@um@upGreekfalse
154     \@um@upgreekfalse
155     \@um@upLatinfalse
156     \@um@uplatinfalse
157     \@um@bfupGreekfalse
158     \@um@bfupgreekfalse
159     \@um@uppartialfalse
160     \@um@bfupLatinfalse
161     \@um@bfuplatinfalse
162     \@um@upNablafalse
163     \bool_set_false:N \g_um_upsans_bool
164     \bool_set_false:N \g_um_texgreek_bool
165   \or
166     \@um@upGreektrue
```

```

167 \um@upgreekfalse
168 \um@upLatinfalse
169 \um@uplatinfalse
170 \um@bfupGreektrue
171 \um@bfupgreekfalse
172 \um@uppartialfalse
173 \um@bfupLatintrue
174 \um@bfuplatintrue
175 \um@upNablatrue
176 \bool_set_true:N \g_um_upsans_bool
177 \bool_set_false:N \g_um_texgreek_bool
178 \or
179 \um@upGreektrue
180 \um@upgreektrue
181 \um@upLatintrue
182 \um@uplatinfalse
183 \um@bfupGreektrue
184 \um@bfupgreektrue
185 \um@uppartialtrue
186 \um@bfupLatintrue
187 \um@bfuplatintrue
188 \um@upNablatrue
189 \bool_set_true:N \g_um_upsans_bool
190 \bool_set_false:N \g_um_texgreek_bool
191 \or
192 \um@upGreektrue
193 \um@upgreektrue
194 \um@upLatintrue
195 \um@uplatintrue
196 \um@bfupGreektrue
197 \um@bfupgreektrue
198 \um@uppartialtrue
199 \um@bfupLatintrue
200 \um@bfuplatintrue
201 \um@upNablatrue
202 \bool_set_true:N \g_um_upsans_bool
203 \bool_set_false:N \g_um_texgreek_bool
204 \or
205 \um@literaltrue
206 \um@bfliteraltrue
207 \bool_set_true:N \g_um_sfliteral_bool
208 \bool_set_false:N \g_um_texgreek_bool
209 \fi
210 }

```

\um_setup_shapes:

```

211 \cs_set:Nn \um_setup_shapes: {

```

```

212 \tl_set:Nx \um_Greek_up_or_it_usv { \if@um@upGreek \um@usv@upGreek \else \um@usv@itGreek
213 \tl_set:Nx \um_varTheta_up_or_it_usv { \if@um@upGreek \um@usv@varTheta \else \um@usv@itvarTh
214 \tl_set:Nx \um_greek_up_or_it_usv { \if@um@upgreek \um@usv@upgreek \else \um@usv@itgreek \fi
215 \tl_set:Nx \um_Latin_up_or_it_usv { \if@um@upLatin \um@usv@upLatin \else \um@usv@itLatin \fi
216 \tl_set:Nx \um_latin_up_or_it_usv { \if@um@uplatin \um@usv@uplatin \else \um@usv@itlatin \fi
217 \tl_set:Nx \um_latinh_up_or_it_usv { \if@um@uplatin \number`\h \else \um@usv@ith \
218 }

```

bold-style

```

219 \define@choicekey*{unicode-math.sty}{bold-style}[\@tempa\@tempb]{iso,tex,upright,literal}{
220 \ifcase\@tempb\relax
221 \um@bfupGreekfalse
222 \um@bfupgreekfalse
223 \um@bfupLatinfalse
224 \um@bfuplatinfalse
225 \um@uppartialfalse
226 \or
227 \um@bfupGreektrue
228 \um@bfupgreekfalse
229 \um@bfupLatintrue
230 \um@bfuplatintrue
231 \um@uppartialfalse
232 \or
233 \um@bfupGreektrue
234 \um@bfupgreektrue
235 \um@bfupLatintrue
236 \um@bfuplatintrue
237 \um@uppartialtrue
238 \or
239 \um@bfliteraltrue
240 \fi
241 }

```

\um_setup_bfshapes:

```

242 \cs_set:Nn \um_setup_bfshapes: {
243 \tl_set:Nx \um_bf_Greek_up_or_it_usv { \if@um@bfupGreek \um@usv@bfupGreek \else \um@usv@bfit
244 \tl_set:Nx \um_bf_greek_up_or_it_usv { \if@um@bfupgreek \um@usv@bfupgreek \else \um@usv@bfit
245 \tl_set:Nx \um_bf_Latin_up_or_it_usv { \if@um@bfupLatin \um@usv@bfupLatin \else \um@usv@bfit
246 \tl_set:Nx \um_bf_latin_up_or_it_usv { \if@um@bfuplatin \um@usv@bfuplatin \else \um@usv@bfit
247 \tl_set:Nx \um_bf_varTheta_up_or_it_usv { \if@um@bfupGreek \um@usv@bfvarTheta \else \um@usv@
248 }

```

sans-style

```

249 \bool_new:N \g_um_upsans_bool
250 \bool_new:N \g_um_sf literal_bool

```

```

251 \define@choicekey*{unicode-math.sty}
252   {sans-style}[\@tempa\@tempb]{italic,upright,literal}{
253   \ifcase\@tempb\relax
254     \bool_set_false:N \g_um_upsans_bool
255   \or
256     \bool_set_true:N \g_um_upsans_bool
257   \or
258     \bool_set_true:N \g_um_sfliteral_bool
259   \fi
260 }

```

\um_setup_sfshapes:

```

261 \cs_set:Nn \um_setup_sfshapes: {
262   \bool_if:NTF \g_um_upsans_bool {
263     \tl_set:Nn \um_sf_Latin_up_or_it_usv { \um@usv@sflatin }
264     \tl_set:Nn \um_sf_latin_up_or_it_usv { \um@usv@sflatin }
265     \tl_set:Nn \um_bfsf_Latin_up_or_it_usv { \um@usv@bfsfuplatin }
266     \tl_set:Nn \um_bfsf_latin_up_or_it_usv { \um@usv@bfsfuplatin }
267     \tl_set:Nn \um_bfsf_Greek_up_or_it_usv { \um@usv@bfsfupgreek }
268     \tl_set:Nn \um_bfsf_greek_up_or_it_usv { \um@usv@bfsfupgreek }
269     \tl_set:Nn \um_bfsf_varTheta_up_or_it_usv { \um@usv@bfsfupvarTheta }
270   }{
271     \tl_set:Nn \um_sf_Latin_up_or_it_usv { \um@usv@sffitLatin }
272     \tl_set:Nn \um_sf_latin_up_or_it_usv { \um@usv@sffitlatin }
273     \tl_set:Nn \um_bfsf_Latin_up_or_it_usv { \um@usv@bfsffitLatin }
274     \tl_set:Nn \um_bfsf_latin_up_or_it_usv { \um@usv@bfsffitlatin }
275     \tl_set:Nn \um_bfsf_Greek_up_or_it_usv { \um@usv@bfsffitGreek }
276     \tl_set:Nn \um_bfsf_greek_up_or_it_usv { \um@usv@bfsffitgreek }
277     \tl_set:Nn \um_bfsf_varTheta_up_or_it_usv { \um@usv@bfsffitvarTheta }
278   }
279 }

```

Symbol obliqueness

```

280 \define@choicekey*{unicode-math.sty}{nabla}[\@tempa\@tempb]{upright,italic}{
281   \ifcase\@tempb\relax
282     \@um@upNablatrue
283   \or
284     \@um@upNablafalse
285   \fi
286 }
287 \cs_set:Nn \um_setup_nabla: {
288   \if@um@upNabla
289     \tl_set:Nn \um_Nabla_up_or_it_usv { \um@usv@Nabla }
290     \tl_set:Nn \um_bfNabla_up_or_it_usv { \um@usv@bfNabla }
291     \tl_set:Nn \um_bfsfNabla_up_or_it_usv { \um@usv@bfsfNabla }
292   \else

```

```

293 \tl_set:Nn \um_Nabla_up_or_it_usv { \um@usv@itNabla }
294 \tl_set:Nn \um_bfNabla_up_or_it_usv { \um@usv@bfitNabla }
295 \tl_set:Nn \um_bfsfNabla_up_or_it_usv { \um@usv@bfsfitNabla }
296 \fi
297 }
298 \define@choicekey*{unicode-math.sty}{partial}[\@tempa\@tempb]{upright,italic}{
299 \ifcase\@tempb\relax
300 \um@uppartialtrue
301 \or
302 \um@uppartialfalse
303 \fi
304 }
305 \cs_set:Nn \um_setup_partial: {
306 \if@um@uppartial
307 \tl_set:Nn \um_partial_up_or_it_usv { \um@usv@partial }
308 \tl_set:Nn \um_bfpartial_up_or_it_usv { \um@usv@bfpartial }
309 \tl_set:Nn \um_bfsfpartial_up_or_it_usv { \um@usv@bfsfpartial }
310 \else
311 \tl_set:Nn \um_partial_up_or_it_usv { \um@usv@itpartial }
312 \tl_set:Nn \um_bfpartial_up_or_it_usv { \um@usv@bfitpartial }
313 \tl_set:Nn \um_bfsfpartial_up_or_it_usv { \um@usv@bfsfitpartial }
314 \fi
315 }

```

Epsilon and phi shapes

```

316 \define@choicekey*{unicode-math.sty}{vargreek-shape}[\@tempa\@tempb]{unicode,TeX}{
317 \ifcase\@tempb\relax
318 \bool_set_false:N \g_um_texgreek_bool
319 \or
320 \bool_set_true:N \g_um_texgreek_bool
321 \fi
322 }

```

Colon style

```

323 \bool_new:N \g_um_literal_colon_bool
324 \define@choicekey*{unicode-math.sty}{colon}[\@tempa\@tempb]{literal,TeX}{
325 \ifcase\@tempb\relax
326 \bool_set_true:N \g_um_literal_colon_bool
327 \or
328 \bool_set_false:N \g_um_literal_colon_bool
329 \fi
330 }

```

Slash delimiter style

```

331 \define@choicekey*{unicode-math.sty}{slash-delimiter}[\@tempa\@tempb]{ascii,frac,div}{
332   \ifcase\@tempb\relax
333     \tl_set:Nn \g_um_slash_delimiter_usv {"002F}
334   \or
335     \tl_set:Nn \g_um_slash_delimiter_usv {"2044}
336   \or
337     \tl_set:Nn \g_um_slash_delimiter_usv {"2215}
338   \fi
339 }
340 \ExecuteOptionsX{math-style=TeX,slash-delimiter=ascii}
341 \ProcessOptionsX

```

6.2 Overcoming \@onlypreamble

The requirement of only setting up the maths fonts in the preamble is now removed. The following list might be overly ambitious.

```

342 \tl_map_inline:nn {
343   \new@mathgroup
344   \cdp@list
345   \cdp@elt
346   \DeclareMathSizes
347   \@DeclareMathSizes
348   \newmathalphabet
349   \newmathalphabet@@
350   \newmathalphabet@@@
351   \DeclareMathVersion
352   \define@mathalphabet
353   \define@mathgroup
354   \addtoversion
355   \version@list
356   \version@elt
357   \alpha@list
358   \alpha@elt
359   \restore@mathversion
360   \init@restore@version
361   \dorestore@version
362   \process@table
363   \new@mathversion
364   \DeclareSymbolFont
365   \group@list
366   \group@elt
367   \new@symbolfont
368   \SetSymbolFont
369   \SetSymbolFont@
370   \get@cdp
371   \DeclareMathAlphabet

```



```

372 \new@mathalphabet
373 \SetMathAlphabet
374 \SetMathAlphabet@
375 \DeclareMathAccent
376 \set@mathaccent
377 \DeclareMathSymbol
378 \set@mathchar
379 \set@mathsymbol
380 \DeclareMathDelimiter
381 \@xxDeclareMathDelimiter
382 \@DeclareMathDelimiter
383 \@xDeclareMathDelimiter
384 \set@mathdelimiter
385 \set@@mathdelimiter
386 \DeclareMathRadical
387 \mathchar@type
388 \DeclareSymbolFontAlphabet
389 \DeclareSymbolFontAlphabet@
390 }{
391 \tl_remove_in:Nn \@preamblecmds {\do#1}
392 }

```

6.3 Other things

`\um@fontdimen@percent` #1 : Font dimen number

`\fontdimens` 10, 11, and 65 aren't actually dimensions, they're percentage values given in units of sp. This macro takes a font dimension number and outputs the decimal value of the associated parameter.

0.73	<code>\font\tmpfont="Cambria Math"</code>
0.60	<code>\um@fontdimen@percent{10}{\tmpfont}\</code>
0.65	<code>\um@fontdimen@percent{11}{\tmpfont}\</code>
	<code>\um@fontdimen@percent{65}{\tmpfont}</code>

```

393 \def\um@fontdimen@percent#1#2{
394   0.\strip@pt\dimexpr\fontdimen#1#2 *65536\relax
395 }

```

`\um@scaled@apply` #1 : A math style

#2 : Macro that takes a non-delimited length argument (like `\kern`)

#3 : Length control sequence to be scaled according to the math style

This macro is used to scale the lengths reported by `\fontdimen` according to the scale factor for script- and scriptscript-size objects.

```

396 \def\um@scaled@apply#1#2#3{
397   \ifx#1\scriptstyle
398     #2\um@fontdimen@percent{10}\um@font#3

```

```

399 \else
400   \ifx#1\scriptscriptstyle
401     #2\um@fontdimen@percent{11}\um@font#3
402   \else
403     #2#3%
404   \fi
405 \fi
406 }

```

7 Fundamentals

7.1 Enlarging the number of maths families

To start with, we've got a power of two as many `\fams` as before. So (from `ltxssbas.dtx`) we want to redefine

```

407 \def\new@mathgroup{\alloc@8\mathgroup\chardef\@cclvi}
408 \let\newfam\new@mathgroup

```

This is sufficient for L^AT_EX's `\DeclareSymbolFont`-type commands to be able to define 256 named maths fonts. Now we need a new `\DeclareMathSymbol`.

7.2 `\DeclareMathSymbol` for unicode ranges

This command is a bit funny at the moment; it doesn't define the actual macro for almost all of the symbols passed to it, but it does assign the `\XeTeXmathchar`.

```

\um@mathsymbol #1 : Symbol, e.g., \alpha
                #2 : Type, e.g., \mathalpha
                #3 : Math font name, e.g., operators
                #4 : Slot, e.g., "221E
409 \def \um@mathsymbol#1#2#3#4{
410   \expandafter\um@set@mathsymbol\csname sym#3\endcsname#1#2{#4}}

```

The final macros that actually define the maths symbol with X_eT_EX primitives.

```

\um@set@mathsymbol #1 : Symbol font number
                   #2 : Symbol macro, e.g., \alpha
                   #3 : Type, e.g., \mathalpha
                   #4 : Slot, e.g., "221E
If the symbol definition is for a macro. There are a bunch of tests to perform to
process the various characters.
411 \def\um@set@mathsymbol#1#2#3#4{

```

Operators In the examples following, say we're defining for the symbol $\sum(\Sigma)$.

```
412 \ifx\mathop#3\relax
```

In order for literal unicode characters to be used in the source and still have the correct limits behaviour, big operators are made math-active.

The active math char is defined to expand to the macro `\sumop`.

```
413 \begingroup
414   \char_make_active:n {#4}
415   \global\mathcode#4="8000\relax
416   \um@scanactivedef #4 \@nil { \csname\cs_to_str:N #2 op\endcsname }
417 \endgroup
```

Some of these require a `\nolimits` suffix. This is controlled by the `\um@nolimits` macro, which contains a list of such characters. This list is checked dynamically because we're not interested in efficiency. Or something. This allows the list to be updated in the middle of a document.

Declare the plain old `\mathchardef` for the control sequence `\sum@sym`.

```
418 \expandafter\global\expandafter\XeTeXmathchardef
419   \csname\string#2@sym\endcsname
420   ="\mathchar@type#3 #1 #4\relax
```

Now define `\sumop` as `\sum@sym`, followed by `\nolimits` if necessary.

```
421 \cs_gset:cpn { \cs_to_str:N #2 op } {
422   \csname\string#2@sym\endcsname
423   \expandafter\in@\expandafter#2\expandafter{\um@nolimits}
424   \ifin@
425     \expandafter\nolimits
426   \fi
427 }
```

Don't forget that the actual `\sum` macro is simply defined in terms of the literal unicode symbol!

```
428 \else
```

Radicals Needs to be before the delimiters because the radical is, for some reason, `\mathopen`.

```
429 \expandafter\in@\expandafter#2\expandafter{\um@radicals,}
430 \ifin@
431   \cs_gset:cpn { \cs_to_str:N #2 sign } { \XeTeXradical #1 #4 \relax }
432 \else
```

Delimiters TODO: sort out which of these three declarations are necessary! (Definitely the first, to work with `\left/\right.`.)

```
433 \ifx\mathopen#3\relax
434   \cs_gset:Npn #2 {\XeTeXdelimiter "\mathchar@type#3 #1 #4\relax}
435   \global\XeTeXdelcode#4=#1 #4\relax
```

```

436     \global\XeTeXmathcode#4="\mathchar@type#3 #1 #4\relax
437 \else
438     \ifx\mathclose#3\relax
439         \cs_gset:Npn #2 {\XeTeXdelimiter "\mathchar@type#3 #1 #4\relax}
440         \global\XeTeXdelcode#4=#1 #4\relax
441         \global\XeTeXmathcode#4="\mathchar@type#3 #1 #4\relax
442     \else

```

Accents

```

443     \ifx\mathaccent#3\relax
444     \cs_gset:Npx #2 {\XeTeXmathaccent "\mathchar@type#3 #1 #4\relax}
445     \else

```

And finally, the general case. We define the unicode mathcode for the character. The macro is defined generically in terms of the unicode character.

```

446         \global\XeTeXmathcode#4="\mathchar@type#3 #1 #4\relax
447     \fi
448 \fi
449 \fi
450 \fi
451 \fi
452 }

```

`\um_set_mathcode:nnnn` [For later] or if it's for a character code (just a wrapper around the primitive). Note that this declaration *isn't* global so that it can be constrained by grouping inside math alphabet switches.

```

453 \cs_set:Nn \um_set_mathcode:nnnn {
454     \XeTeXmathcode#1="\mathchar@type#2 \csname sym#3\endcsname #4\relax
455 }

```

7.3 The main `\setmathfont` macro

Using a range including large character sets such as `\mathrel`, `\mathalpha`, *etc.*, is *very slow*! I hope to improve the performance somehow.

`\setmathfont` [**#1**]: font features
#2 : font name

```

456 \DeclareDocumentCommand \setmathfont { O{ } m } {

```

- Erase any conception \LaTeX has of previously defined math symbol fonts; this allows `\DeclareSymbolFont` at any point in the document.

```

457     \let\glb@currsizel\relax

```

- To start with, assume we're defining the font for every math symbol character.

```

458      \bool_set_true:N \l_um_init_bool
459      \seq_clear:N \l_um_char_range_seq
460      \let\um@char@num@range\@empty

```

- Tell fontspec that maths font features are actually allowed.

```

461      \@um@fontspec@featuretrue

```

- Grab the current size information (is this robust enough? Maybe it should be preceded by `\normalsize`).

```

462      \csname S@\f@size\endcsname

```

- Set the name of the math version being defined. (obviously more needs to be done here!)

```

463      \def\um@mversion{normal}
464      \DeclareMathVersion{\um@mversion}

```

Define default font features for the script and scriptscript font.

```

465      \tl_set:Nn \l_um_script_features_tl {ScriptStyle}
466      \tl_set:Nn \l_um_sscript_features_tl {ScriptScriptStyle}
467      \tl_set:Nn \l_um_script_font_tl      {#2}
468      \tl_set:Nn \l_um_sscript_font_tl    {#2}

```

Use fontspec to select a font to use. The macro `\S@{size}` contains the definitions of the sizes used for maths letters, subscripts and subsubscripts in `\tf@size`, `\sf@size`, and `\ssf@size`, respectively.

```

469      \setkeys*{unicode-math.sty}{#1}
470      \cs_set:Npx \um_tmp: {
471        \exp_not:N \setkeys*[um]{options}{\exp_not:N \XKV@rm}
472      }
473      \um_tmp:
474      \cs_set:Npx \um_tmp: {
475        \exp_not:N \zf@fontspec {
476          BoldFont = {}, ItalicFont = {},
477          Script = Math,
478          SizeFeatures = {
479            {Size = \tf@size-} ,
480            {Size = \sf@size-\tf@size ,
481              Font = \l_um_script_font_tl ,
482              \l_um_script_features_tl
483            } ,
484            {Size = -\sf@size ,

```

```

485         Font = \l_um_sscript_font_tl ,
486         \l_um_sscript_features_tl
487     }
488 },
489 \XKV@rm
490 }{#2}
491 }
492 \um_tmp:

```

Check for the correct number of \fontdimens:

```

493 \font\um@font="#2"\relax
494 %% \ifdim \dimexpr\fontdimen9\um@font*65536\relax =65pt\relax
495 %% \@um@ot@math@true
496 %% \else
497 %% \PackageWarningNoLine{unicode-math}{
498 %%     The~ font~ '#2' ~is~ not~ a~ valid~ OpenType~ maths~ font.~
499 %%     Some~ maths~ features~ will~ not~ be~ available~ or~ behave~
500 %%     in~ a~ substandard~ manner
501 %% }
502 %% \fi

```

If we're defining the full unicode math repertoire, then we skip all the parsing processing needed if we're only defining a subset.

- Math symbols are defined with \UnicodeMathSymbol; see section §7.3.1 for the individual definitions

```

503 \bool_if:NTF \l_um_init_bool {
504     \tl_set:Nn \um_symfont_tl {um@allsym}
505     \PackageInfo{unicode-math}{Defining~ the~ default~ maths~ font~ as~ '#2'}
506     \cs_set_eq:NN \UnicodeMathSymbol \um_process_symbol_noparse:nnnn
507     \cs_set_eq:NN \um_mathmap:Nnn \um_mathmap_noparse:Nnn
508     \cs_set_eq:NN \um_remap_symbol:nnn \um_remap_symbol_noparse:nnn
509     \cs_set_eq:NN \um_maybe_init_alphabet:n \um_init_alphabet:n
510 }{
511     \stepcounter{um@fam}
512     \tl_set:Nx \um_symfont_tl {um@fam\theum@fam}
513     \cs_set_eq:NN \UnicodeMathSymbol \um_process_symbol_parse:nnnn
514     \cs_set_eq:NN \um_mathmap:Nnn \um_mathmap_parse:Nnn
515     \cs_set_eq:NN \um_remap_symbol:nnn \um_remap_symbol_parse:nnn
516     \cs_set_eq:NN \um_maybe_init_alphabet:n \use_none:n
517 }

```

Now defined \um_symfont_tl as the L^AT_EX math font to access everything:

```

518 \DeclareSymbolFont{\um_symfont_tl}
519 {\encodingdefault}{\zf@family}{\mddefault}{\updefault}

```

And now we input every single maths char. See File II for the source to unicode-math.tex which is used to create unicode-math-table.tex.

```
520 \input{unicode-math-table.tex}
```

Finally,

- Set up shapes for italic/upright or ordinary/var symbols as per package options.
- Remap symbols that don't take their natural mathcode
- Activate any symbols that need to be math-active
- Setup all symbols not covered by the table (mostly alphanumerics)
- Setup the maths alphabets (`\mathbf` etc.)

```
521 \um_setup_all_shapes:
522 \um_remap_symbols:
523 \um_setup_mathactives:
524 \um_setup_delcodes:
525 \um_setup_alphanum:
526 \um_setup_alphabets:
```

End of the `\setmathfont` macro.

```
527 }

528 \cs_new:Nn \um_setup_all_shapes: {
529   \um_setup_nabla:
530   \um_setup_partial:
531   \um_setup_shapes:
532   \um_setup_sfshapes:
533   \um_setup_bfshapes:
534 }
```

7.3.1 Functions for setting up symbols with mathcodes

`\um_process_symbol_noparse:nnnn` If the range font feature has been used, then only a subset of the unicode glyphs
`\um_process_symbol_parse:nnnn` are to be defined. See section §8.3 for the code that enables this.

```
535 \cs_set:Nn \um_process_symbol_noparse:nnnn {
536   \um@mathsymbol{#2}{#3}{\um_symfont_t1}{#1}
537 }

538 \cs_set:Nn \um_process_symbol_parse:nnnn {
539   \um@parse@term{#1}{#2}{#3}{
540     \um_process_symbol_noparse:nnnn{#1}{#2}{#3}{#4}
541   }
542 }
```

`\um_remap_symbols:` This function is used to define the mathcodes for those chars which should be
`\um_remap_symbol_noparse:nnn` mapped to a different glyph than themselves.
`\um_remap_symbol_parse:nnn`

```

543 \cs_new:Nn \um_remap_symbols: {
544   \um_remap_symbol:nnn{\-}{\mathbin}{"02212}% hyphen to minus
545   \um_remap_symbol:nnn{\*}{\mathbin}{"02217}% text asterisk to "cen-
      tred asterisk"
546   \bool_if:NF \g_um_literal_colon_bool {
547     \um_remap_symbol:nnn{\:}{\mathrel}{"02236}% colon to ratio (i.e., punct to rel)
548   }
549   \if@um@literal
550     \um_remap_symbol:nnn {\um@usv@Nabla}{\mathord}{\um@usv@Nabla}
551     \um_remap_symbol:nnn {\um@usv@itNabla}{\mathord}{\um@usv@itNabla}
552     \um_remap_symbol:nnn {\um@usv@partial}{\mathord}{\um@usv@partial}
553     \um_remap_symbol:nnn {\um@usv@itpartial}{\mathord}{\um@usv@itpartial}
554   \else
555     \um_remap_symbol:nnn {\um@usv@Nabla,\um@usv@itNabla}{\mathord}{\um_Nabla_up_or_it_usv}
556     \um_remap_symbol:nnn {\um@usv@partial,\um@usv@itpartial}{\mathord}{\um_partial_up_or_it_usv}
557   \fi

```

Some of these in the `bfliteral` block may be redundant, but that's okay:

```

558   \if@um@bfliteral
559     \um_remap_symbol:nnn {\um@usv@bfNabla}{\mathord}{\um@usv@bfNabla}
560     \um_remap_symbol:nnn {\um@usv@bfitNabla}{\mathord}{\um@usv@bfitNabla}
561     \um_remap_symbol:nnn {\um@usv@bfsfNabla}{\mathord}{\um@usv@bfsfNabla}
562     \um_remap_symbol:nnn {\um@usv@bfsfitNabla}{\mathord}{\um@usv@bfsfitNabla}
563     \um_remap_symbol:nnn {\um@usv@bfpartial}{\mathord}{\um@usv@bfpartial}
564     \um_remap_symbol:nnn {\um@usv@bfitpartial}{\mathord}{\um@usv@bfitpartial}
565     \um_remap_symbol:nnn {\um@usv@bfsfpartial}{\mathord}{\um@usv@bfsfpartial}
566     \um_remap_symbol:nnn {\um@usv@bfsfitpartial}{\mathord}{\um@usv@bfsfitpartial}
567   \else
568     \um_remap_symbol:nnn {\um@usv@bfNabla,\um@usv@bfitNabla}{\mathord}{\um_bfNabla_up_or_it_usv}
569     \um_remap_symbol:nnn {\um@usv@bfsfNabla,\um@usv@bfsfitNabla}{\mathord}{\um_bfsfNabla_up_or_it_usv}
570     \um_remap_symbol:nnn {\um@usv@bfpartial,\um@usv@bfitpartial}{\mathord}{\um_bfpartial_up_or_it_usv}
571     \um_remap_symbol:nnn {\um@usv@bfsfpartial,\um@usv@bfsfitpartial}{\mathord}{\um_bfsfpartial_up_or_it_usv}
572   \fi
573 }

```

Where `\um_remap_symbol:nnn` is defined to be one of these two, depending on the range setup:

```

574 \cs_new:Nn \um_remap_symbol_parse:nnn {
575   \um@parse@term {#3} {\@nil} {#2} {
576     \um_remap_symbol_noparse:nnn {#1} {#2} {#3}
577   }
578 }
579 \cs_new:Nn \um_remap_symbol_noparse:nnn {
580   \clist_map_inline:nn {#1} {
581     \um_set_mathcode:nnnn {##1} {#2} {\um_symfont_tl} {#3}
582   }
583 }

```


7.3.2 Active math characters

There are more math active chars later in the subscript/superscript section. But they don't need to be able to be typeset directly.

`\um_setup_mathactives:`

```
584 \cs_new:Nn \um_setup_mathactives: {  
585   \um_make_mathactive:nnn {"2032} \primesingle \mathord  
586 }
```

`\um_make_mathactive:nnn` : TODO : hook into range feature Makes #1 a mathactive char, and gives cs #2 the meaning of mathchar #1 with class #3. You are responsible for giving active #1 a particular meaning!

```
587 \cs_new:Nn \um_make_mathactive:nnn {  
588   \XeTeXmathchardef #2 = "\mathchar@type #3  
589                       \csname sym\um_symfont_tl\endcsname  
590                       #1 \scan_stop:  
591   \XeTeXmathcodenum #1 = "1FFFFF \scan_stop:  
592 }
```

7.3.3 Delimiter codes

Some symbols that aren't mathopen/mathclose still need to have delimiter codes assigned. The list of vertical arrows may be incomplete. On the other hand, many fonts won't support them all being stretchy.

`\um_setup_delcodes:`

```
593 \cs_new:Nn \um_setup_delcodes: {  
594   \um_set_delcode:nn {\`\/} {\g_um_slash_delimiter_usv}  
595   \um_set_delcode:nn {"2044} {\g_um_slash_delimiter_usv} % fracslash  
596   \um_set_delcode:nn {"2215} {\g_um_slash_delimiter_usv} % divslash  
597   \um_set_delcode:n {"005C} % backslash  
598   \um_set_delcode:nn {\`\<} {"27E8} % angle brackets with ascii notation  
599   \um_set_delcode:nn {\`>} {"27E9} % angle brackets with ascii notation  
600   \um_set_delcode:n {"2191} % up arrow  
601   \um_set_delcode:n {"2193} % down arrow  
602   \um_set_delcode:n {"2195} % updown arrow  
603   \um_set_delcode:n {"219F} % up arrow twohead  
604   \um_set_delcode:n {"21A1} % down arrow twohead  
605   \um_set_delcode:n {"21A5} % up arrow from bar  
606   \um_set_delcode:n {"21A7} % down arrow from bar  
607   \um_set_delcode:n {"21A8} % updown arrow from bar  
608   \um_set_delcode:n {"21BE} % up harpoon right  
609   \um_set_delcode:n {"21BF} % up harpoon left  
610   \um_set_delcode:n {"21C2} % down harpoon right  
611   \um_set_delcode:n {"21C3} % down harpoon left
```

```

612 \um_set_delcode:n {"21C5} % arrows up down
613 \um_set_delcode:n {"21F5} % arrows down up
614 \um_set_delcode:n {"21C8} % arrows up up
615 \um_set_delcode:n {"21CA} % arrows down down
616 \um_set_delcode:n {"21D1} % double up arrow
617 \um_set_delcode:n {"21D3} % double down arrow
618 \um_set_delcode:n {"21D5} % double updown arrow
619 \um_set_delcode:n {"21DE} % up arrow double stroke
620 \um_set_delcode:n {"21DF} % down arrow double stroke
621 \um_set_delcode:n {"21E1} % up arrow dashed
622 \um_set_delcode:n {"21E3} % down arrow dashed
623 }

```

`\um_setup_delcodes:` : TODO : hook into range feature

```

624 \cs_new:Nn \um_set_delcode:nn {
625   \XeTeXdelcode#1 = \csname sym\um_symfont_tl\endcsname #2
626 }
627 \cs_new:Nn \um_set_delcode:n {
628   \XeTeXdelcode#1 = \csname sym\um_symfont_tl\endcsname #1
629 }

```

7.3.4 Maths alphabets' character mapping

`\um_setup_alphanum:`

```

630 \cs_set:Nn \um_setup_alphanum: {
631   \bool_if:NTF \l_um_init_bool {
632     \um_map_chars_numbers:nn {\um@usv@upnum}{\um@usv@upnum}
633     \if@um@literal
634       \um_setup_literals:
635     \else
636       \um_setup_Latin:
637       \um_setup_latin:
638       \um_setup_Greek:
639       \um_setup_greek:
640     \fi
641     \if@um@bfliteral
642       \um_setup_bf_literals:
643     \else
644       \um_setup_bf_Latin:
645       \um_setup_bf_latin:
646       \um_setup_bf_Greek:
647       \um_setup_bf_greek:
648     \fi
649     \bool_if:NTF \g_um_sfliteral_bool {
650       \um_setup_sf_literals:
651     }{

```

```

652     \um_setup_sf_Latin:
653     \um_setup_sf_latin:
654     \um_setup_sf_Greek:
655     \um_setup_sf_greek:
656   }
657   }{
: TODO : what is supposed to happen here?
658   }
659 }

```

7.3.5 Functions for setting up the maths alphabets

`\um_mathmap_noparse:Nnn` #1 : Maths alphabet, *e.g.*, `\mathbb`
 #2 : Input slot(s), *e.g.*, the slot for ‘A’ (comma separated)
 #3 : Output slot, *e.g.*, the slot for ‘A’
 Adds `\um_set_mathcode:nnnn` declarations to the specified maths alphabet’s definition (*e.g.*, `\um@mathscr`). Uses `\um@addto@mathmap` (below) to expand the name of the current symbol font.

```

660 \cs_set:Nn \um_mathmap_noparse:Nnn {
661   \clist_map_inline:nn {#2} {
662     \exp_args:No \um@addto@mathmap \um_symfont_t1 {##1}{#1}{#3}
663   }
664 }

```

`\um_mathmap_parse:Nnn` #1 : Maths alphabet, *e.g.*, `\mathbb`
 #2 : Input slot(s), *e.g.*, the slot for ‘A’ (comma separated)
 #3 : Output slot, *e.g.*, the slot for ‘A’
 When `\um@parse@term` is executed, it populates the `\um@char@num@range` macro with slot numbers corresponding to the specified range. This range is used to conditionally add `\um_set_mathcode:nnnn` declarations to the maths alphabet definition (*e.g.*, `\um@mathscr`).

```

665 \cs_set:Nn \um_mathmap_parse:Nnn {
666   \clist_map_inline:Nn \um@char@num@range {
667     \ifnum##1=#3\relax
668       \clist_map_inline:nn {#2} {
669         \exp_args:No \um@addto@mathmap \um_symfont_t1 {####1}{#1}{#3}
670       }
671     \fi
672   }
673 }

```

`\um@addto@mathmap` #1 : Math symbol font, always/usually the expansion of `\um_symfont_t1`
 #2 : Input slot, *e.g.*, the slot for ‘A’
 #3 : Maths alphabet, *e.g.*, `\mathbb`
 #4 : Output slot, *e.g.*, the slot for ‘A’

This macro is used so that `\um_symfont_t1` can be expanded before entering the `\g@addto@macro` command.

```

674 \newcommand\um@addto@mathmap[4]{
675   \tl_put_right:cn {um_setup_\cs_to_str:N #3:} {
676     \um_set_mathcode:nnnn{#2}{\mathalpha}{#1}{#4}
677   }
678 }

```

7.4 (Big) operators























Turns out that \XeTeX is clever enough to deal with big operators for us automatically with `\XeTeXmathchardef`. Amazing!


However, the limits aren't set automatically; that is, we want to define, a la Plain \TeX *etc.*, `\def\int{\intop\nolimits}`, so there needs to be a transformation from `\int` to `\intop` during the expansion of `\UnicodeMathSymbol` in the appropriate contexts.

Following is a table of every math operator (`\mathop`) defined in `unicode-math-table.tex`, from which a subset need to be flagged for `\nolimits` adjustments. The limits behaviour as specified by `unicode-math` are shown (with grey 'scripts').

usv	Ex.	Macro	Description
U+02140	$\sum\limits_0^1$	<code>\Bbbsum</code>	DOUBLE-STRUCK N-ARY SUMMATION
U+0220F	$\prod\limits_0^1$	<code>\prod</code>	PRODUCT OPERATOR
U+02210	$\coprod\limits_0^1$	<code>\coprod</code>	COPRODUCT OPERATOR
U+02211	$\sum\limits_0^1$	<code>\sum</code>	SUMMATION OPERATOR
U+0222B	$\int\limits_0^1$	<code>\int</code>	INTEGRAL OPERATOR
U+0222C	$\iint\limits_0^1$	<code>\iint</code>	DOUBLE INTEGRAL OPERATOR
U+0222D	$\iiint\limits_0^1$	<code>\iiint</code>	TRIPLE INTEGRAL OPERATOR
U+0222E	$\oint\limits_0^1$	<code>\oint</code>	CONTOUR INTEGRAL OPERATOR
U+0222F	$\oiint\limits_0^1$	<code>\oiint</code>	DOUBLE CONTOUR INTEGRAL OPERATOR
U+02230	$\oiiint\limits_0^1$	<code>\oiiint</code>	TRIPLE CONTOUR INTEGRAL OPERATOR
U+02231	$\int\limits_0^1$	<code>\intclockwise</code>	CLOCKWISE INTEGRAL
U+02232	$\oint\limits_0^1$	<code>\varointclockwise</code>	CONTOUR INTEGRAL, CLOCKWISE
U+02233	$\oint\limits_0^1$	<code>\ointctrclockwise</code>	CONTOUR INTEGRAL, ANTICLOCKWISE
U+022C0	$\bigwedge\limits_0^1$	<code>\bigwedge</code>	LOGICAL OR OPERATOR

U+022C1	\bigvee	<code>\bigvee</code>	LOGICAL AND OPERATOR
U+022C2	\bigcap	<code>\bigcap</code>	INTERSECTION OPERATOR
U+022C3	\bigcup	<code>\bigcup</code>	UNION OPERATOR
U+027D5	\leftthreetimes	<code>\leftthreetimes</code>	LEFT OUTER JOIN
U+027D6	\rightthreetimes	<code>\rightthreetimes</code>	RIGHT OUTER JOIN
U+027D7	\fullthreetimes	<code>\fullthreetimes</code>	FULL OUTER JOIN
U+027D8	\Uparrow	<code>\biguparrow</code>	LARGE UP TACK
U+027D9	\Downarrow	<code>\bigdownarrow</code>	LARGE DOWN TACK
U+029F8	$\big/$	<code>\xsol</code>	BIG SOLIDUS
U+029F9	$\big\backslash$	<code>\xbsol</code>	BIG REVERSE SOLIDUS
U+02A00	\bigodot	<code>\bigodot</code>	N-ARY CIRCLED DOT OPERATOR
U+02A01	\bigoplus	<code>\bigoplus</code>	N-ARY CIRCLED PLUS OPERATOR
U+02A02	\bigotimes	<code>\bigotimes</code>	N-ARY CIRCLED TIMES OPERATOR
U+02A03	$\bigcup\!\!\cdot$	<code>\bigcupdot</code>	N-ARY UNION OPERATOR WITH DOT
U+02A04	$\bigcup\!\!+$	<code>\bigcupplus</code>	N-ARY UNION OPERATOR WITH PLUS
U+02A05	\bigsqcap	<code>\bigsqcap</code>	N-ARY SQUARE INTERSECTION OPERATOR
U+02A06	\bigsqcup	<code>\bigsqcup</code>	N-ARY SQUARE UNION OPERATOR
U+02A07	\And	<code>\conjquant</code>	TWO LOGICAL AND OPERATOR
U+02A08	\Or	<code>\disjquant</code>	TWO LOGICAL OR OPERATOR
U+02A09	\bigtimes	<code>\bigtimes</code>	N-ARY TIMES OPERATOR
U+02A0B	$\sum\limits_0^1$	<code>\sumint</code>	SUMMATION WITH INTEGRAL
U+02A0C	$\iiint\limits_0^1$	<code>\iiint</code>	QUADRUPLE INTEGRAL OPERATOR
U+02A0D	$\int\limits_0^1$	<code>\intbar</code>	FINITE PART INTEGRAL

U+02A0E		<code>\intBar</code>	INTEGRAL WITH DOUBLE STROKE
U+02A0F		<code>\fint</code>	INTEGRAL AVERAGE WITH SLASH
U+02A10		<code>\cirfnint</code>	CIRCULATION FUNCTION
U+02A11		<code>\awint</code>	ANTICLOCKWISE INTEGRATION
U+02A12		<code>\rppolint</code>	PATH AROUND POLE LINE INTEGRATION WITH SEMICIRCULAR
U+02A13		<code>\scpolint</code>	PATH AROUND POLE LINE INTEGRATION NOT INCLUDING THE
U+02A14		<code>\npolint</code>	POLE
U+02A15		<code>\pointint</code>	INTEGRAL AROUND A POINT OPERATOR
U+02A16		<code>\sqint</code>	QUATERNION INTEGRAL OPERATOR INTEGRAL WITH LEFTWARDS ARROW WITH
U+02A17		<code>\intlarhk</code>	HOOK
U+02A18		<code>\intx</code>	INTEGRAL WITH TIMES SIGN
U+02A19		<code>\intcap</code>	INTEGRAL WITH INTERSECTION
U+02A1A		<code>\intcup</code>	INTEGRAL WITH UNION
U+02A1B		<code>\upint</code>	INTEGRAL WITH OVERBAR
U+02A1C		<code>\lowint</code>	INTEGRAL WITH UNDERBAR
U+02A1D		<code>\Join</code>	JOIN
U+02A1E		<code>\bigtriangleleft</code>	LARGE LEFT TRIANGLE OPERATOR
U+02A1F		<code>\zcmp</code>	Z NOTATION SCHEMA COMPOSITION
U+02A20		<code>\zpipe</code>	Z NOTATION SCHEMA PIPING
U+02A21		<code>\zproject</code>	Z NOTATION SCHEMA PROJECTION
U+02AFC		<code>\biginterleave</code>	LARGE TRIPLE VERTICAL BAR OPERATOR
U+02AFF		<code>\bigtalloblong</code>	N-ARY WHITE VERTICAL BAR

`\um@nolimits` This macro is a sequence containing those maths operators that require a `\nolimits` suffix. This list is used when processing `unicode-math-table.tex` to define such commands automatically (see the macro `\um@set@mathsymbol` on page 26). I’ve chosen essentially just the operators that look like integrals; hopefully a better mathematician can help me out here. I’ve a feeling that it’s more useful *not* to include the multiple integrals such as , but that might be a matter of preference.

```

679 \def\um@nolimits{
680   \@elt\int\@elt\iint\@elt\iiint\@elt\iiint\@elt\oint\@elt\oint\@elt\oiint
681   \@elt\intclockwise\@elt\varointclockwise\@elt\ointctrclockwise\@elt\sumint
682   \@elt\intbar\@elt\intBar\@elt\fint\@elt\cirfnint\@elt\awint\@elt\rppolint

```

```

683 \@elt\scpolint\@elt\npolint\@elt\pointint\@elt\sqint\@elt\intlarkh\@elt\intx
684 \@elt\intcap\@elt\intcup\@elt\upint\@elt\lowint
685 }

```

\addnolimits This macro appends material to the macro containing the list of operators that don't take limits. See example following for usage. Note at present that this command must have taken effect before `\setmathfont`.

```

686 \newcommand\addnolimits[1]{
687   \expandafter\def\expandafter\um@nolimits\expandafter{\um@nolimits\@elt#1}
688 }

```

\removenolimits Can this macro be given a better name? It removes (globally) an item from the `nolimits` list. See example following for usage.

```

689 \def\removenolimits#1{
690   \begingroup
691     \def\@elt##1{
692       \ifx##1#1\else
693         \noexpand\@elt\noexpand##1
694       \fi}
695     \xdef\um@nolimits{\um@nolimits}
696   \endgroup
697 }

```

7.5 Radicals

The radical for square root is organised in `\um@set@mathsymbol` on page ?? I think it's the only radical ever. (Actually, there is also `\cuberoot` and `\fourthroot`, but they don't seem to behave as proper radicals.)

Also, what about right-to-left square roots?

\um@radicals We organise radicals in the same way as `nolimits`-operators; that is, in a comma-list.

```

698 \def\um@radicals{\sqrt}

```

$$\sqrt[2]{1 + \sqrt[3]{1+x}}$$

```

\setmathfont{Cambria Math}
\[ \sqrt[2]{1+\sqrt[3]{1+x}} \]

```

7.6 Delimiters

\left We redefine the primitive to be preceded by `\mathopen`; this gives much better spacing in cases such as `\sin\left....` Courtesy of Frank Mittelbach:

<http://www.latex-project.org/cgi-bin/ltxbugs2html?pr=latex/3853&prlatex/3754>

699 `\let\left@primitive\left`

700 `\def\left{\mathopen{}\left@primitive}`

No re-definition is made for `\right` because it's not necessary.

Here are all `\mathopen` characters:

USV	Ex.	Macro	Description
U+00028	(<code>\lparen</code>	LEFT PARENTHESIS
U+0005B	[<code>\lbrack</code>	LEFT SQUARE BRACKET
U+0007B	{	<code>\lbrace</code>	LEFT CURLY BRACKET
U+0007C		<code>\lvert</code>	VERTICAL BAR
U+02016		<code>\lVert</code>	DOUBLE VERTICAL BAR
U+0221A	√	<code>\sqrt</code>	RADICAL
U+0221B	∛	<code>\cuberoot</code>	CUBE ROOT
U+0221C	∜	<code>\fourthroot</code>	FOURTH ROOT
U+02308	⌈	<code>\lceil</code>	LEFT CEILING
U+0230A	⌋	<code>\lfloor</code>	LEFT FLOOR
U+0231C	⌜	<code>\ulcorner</code>	UPPER LEFT CORNER
U+0231E	⌞	<code>\llcorner</code>	LOWER LEFT CORNER
			LIGHT LEFT TORTOISE SHELL BRACKET
U+02772		<code>\lbrbrak</code>	ORNAMENT
U+027C5	⌢	<code>\lbag</code>	LEFT S-SHAPED BAG DELIMITER
U+027CC)	<code>\longdivision</code>	LONG DIVISION
			MATHEMATICAL LEFT WHITE SQUARE
U+027E6	⌌	<code>\lBrack</code>	BRACKET
U+027E8	⌠	<code>\langle</code>	MATHEMATICAL LEFT ANGLE BRACKET
			MATHEMATICAL LEFT DOUBLE ANGLE
U+027EA	⌡	<code>\lAngle</code>	BRACKET
			MATHEMATICAL LEFT WHITE TORTOISE
U+027EC		<code>\lbrbrak</code>	SHELL BRACKET
U+02983	⌸	<code>\lBrace</code>	LEFT WHITE CURLY BRACKET
U+02985	(<code>\lParen</code>	LEFT WHITE PARENTHESIS
U+02987	⌞	<code>\llparenthesis</code>	Z NOTATION LEFT IMAGE BRACKET
U+02989	⌞	<code>\llangle</code>	Z NOTATION LEFT BINDING BRACKET
U+0298B	⌋	<code>\lbrackubar</code>	LEFT SQUARE BRACKET WITH UNDERBAR
			LEFT SQUARE BRACKET WITH TICK IN TOP
U+0298D	⌜	<code>\lbrackultick</code>	CORNER
			LEFT SQUARE BRACKET WITH TICK IN
U+0298F	⌞	<code>\lbracklltick</code>	BOTTOM CORNER
U+02991	⌠	<code>\langedot</code>	LEFT ANGLE BRACKET WITH DOT
U+02993	⌢	<code>\lparenless</code>	LEFT ARC LESS-THAN BRACKET
U+02997	⌞	<code>\lblkbrbrak</code>	LEFT BLACK TORTOISE SHELL BRACKET
U+029D8	⌚	<code>\lvzigzag</code>	LEFT WIGGLY FENCE

U+029DA	⋈	<code>\lvzigzag</code>	LEFT DOUBLE WIGGLY FENCE
U+029FC	⋞	<code>\lcurvyangle</code>	LEFT POINTING CURVED ANGLE BRACKET
U+03014		<code>\lbrbrak</code>	LEFT BROKEN BRACKET
U+03018		<code>\Lbrbrak</code>	LEFT WHITE TORTOISE SHELL BRACKET

And `\mathclose`:

USV	Ex.	Macro	Description
U+00029)	<code>\rparen</code>	RIGHT PARENTHESIS
U+0005D]	<code>\rbrack</code>	RIGHT SQUARE BRACKET
U+0007C		<code>\rvert</code>	VERTICAL BAR
U+0007D	}	<code>\rbrace</code>	RIGHT CURLY BRACKET
U+02016		<code>\rVert</code>	DOUBLE VERTICAL BAR
U+02309	⌋	<code>\rceil</code>	RIGHT CEILING
U+0230B	⌋	<code>\rfloor</code>	RIGHT FLOOR
U+0231D	⌋	<code>\urcorner</code>	UPPER RIGHT CORNER
U+0231F	⌋	<code>\lrcorner</code>	LOWER RIGHT CORNER
			LIGHT RIGHT TORTOISE SHELL BRACKET
U+02773		<code>\rbrbrak</code>	ORNAMENT
U+027C6	⌋	<code>\rbag</code>	RIGHT S-SHAPED BAG DELIMITER
			MATHEMATICAL RIGHT WHITE SQUARE
U+027E7	⌋	<code>\rBrack</code>	BRACKET
U+027E9	⌋	<code>\rangle</code>	MATHEMATICAL RIGHT ANGLE BRACKET
			MATHEMATICAL RIGHT DOUBLE ANGLE
U+027EB	⌋	<code>\rAngle</code>	BRACKET
			MATHEMATICAL RIGHT WHITE TORTOISE
U+027ED		<code>\Rbrbrak</code>	SHELL BRACKET
U+02984	⌋	<code>\rBrace</code>	RIGHT WHITE CURLY BRACKET
U+02986	⌋	<code>\rParen</code>	RIGHT WHITE PARENTHESIS
U+02988	⌋	<code>\rrparenthesis</code>	Z NOTATION RIGHT IMAGE BRACKET
U+0298A	⌋	<code>\rrangle</code>	Z NOTATION RIGHT BINDING BRACKET
U+0298C	⌋	<code>\rbrackubar</code>	RIGHT SQUARE BRACKET WITH UNDERBAR
			RIGHT SQUARE BRACKET WITH TICK IN
U+0298E	⌋	<code>\rbracklrtick</code>	BOTTOM CORNER
			RIGHT SQUARE BRACKET WITH TICK IN TOP
U+02990	⌋	<code>\rbrackurtick</code>	CORNER
U+02992	⌋	<code>\rangledot</code>	RIGHT ANGLE BRACKET WITH DOT
U+02994	⌋	<code>\rpangtr</code>	RIGHT ARC GREATER-THAN BRACKET
U+02998	⌋	<code>\rblbrbrak</code>	RIGHT BLACK TORTOISE SHELL BRACKET
U+029D9	⋈	<code>\rvzigzag</code>	RIGHT WIGGLY FENCE
U+029DB	⋈	<code>\Rvzigzag</code>	RIGHT DOUBLE WIGGLY FENCE
U+029FD	⋞	<code>\rcurvyangle</code>	RIGHT POINTING CURVED ANGLE BRACKET
U+03015		<code>\rbrbrak</code>	RIGHT BROKEN BRACKET
U+03019		<code>\Rbrbrak</code>	RIGHT WHITE TORTOISE SHELL BRACKET

7.7 Maths accents

Maths accents should just work *if they are available in the font*.

USV	Ex.	Macro	Description
U+00300	̀	<code>\grave</code>	GRAVE ACCENT
U+00301	́	<code>\acute</code>	ACUTE ACCENT
U+00302	̂	<code>\hat</code>	CIRCUMFLEX ACCENT
U+00303	̃	<code>\tilde</code>	TILDE
U+00304	̄	<code>\bar</code>	MACRON
U+00305	̅	<code>\overbar</code>	OVERBAR EMBELLISHMENT
U+00306	̇	<code>\breve</code>	BREVE
U+00307	̈	<code>\dot</code>	DOT ABOVE
U+00308	̉	<code>\ddot</code>	DIERESIS
U+00309	̊	<code>\ovhook</code>	COMBINING HOOK ABOVE
U+0030A	̋	<code>\ocirc</code>	RING
U+0030C	̌	<code>\check</code>	CARON
U+00310	̎	<code>\candra</code>	CANDRABINDU (NON-SPACING)
U+00312	̏	<code>\oturnedcomma</code>	COMBINING TURNED COMMA ABOVE GREEK PSILI (SMOOTH BREATHING)
U+00313	̐	<code>\osmooth</code>	(NON-SPACING) GREEK DASIA (ROUGH BREATHING)
U+00314	̑	<code>\orough</code>	(NON-SPACING)
U+00315	̒	<code>\ocommatopright</code>	COMBINING COMMA ABOVE RIGHT
U+0031A	̓	<code>\droang</code>	LEFT ANGLE ABOVE (NON-SPACING) COMBINING LONG SOLIDUS OVERLAY
U+00338	̔	<code>\not</code>	OVERLAY
U+020D0	̰	<code>\leftharpoonaccent</code>	COMBINING LEFT HARPOON ABOVE
U+020D1	̱	<code>\rightharpoonaccent</code>	COMBINING RIGHT HARPOON ABOVE
U+020D2	̲	<code>\vertoverlay</code>	COMBINING LONG VERTICAL LINE OVERLAY
U+020D6	̶	<code>\overleftarrow</code>	COMBINING LEFT ARROW ABOVE
U+020D7	̷	<code>\overrightarrow</code>	COMBINING RIGHT ARROW ABOVE
U+020DB	̸	<code>\dddot</code>	COMBINING THREE DOTS ABOVE
U+020DC	̹	<code>\ddddot</code>	COMBINING FOUR DOTS ABOVE
U+020E1	̺	<code>\overleftrightharpoon</code>	COMBINING LEFT RIGHT ARROW ABOVE
U+020E7	̻	<code>\annuity</code>	COMBINING ANNUITY SYMBOL
U+020E8	̼	<code>\threeunderdot</code>	COMBINING TRIPLE UNDERDOT
U+020E9	̽	<code>\widebridgeabove</code>	COMBINING WIDE BRIDGE ABOVE COMBINING RIGHTWARDS HARPOON WITH
U+020EC	̾	<code>\underrightharpoondown</code>	BARB DOWNWARDS COMBINING LEFTWARDS HARPOON WITH
U+020ED	̿	<code>\underleftharpoondown</code>	BARB DOWNWARDS
U+020EE	̀	<code>\underleftarrow</code>	COMBINING LEFT ARROW BELOW
U+020EF	̽	<code>\underrightarrow</code>	COMBINING RIGHT ARROW BELOW

8 Font features

`\um@zf@feature` Use the same method as `fontspec` for feature definition (*i.e.*, using `xkeyval`) but with a conditional to restrict the scope of these features to unicode-math commands.

```

701 \newcommand\um@zf@feature[2]{
702   \define@key[zf]{options}{#1}[]{}
703   \if@um@fontspec@feature
704     #2
705   \else
706     \PackageError{fontspec/unicode-math}
707       {The ‘#1’ font feature can only be used for maths fonts}
708       {The feature you tried to use can only be in commands
709         like \protect\setmathfont}
710   \fi
711 }
712 }
```

8.1 OpenType maths font features

```

713 \um@zf@feature{ScriptStyle}{
714   \zf@update@ff{+ssty=0}
715 }
716 \um@zf@feature{ScriptScriptStyle}{
717   \zf@update@ff{+ssty=1}
718 }
```

8.2 Script and scriptscript font options

```

719 \define@cmdkey[um]{options}[um@]{script-features}{}
720 \define@cmdkey[um]{options}[um@]{sscript-features}{}
721 \define@cmdkey[um]{options}[um@]{script-font}{}
722 \define@cmdkey[um]{options}[um@]{sscript-font}{}

```

8.3 Range processing

The ‘ALL’ branch here is deprecated and happens automatically.

```

723 \seq_new:N \g_um_mathalph_seq
724 \seq_new:N \l_um_mathalph_seq
725 \seq_new:N \l_um_char_range_seq
726 \define@choicekey+[um]{options}{range}[\@tempa\@tempb]{ALL}{
727   \ifcase\@tempb\relax
728     \bool_set_true:N \l_um_init_bool
729   \fi

```

```

730 }{
731   \bool_set_false:N \l_um_init_bool
732   \seq_clear:N \l_um_char_range_seq
733   \seq_clear:N \l_um_mathalph_seq
734   \clist_map_inline:nn {#1} {
735     \um_if_mathalph_decl:nTF {##1} {
736       \seq_put_right:Nx \l_um_mathalph_seq { {\exp_not:V\l_um_tmpa_tl} {\exp_not:V\l_um_tmpb_tl} }
737     }{
738       \seq_put_right:Nn \l_um_char_range_seq {##1}
739     }
740   }
741 }
742 \cs_generate_variant:Nn \seq_if_in:NnTF {NV}
743 \prg_new_conditional:Nnn \um_if_mathalph_decl:n {TF} {
744   \tl_set:Nn \l_um_tmpa_tl {#1}
745   \tl_set:Nn \l_um_tmpb_tl {}
746   \tl_set:Nn \l_um_tmpc_tl {}
747   \tl_if_in:NnT \l_um_tmpa_tl {->} {
748     \exp_after:wN \um_split_arrow:w \l_um_tmpa_tl \q_nil
749   }
750   \tl_if_in:NnT \l_um_tmpa_tl {/#} {
751     \exp_after:wN \um_split_slash:w \l_um_tmpa_tl \q_nil
752   }
753   \seq_if_in:NVTF \g_um_mathalph_seq \l_um_tmpa_tl {
754     \prg_return_true:
755   }{
756     \prg_return_false:
757   }
758 }
759 \cs_set:Npn \um_split_arrow:w #1->#2 \q_nil {
760   \tl_set:Nn \l_um_tmpa_tl {#1}
761   \tl_set:Nn \l_um_tmpc_tl {#2}
762 }
763 \cs_set:Npn \um_split_slash:w #1/#2 \q_nil {
764   \tl_set:Nn \l_um_tmpa_tl {#1}
765   \tl_set:Nn \l_um_tmpb_tl {#2}
766 }

```

Pretty basic comma separated range processing. Donald Arseneau's selectp package has a cleverer technique.

`\um@parse@term` #1 : unicode character slot
 #2 : control sequence (character macro)
 #3 : control sequence (math type)
 #4 : code to execute

This macro expands to #4 if any of its arguments are contained in `\l_um_char_range_seq`. This list can contain either character ranges (for checking with #1) or

control sequences. These latter can either be the command name of a specific character, *or* the math type of one (*e.g.*, `\mathbin`).

Character ranges are passed to `\um@parse@range`, which accepts input in the form shown in table 13.

Table 13: Ranges accepted by `\um@parse@range`.

Input	Range
x	$r = x$
$x-$	$r \geq x$
$-y$	$r \leq y$
$x-y$	$x \leq r \leq y$

Start by iterating over the commalist, ignoring empties, and initialising the scratch conditional:

```

767 \newcommand\um@parse@term[4]{
768   \seq_map_variable:NNn \l_um_char_range_seq \@ii {
769     \unless\ifx\@ii\@empty
770       \@tempswafalse

```

Match to either the character macro (`\alpha`) or the math type (`\mathbin`):

```

771     \expandafter\um@firstchar\expandafter{\@ii}
772     \ifx\@tempa\um@backslash
773       \expandafter\ifx\@ii#2\relax
774         \@tempswatrue
775       \else
776         \expandafter\ifx\@ii#3\relax
777           \@tempswatrue
778         \fi
779       \fi

```

Otherwise, we have a number range, which is passed to another macro:

```

780     \else
781       \expandafter\um@parse@range\@ii-\@marker-\@nil#1\@nil
782     \fi

```

If we have a match, execute the code! It also populates the `\um@char@num@range` macro, which is used when defining `\mathbf` (*etc.*) `\mathchar` remappings.

```

783     \if@tempswa
784       \ifx\um@char@num@range\@empty
785         \g@addto@macro\um@char@num@range{#1}
786       \else
787         \g@addto@macro\um@char@num@range{, #1}
788       \fi
789     #4%
790   \fi
791 \fi

```

```

792 }
793 }
794 \def\um@firstof#1#2\@nil{#1}
795 \edef\um@backslash{\expandafter\um@firstof\string\string\@nil}
796 \def\um@firstchar#1{\edef\@tempa{\expandafter\um@firstof\string#1\@nil}}

```

\um@parse@range Weird syntax. As shown previously in table 13, this macro can be passed four different input types via \um@parse@term.

```

797 \def\um@parse@range#1-#2-#3\@nil#4\@nil{
798   \def\@tempa{#1}
799   \def\@tempb{#2}

```

Range	$r = x$
C-list input	\@ii=X
Macro input	\um@parse@range X-\@marker-\@nil#1\@nil
Arguments	$\textcolor{red}{\#1}-\textcolor{blue}{\#2}-\textcolor{red}{\#3} = \textcolor{red}{X}-\textcolor{blue}{\@marker}-\textcolor{green}{\{}}$

```

800   \expandafter\ifx\expandafter\@marker\@tempb\relax
801     \ifnum#4=#1\relax
802       \@tempswatrue
803     \fi
804   \else

```

Range	$r \geq x$
C-list input	\@ii=X-
Macro input	\um@parse@range X--\@marker-\@nil#1\@nil
Arguments	$\textcolor{red}{\#1}-\textcolor{blue}{\#2}-\textcolor{red}{\#3} = \textcolor{red}{X}-\textcolor{blue}{\{}}-\textcolor{green}{\@marker}-$

```

805     \ifx\@empty\@tempb
806       \ifnum#4>\numexpr#1-1\relax
807         \@tempswatrue
808       \fi
809     \else

```

Range	$r \leq y$
C-list input	\@ii=-Y
Macro input	\um@parse@range -Y-\@marker-\@nil#1\@nil
Arguments	$\textcolor{red}{\#1}-\textcolor{blue}{\#2}-\textcolor{red}{\#3} = \textcolor{red}{\{}}-\textcolor{green}{Y}-\textcolor{blue}{\@marker}-$

```

810     \ifx\@empty\@tempa
811       \ifnum#4<\numexpr#2+1\relax
812         \@tempswatrue
813       \fi

```

Range	$x \leq r \leq y$
C-list input	\@ii=X-Y
Macro input	\um@parse@range X-Y-\@marker-\@nil#1\@nil
Arguments	$\textcolor{red}{\#1}-\textcolor{blue}{\#2}-\textcolor{red}{\#3} = \textcolor{red}{X}-\textcolor{green}{Y}-\textcolor{blue}{\@marker}-$

```

814   \else
815     \ifnum#4>\numexpr#1-1\relax
816     \ifnum#4<\numexpr#2+1\relax

```

```

817         \@tempswattrue
818         \fi
819     \fi
820 \fi
821 \fi
822 \fi
823 }

```

```

\um_map_char:nn #1 : Number of iterations
                #2 : Starting input char(s)
                #3 : Starting output char
                Loops through character ranges setting \mathcode.
824 \cs_set:Nn \um_map_chars_range:nnn {
825     \clist_map_variable:nNn {#2} \l_um_input_num {
826         \prg_stepwise_variable:nnnNn{0}{1}{#1} \l_um_incr_num {
827             \um_set_mathcode:nnnn
828             {\numexpr \l_um_incr_num+ \l_um_input_num \relax}
829             {\mathalpha}{\um_symfont_t1}
830             {\numexpr \l_um_incr_num + #3 \relax}
831         }
832     }
833 }
834 \cs_set:Nn \um_map_chars_latin:nn {
835     \um_map_chars_range:nnn {25}{#1}{#2}
836 }
837 \cs_set:Nn \um_map_chars_greek:nn {
838     \um_map_chars_range:nnn {24}{#1}{#2}
839 }
840 \cs_set:Nn \um_map_chars_numbers:nn {
841     \um_map_chars_range:nnn {9}{#1}{#2}
842 }
843 \cs_set:Nn \um_map_char:nn {
844     \um_map_chars_range:nnn {0}{#1}{#2}
845 }

```

```

\um_set_mathalphabet_char:Nnn #1 : Maths alphabet
                              #2 : Input char(s)
                              #3 : Output char
                              Loops through character ranges setting \mathcode.
846 \cs_set:Npn \exp_args:Nnff {\::n\::f\::f\:::}
847 \cs_new:Nn \um_set_mathalphabet_char:Nnn {
848     \clist_map_variable:nNn {#2} \l_um_input_num {
849         \exp_args:Nnff \um_mathmap:Nnn {#1}
850         {\number\numexpr\l_um_input_num\relax} {\number\numexpr#3\relax}
851     }
852 }

```

```

\um_set_mathalph_range:Nnn [(Number of iterations)] #1 : Maths alphabet
#2 : Starting input char(s)
#3 : Starting output char
Loops through character ranges setting \mathcode.

853 \cs_new:Nn \um_set_mathalph_range:nNnn {
854   \clist_map_variable:nNn {#3} \l_um_input_num {
855     \errorcontextlines=999
856     \prg_stepwise_variable:nnnNn {0}{1}{#1} \l_um_inc_num {
857       \exp_args:Nnff \um_mathmap:Nnn {#2}
858       {\number\numexpr \l_um_inc_num + \l_um_input_num \relax}
859       {\number\numexpr \l_um_inc_num + #4 \relax}
860     }
861   }
862 }
863 \cs_new:Nn \um_set_mathalphabet_numbers:Nnn {
864   \um_set_mathalph_range:nNnn {9}{#1}{#2}{#3}
865 }
866 \cs_new:Nn \um_set_mathalphabet_latin:Nnn {
867   \um_set_mathalph_range:nNnn {25}{#1}{#2}{#3}
868 }
869 \cs_new:Nn \um_set_mathalphabet_greek:Nnn {
870   \um_set_mathalph_range:nNnn {24}{#1}{#2}{#3}
871 }

```

BCDBCDEABCDEF

`\ExplSyntaxOn`
`{\um_map_chars_range:nnn{3}{`A,`D}{`B}`
`$ABCDEF$} $ABCDEF$`

8.4 Resolving Greek symbol name control sequences

`\um@resolve@greek` This macro defines `\Alpha...``\omega` as their corresponding unicode (mathematical italic) character. Remember that the mapping to upright or italic happens with the `mathcode` definitions, whereas these macros just stand for the literal unicode characters.

```

872 \AtBeginDocument{\um@resolve@greek}
873 \newcommand\um@resolve@greek{
874   \def\Alpha{\mitAlpha}
875   \def\Beta{\mitBeta}
876   \def\Gamma{\mitGamma}
877   \def\Delta{\mitDelta}
878   \def\Epsilon{\mitEpsilon}
879   \def\Zeta{\mitZeta}
880   \def\Eta{\mitEta}
881   \def\Theta{\mitTheta}

```



```

882 \def\Iota{\mitIota}
883 \def\Kappa{\mitKappa}
884 \def\Lambda{\mitLambda}
885 \def\Mu{\mitMu}
886 \def\Nu{\mitNu}
887 \def\Xi{\mitXi}
888 \def\Omicron{\mitOmicron}
889 \def\Pi{\mitPi}
890 \def\Rho{\mitRho}
891 \def\varTheta{\mitvarTheta}
892 \def\Sigma{\mitSigma}
893 \def\Tau{\mitTau}
894 \def\Upsilon{\mitUpsilon}
895 \def\Phi{\mitPhi}
896 \def\Chi{\mitChi}
897 \def\Psi{\mitPsi}
898 \def\Omega{\mitOmega}

```

Lowercase:

```

899 \def\alpha{\mitalpha}
900 \def\beta{\mitbeta}
901 \def\gamma{\mitgamma}
902 \def\delta{\mitdelta}
903 \def\epsilon{
904   \bool_if:NTF \g_um_texgreek_bool {\mitvarepsilon}{\mitepsilon}
905 }
906 \def\zeta{\mitzeta}
907 \def\eta{\miteta}
908 \def\theta{\mittheta}
909 \def\iota{\mitiota}
910 \def\kappa{\mitkappa}
911 \def\lambda{\mitlambda}
912 \def\mu{\mitmu}
913 \def\nu{\mitnu}
914 \def\xi{\mitxi}
915 \def\omicron{\mitomicron}
916 \def\pi{\mitpi}
917 \def\rho{\mitrho}
918 \def\varsigma{\mitvarsigma}
919 \def\sigma{\mitsigma}
920 \def\tau{\mittau}
921 \def\upsilon{\mitupsilon}
922 \def\phi{
923   \bool_if:NTF \g_um_texgreek_bool {\mitvarphi}{\mitphi}
924 }
925 \def\chi{\mitchi}
926 \def\psi{\mitpsi}

```

```

927 \def\omega{\mitomega}
928 \def\varepsilon{
929   \bool_if:NTF \g_um_texgreek_bool {\mitepsilon}{\mitvarepsilon}
930 }
931 \def\vartheta{\mitvartheta}
932 \def\varkappa{\mitvarkappa}
933 \def\varphi{
934   \bool_if:NTF \g_um_texgreek_bool {\mitphi}{\mitvarphi}
935 }
936 \def\varrho{\mitvarrho}
937 \def\varpi{\mitvarpi}
938 }

```

8.5 Setting up the mappings

\um_setup_literals:

```

939 \cs_set:Nn \um_setup_literals: {
940   \um_map_chars_latin:nn {\um@usv@upLatin}{\um@usv@upLatin}
941   \um_map_chars_latin:nn {\um@usv@itLatin}{\um@usv@itLatin}
942   \um_map_chars_latin:nn {\um@usv@itlatin}{\um@usv@itlatin}
943   \um_map_char:nn {\um@usv@ith}{\um@usv@ith}
944   \um_map_chars_latin:nn {\um@usv@uplatin}{\um@usv@uplatin}
945   \um_map_chars_greek:nn {\um@usv@upGreek}{\um@usv@upGreek}
946   \um_map_char:nn {\um@usv@varTheta}{\um@usv@varTheta}
947   \um_map_char:nn {\um@usv@itvarTheta}{\um@usv@itvarTheta}
948   \um_map_chars_greek:nn {\um@usv@itGreek}{\um@usv@itGreek}
949   \um_map_chars_greek:nn {\um@usv@upgreek}{\um@usv@upgreek}
950   \um_map_char:nn {\um@usv@varepsilon}{\um@usv@varepsilon}
951   \um_map_char:nn {\um@usv@vartheta}{\um@usv@vartheta}
952   \um_map_char:nn {\um@usv@varkappa}{\um@usv@varkappa}
953   \um_map_char:nn {\um@usv@varphi}{\um@usv@varphi}
954   \um_map_char:nn {\um@usv@varrho}{\um@usv@varrho}
955   \um_map_char:nn {\um@usv@varpi}{\um@usv@varpi}
956   \um_map_char:nn {\um@usv@itvarepsilon}{\um@usv@itvarepsilon}
957   \um_map_char:nn {\um@usv@itvartheta}{\um@usv@itvartheta}
958   \um_map_char:nn {\um@usv@itvarkappa}{\um@usv@itvarkappa}
959   \um_map_char:nn {\um@usv@itvarphi}{\um@usv@itvarphi}
960   \um_map_char:nn {\um@usv@itvarrho}{\um@usv@itvarrho}
961   \um_map_char:nn {\um@usv@itvarpi}{\um@usv@itvarpi}
962 }

```

\um_setup_bf_literals:

```

963 \cs_set:Nn \um_setup_bf_literals: {
964   \um_map_chars_latin:nn {\um@usv@bfupLatin}{\um@usv@bfupLatin}
965   \um_map_chars_latin:nn {\um@usv@bfuplatin}{\um@usv@bfuplatin}
966   \um_map_chars_latin:nn {\um@usv@bfitLatin}{\um@usv@bfitLatin}

```

```

967 \um_map_chars_latin:nn {\um@usv@bfitlatin}{\um@usv@bfitlatin}
968 \um_map_chars_greek:nn {\um@usv@bfupgreek}{\um@usv@bfupgreek}
969 \um_map_chars_greek:nn {\um@usv@bfupgreek}{\um@usv@bfupgreek}
970 \um_map_chars_greek:nn {\um@usv@bfitgreek}{\um@usv@bfitgreek}
971 \um_map_chars_greek:nn {\um@usv@bfitgreek}{\um@usv@bfitgreek}
972 \um_map_char:nn {\um@usv@bfvarTheta}{\um@usv@bfvarTheta}
973 \um_map_char:nn {\um@usv@bfitvarTheta}{\um@usv@bfitvarTheta}
974 \um_map_char:nn {\um@usv@bfvarepsilon}{\um@usv@bfvarepsilon}
975 \um_map_char:nn {\um@usv@bfvartheta}{\um@usv@bfvartheta}
976 \um_map_char:nn {\um@usv@bfvarkappa}{\um@usv@bfvarkappa}
977 \um_map_char:nn {\um@usv@bfvarphi}{\um@usv@bfvarphi}
978 \um_map_char:nn {\um@usv@bfvarrho}{\um@usv@bfvarrho}
979 \um_map_char:nn {\um@usv@bfvarpi}{\um@usv@bfvarpi}
980 \um_map_char:nn {\um@usv@bfitvarepsilon}{\um@usv@bfitvarepsilon}
981 \um_map_char:nn {\um@usv@bfitvartheta}{\um@usv@bfitvartheta}
982 \um_map_char:nn {\um@usv@bfitvarkappa}{\um@usv@bfitvarkappa}
983 \um_map_char:nn {\um@usv@bfitvarphi}{\um@usv@bfitvarphi}
984 \um_map_char:nn {\um@usv@bfitvarrho}{\um@usv@bfitvarrho}
985 \um_map_char:nn {\um@usv@bfitvarpi}{\um@usv@bfitvarpi}
986 }

```

\um_setup_sf_literals:

```

987 \cs_set:Nn \um_setup_sf_literals: {
988 \um_map_chars_latin:nn {\um@usv@sfupLatin}{\um@usv@sfupLatin}
989 \um_map_chars_latin:nn {\um@usv@sfupLatin}{\um@usv@sfupLatin}
990 \um_map_chars_latin:nn {\um@usv@sfitLatin}{\um@usv@sfitLatin}
991 \um_map_chars_latin:nn {\um@usv@sfitLatin}{\um@usv@sfitLatin}
992 \um_map_chars_latin:nn {\um@usv@bfsfupLatin}{\um@usv@bfsfupLatin}
993 \um_map_chars_latin:nn {\um@usv@bfsfupLatin}{\um@usv@bfsfupLatin}
994 \um_map_chars_latin:nn {\um@usv@bfsfitLatin}{\um@usv@bfsfitLatin}
995 \um_map_chars_latin:nn {\um@usv@bfsfitLatin}{\um@usv@bfsfitLatin}
996 \um_map_chars_greek:nn {\um@usv@bfsfupGreek}{\um@usv@bfsfupGreek}
997 \um_map_chars_greek:nn {\um@usv@bfsfupgreek}{\um@usv@bfsfupgreek}
998 \um_map_chars_greek:nn {\um@usv@bfsfitGreek}{\um@usv@bfsfitGreek}
999 \um_map_chars_greek:nn {\um@usv@bfsfitgreek}{\um@usv@bfsfitgreek}
1000 \um_map_char:nn {\um@usv@bfsfupvarTheta}{\um@usv@bfsfupvarTheta}
1001 \um_map_char:nn {\um@usv@bfsfitvarTheta}{\um@usv@bfsfitvarTheta}
1002 \um_map_char:nn {\um@usv@bfsfupvarepsilon}{\um@usv@bfsfupvarepsilon}
1003 \um_map_char:nn {\um@usv@bfsfupvartheta}{\um@usv@bfsfupvartheta}
1004 \um_map_char:nn {\um@usv@bfsfupvarkappa}{\um@usv@bfsfupvarkappa}
1005 \um_map_char:nn {\um@usv@bfsfupvarphi}{\um@usv@bfsfupvarphi}
1006 \um_map_char:nn {\um@usv@bfsfupvarrho}{\um@usv@bfsfupvarrho}
1007 \um_map_char:nn {\um@usv@bfsfupvarpi}{\um@usv@bfsfupvarpi}
1008 \um_map_char:nn {\um@usv@bfsfitvarepsilon}{\um@usv@bfsfitvarepsilon}
1009 \um_map_char:nn {\um@usv@bfsfitvartheta}{\um@usv@bfsfitvartheta}
1010 \um_map_char:nn {\um@usv@bfsfitvarkappa}{\um@usv@bfsfitvarkappa}
1011 \um_map_char:nn {\um@usv@bfsfitvarphi}{\um@usv@bfsfitvarphi}

```

```

1012 \um_map_char:nn {\um@usv@bfsfitvarrho} {\um@usv@bfsfitvarrho}
1013 \um_map_char:nn {\um@usv@bfsfitvarpi} {\um@usv@bfsfitvarpi}
1014 }

```

\um_setup_Latin:

```

1015 \cs_set:Nn \um_setup_Latin: {
1016 \um_map_chars_latin:nn {\um@usv@upLatin,\um@usv@itLatin}{\um_Latin_up_or_it_usv}
1017 }

```

\um_setup_latin:

```

1018 \cs_set:Nn \um_setup_latin: {
1019 \um_map_chars_latin:nn {\um@usv@uplatin,\um@usv@itlatin}{\um_latin_up_or_it_usv}
1020 \um_map_char:nn {\`h,\um@usv@ith}{\um_latinh_up_or_it_usv}
1021 }

```

\um_setup_Greek:

```

1022 \cs_set:Nn \um_setup_Greek: {
1023 \um_map_chars_greek:nn {\um@usv@upGreek,\um@usv@itGreek}{\um_Greek_up_or_it_usv}
1024 \um_map_char:nn {\um@usv@varTheta,\um@usv@itvarTheta}{\um_varTheta_up_or_it_usv}
1025 }

```

\um_setup_greek:

```

1026 \cs_set:Nn \um_setup_greek: {
1027 \um_map_chars_greek:nn {\um@usv@upgreek,\um@usv@itgreek}{\um_greek_up_or_it_usv}
1028 \if@um@upgreek
1029 \um_map_char:nn {\um@usv@varepsilon,\um@usv@itvarepsilon}{\um@usv@varepsilon}
1030 \um_map_char:nn {\um@usv@vartheta,\um@usv@itvartheta}{\um@usv@vartheta}
1031 \um_map_char:nn {\um@usv@varkappa,\um@usv@itvarkappa}{\um@usv@varkappa}
1032 \um_map_char:nn {\um@usv@varphi,\um@usv@itvarphi}{\um@usv@varphi}
1033 \um_map_char:nn {\um@usv@varrho,\um@usv@itvarrho}{\um@usv@varrho}
1034 \um_map_char:nn {\um@usv@varpi,\um@usv@itvarpi}{\um@usv@varpi}
1035 \else
1036 \um_map_char:nn {\um@usv@varepsilon,\um@usv@itvarepsilon}{\um@usv@itvarepsilon}
1037 \um_map_char:nn {\um@usv@vartheta,\um@usv@itvartheta}{\um@usv@itvartheta}
1038 \um_map_char:nn {\um@usv@varkappa,\um@usv@itvarkappa}{\um@usv@itvarkappa}
1039 \um_map_char:nn {\um@usv@varphi,\um@usv@itvarphi}{\um@usv@itvarphi}
1040 \um_map_char:nn {\um@usv@varrho,\um@usv@itvarrho}{\um@usv@itvarrho}
1041 \um_map_char:nn {\um@usv@varpi,\um@usv@itvarpi}{\um@usv@itvarpi}
1042 \fi
1043 }

```

\um_setup_bf_Latin:

```

1044 \cs_set:Nn \um_setup_bf_Latin: {
1045 \um_map_chars_latin:nn {\um@usv@bfupLatin,\um@usv@bfitLatin}{\um_bf_Latin_up_or_it_usv}
1046 }

```

\um_setup_bf_latin:

```
1047 \cs_set:Nn \um_setup_bf_latin: {  
1048   \um_map_chars_latin:nn {\um@usv@bfuplatin,\um@usv@bfitlatin}{\um_bf_latin_up_or_it_usv}  
1049 }
```

\um_setup_bf_Greek:

```
1050 \cs_set:Nn \um_setup_bf_Greek: {  
1051   \um_map_chars_greek:nn {\um@usv@bfupGreek,\um@usv@bfitGreek}{\um_bf_Greek_up_or_it_usv}  
1052   \um_map_char:nn {\um@usv@bfvarTheta,\um@usv@bfitvarTheta}{\um_bf_varTheta_up_or_it_usv}  
1053 }
```

\um_setup_bf_greek:

```
1054 \cs_set:Nn \um_setup_bf_greek: {  
1055   \um_map_chars_greek:nn {\um@usv@bfupgreek,\um@usv@bfitgreek}{\um_bf_greek_up_or_it_usv}  
1056   \if@um@bfupgreek  
1057     \um_map_char:nn {\um@usv@bfvarepsilon,\um@usv@bfitvarepsilon}{\um@usv@bfvarepsilon}  
1058     \um_map_char:nn {\um@usv@bfvartheta,\um@usv@bfitvartheta}{\um@usv@bfvartheta}  
1059     \um_map_char:nn {\um@usv@bfvarkappa,\um@usv@bfitvarkappa}{\um@usv@bfvarkappa}  
1060     \um_map_char:nn {\um@usv@bfvarphi,\um@usv@bfitvarphi}{\um@usv@bfvarphi}  
1061     \um_map_char:nn {\um@usv@bfvarrho,\um@usv@bfitvarrho}{\um@usv@bfvarrho}  
1062     \um_map_char:nn {\um@usv@bfvarpi,\um@usv@bfitvarpi}{\um@usv@bfvarpi}  
1063     \else  
1064     \um_map_char:nn {\um@usv@bfvarepsilon,\um@usv@bfitvarepsilon}{\um@usv@bfitvarepsilon}  
1065     \um_map_char:nn {\um@usv@bfvartheta,\um@usv@bfitvartheta}{\um@usv@bfitvartheta}  
1066     \um_map_char:nn {\um@usv@bfvarkappa,\um@usv@bfitvarkappa}{\um@usv@bfitvarkappa}  
1067     \um_map_char:nn {\um@usv@bfvarphi,\um@usv@bfitvarphi}{\um@usv@bfitvarphi}  
1068     \um_map_char:nn {\um@usv@bfvarrho,\um@usv@bfitvarrho}{\um@usv@bfitvarrho}  
1069     \um_map_char:nn {\um@usv@bfvarpi,\um@usv@bfitvarpi}{\um@usv@bfitvarpi}  
1070     \fi  
1071 }
```

\um_setup_sf_Latin:

```
1072 \cs_set:Nn \um_setup_sf_Latin: {  
1073   \um_map_chars_latin:nn {\um@usv@sfupLatin,\um@usv@sfitLatin}{\um_sf_Latin_up_or_it_usv}  
1074   \um_map_chars_latin:nn {\um@usv@bfsfupLatin,\um@usv@bfsfitLatin}{\um_bfsf_Latin_up_or_it_usv}  
1075 }
```

\um_setup_sf_latin:

```
1076 \cs_set:Nn \um_setup_sf_latin: {  
1077   \um_map_chars_latin:nn {\um@usv@sfuplatin,\um@usv@sfitlatin}{\um_sf_latin_up_or_it_usv}  
1078   \um_map_chars_latin:nn {\um@usv@bfsfuplatin,\um@usv@bfsfitlatin}{\um_bfsf_latin_up_or_it_usv}  
1079 }
```

\um_setup_sf_Greek:

```
1080 \cs_set:Nn \um_setup_sf_Greek: {  
1081   \um_map_chars_greek:nn {\um@usv@bfsfupGreek,\um@usv@bfsfitGreek}{\um_bfsf_Greek_up_or_it_usv}
```

```

1082 \um_map_char:nn {\um@usv@bfsfupvarTheta,\um@usv@bfsfitvarTheta}{\um_bfsf_varTheta_up_or_
1083 }

\um_setup_sf_greek:
1084 \cs_set:Nn \um_setup_sf_greek: {
1085   \um_map_chars_greek:nn {\um@usv@bfsfupgreek,\um@usv@bfsfitgreek}{\um_bfsf_greek_up_or_it
1086   \if@um@bfupgreek
1087   \um_map_char:nn {\um@usv@bfsfupvarepsilon,\um@usv@bfsfitvarepsilon}{\um@usv@bfsfupvarep
1088   \um_map_char:nn {\um@usv@bfsfupvartheta,\um@usv@bfsfitvartheta} {\um@usv@bfsfupvarthe
1089   \um_map_char:nn {\um@usv@bfsfupvarkappa,\um@usv@bfsfitvarkappa} {\um@usv@bfsfupvarkap
1090   \um_map_char:nn {\um@usv@bfsfupvarphi,\um@usv@bfsfitvarphi} {\um@usv@bfsfupvarphi}
1091   \um_map_char:nn {\um@usv@bfsfupvarrho,\um@usv@bfsfitvarrho} {\um@usv@bfsfupvarrho}
1092   \um_map_char:nn {\um@usv@bfsfupvarpi,\um@usv@bfsfitvarpi} {\um@usv@bfsfupvarpi}
1093   \else
1094   \um_map_char:nn {\um@usv@bfsfupvarepsilon,\um@usv@bfsfitvarepsilon}{\um@usv@bfsfitvarep
1095   \um_map_char:nn {\um@usv@bfsfupvartheta,\um@usv@bfsfitvartheta} {\um@usv@bfsfitvarthe
1096   \um_map_char:nn {\um@usv@bfsfupvarkappa,\um@usv@bfsfitvarkappa} {\um@usv@bfsfitvarkap
1097   \um_map_char:nn {\um@usv@bfsfupvarphi,\um@usv@bfsfitvarphi} {\um@usv@bfsfitvarphi}
1098   \um_map_char:nn {\um@usv@bfsfupvarrho,\um@usv@bfsfitvarrho} {\um@usv@bfsfitvarrho}
1099   \um_map_char:nn {\um@usv@bfsfupvarpi,\um@usv@bfsfitvarpi} {\um@usv@bfsfitvarpi}
1100   \fi
1101 }

```

9 Maths alphabets mapping definitions

Algorithm for setting alphabet fonts. By default, when `range` is empty, we are in *implicit* mode. If `range` contains the name of the math alphabet, we are in *explicit* mode and do things slightly differently.

Implicit mode:

- Try and set all of the alphabet shapes.
- Check for the first glyph of the uppercase Latin alphabet to detect if the font supports each alphabet shape. (This doesn't work to distinguish Latin/Greek but we hope all maths fonts will have at least them!)
- For alphabets that do exist, overwrite whatever's already there.
- For alphabets that are not supported, *do nothing*. (This includes leaving the old alphabet definition in place.)

Explicit mode:

- Only set the alphabets specified.
- Check for the first glyph of the alphabet to detect if the font contains the alphabet shape in the unicode math plane.

- For unicode math alphabets, overwrite whatever's already there.
- Otherwise, use the ASCII letters instead.

This is every math alphabet known to unicode-math:

`\g_um_mathalph_seq`

```

1102 \seq_clear:N \g_um_mathalph_seq
1103 \tl_map_inline:nn {
1104   \mathup\mathit
1105   \mathbb\mathscr\mathfrak\mathtt
1106   \mathsf\mathsfup\mathsfit
1107   \mathbf\mathbfup\mathbfit
1108   \mathbfscr\mathbffrak
1109   \mathbfssf\mathbfsfup\mathbfsfitt
1110 }{
1111   \seq_put_right:Nn \g_um_mathalph_seq {#1}
1112 }

```

`\um_setup_alphabets:`

```

1113
1114 \tl_new:Nn \g_um_mathup_alph_clist {latin, Latin, greek, Greek, num}
1115 \tl_new:Nn \g_um_mathit_alph_clist {latin, Latin, greek, Greek}
1116 \tl_new:Nn \g_um_mathscr_alph_clist {latin, Latin}
1117 \tl_new:Nn \g_um_mathfrak_alph_clist {latin, Latin}
1118 \tl_new:Nn \g_um_mathbfscr_alph_clist {latin, Latin}
1119 \tl_new:Nn \g_um_mathbffrak_alph_clist {latin, Latin}
1120 \tl_new:Nn \g_um_mathbb_alph_clist {latin, Latin, num}
1121 \tl_new:Nn \g_um_mathtt_alph_clist {latin, Latin, num}
1122 \tl_new:Nn \g_um_mathsf_alph_clist {latin, Latin, num}
1123 \tl_new:Nn \g_um_mathsfup_alph_clist {latin, Latin, num}
1124 \tl_new:Nn \g_um_mathsfitt_alph_clist {latin, Latin, num}
1125 \tl_new:Nn \g_um_mathbf_alph_clist {latin, Latin, greek, Greek, num}
1126 \tl_new:Nn \g_um_mathbfup_alph_clist {latin, Latin, greek, Greek, num}
1127 \tl_new:Nn \g_um_mathbfitt_alph_clist {latin, Latin, greek, Greek, num}
1128 \tl_new:Nn \g_um_mathbfssf_alph_clist {latin, Latin, greek, Greek, num}
1129 \tl_new:Nn \g_um_mathbfssfup_alph_clist {latin, Latin, greek, Greek, num}
1130 \tl_new:Nn \g_um_mathbfsfitt_alph_clist {latin, Latin, greek, Greek, num}
1131
1132 \tl_new:Nn \g_um_mathup_latin_usv {\a-\z}
1133 \tl_new:Nn \g_um_mathup_Latin_usv {\A-\Z}
1134 \tl_new:Nn \g_um_mathup_greek_usv {"3B1-"3C9,"3F5,"3D1,"3F0,"3D5,"3F1,"3D6,"3DD}
1135 \tl_new:Nn \g_um_mathup_Greek_usv {"391-"3A9,"3F4,"3DC}
1136 \tl_new:Nn \g_um_mathup_num_usv {\0-\9}
1137
1138 \tl_new:Nn \g_um_mathit_latin_usv {"1D44E-"1D467,\um@usv@ith}
1139 \tl_new:Nn \g_um_mathit_Latin_usv {"1D434-"1D44C}

```

```

1140 \tl_new:Nn \g_um_mathit_greek_usv {"1D6FC-"1D714,"1D716-1D71B}
1141 \tl_new:Nn \g_um_mathit_greek_usv {"1D6E2-"1D6FA}
1142
1143 \tl_new:Nn \g_um_mathscr_usv {latin, Latin}
1144 \tl_new:Nn \g_um_mathfrak_usv {latin, Latin}
1145 \tl_new:Nn \g_um_mathbfscr_usv {latin, Latin}
1146 \tl_new:Nn \g_um_mathbffrak_usv {latin, Latin}
1147 \tl_new:Nn \g_um_mathbb_usv {latin, Latin, num}
1148 \tl_new:Nn \g_um_mathhtt_usv {latin, Latin, num}
1149 \tl_new:Nn \g_um_mathsf_usv {latin, Latin, num}
1150 \tl_new:Nn \g_um_mathsfup_usv {latin, Latin, num}
1151 \tl_new:Nn \g_um_mathsfitt_usv {latin, Latin, num}
1152 \tl_new:Nn \g_um_mathbf_usv {latin, Latin, greek, Greek, num}
1153 \tl_new:Nn \g_um_mathbfup_usv {latin, Latin, greek, Greek, num}
1154 \tl_new:Nn \g_um_mathbfit_usv {latin, Latin, greek, Greek, num}
1155 \tl_new:Nn \g_um_mathbfsf_usv {latin, Latin, greek, Greek, num}
1156 \tl_new:Nn \g_um_mathbfsfup_usv {latin, Latin, greek, Greek, num}
1157 \tl_new:Nn \g_um_mathbfsfitt_usv {latin, Latin, greek, Greek, num}
1158
1159 \cs_new:Nn \um_setup_alphabets: {
1160   \seq_if_empty:NTF \l_um_mathalph_seq {
1161     \um_setup_math_alphabet:NV \mathup \g_um_mathup_alph_clist
1162     \um_setup_math_alphabet:NV \mathit \g_um_mathit_alph_clist
1163     \um_setup_math_alphabet:NV \mathbb \g_um_mathbb_alph_clist
1164     \um_setup_math_mapping:n {bb }
1165     \um_maybe_init_alphabet:n {bbit }
1166     \um_setup_math_mapping:n {bbit }
1167     \um_setup_math_alphabet:NV \mathscr \g_um_mathscr_alph_clist
1168     \um_setup_math_alphabet:NV \mathfrak \g_um_mathfrak_alph_clist
1169     \um_setup_math_alphabet:NV \mathsf \g_um_mathsf_alph_clist
1170     \um_setup_math_alphabet:NV \mathsfup \g_um_mathsfup_alph_clist
1171     \um_setup_math_alphabet:NV \mathsfitt \g_um_mathsfitt_alph_clist
1172     \um_setup_math_alphabet:NV \mathhtt \g_um_mathhtt_alph_clist
1173     \um_setup_math_alphabet:NV \mathbf \g_um_mathbf_alph_clist
1174     \um_setup_math_alphabet:NV \mathbfup \g_um_mathbfup_alph_clist
1175     \um_setup_math_alphabet:NV \mathbfit \g_um_mathbfit_alph_clist
1176     \um_setup_math_alphabet:NV \mathbfscr \g_um_mathbfscr_alph_clist
1177     \um_setup_math_alphabet:NV \mathbffrak \g_um_mathbffrak_alph_clist
1178     \um_setup_math_alphabet:NV \mathbfsf \g_um_mathbfsf_alph_clist
1179     \um_setup_math_alphabet:NV \mathbfsfup \g_um_mathbfsfup_alph_clist
1180     \um_setup_math_alphabet:NV \mathbfsfitt \g_um_mathbfsfitt_alph_clist
1181   }{
1182     \cs_set_eq:NN \um_mathmap:Nnn \um_mathmap_noparse:Nnn
1183     \seq_map_inline:Nn \l_um_mathalph_seq {
1184       \tl_set:No \l_um_tmpa_tl { \use_i:nnn ##1 }
1185       \tl_set:No \l_um_tmpb_tl { \use_ii:nnn ##1 }

```



```

1186 \tl_set:No \l_um_tmpc_tl { \use_iii:nnn ##1 }
1187 \tl_if_empty:NF \l_um_tmpc_tl {
1188 \PackageWarning{unicode-math}{alphabet~remapping~not~yet~implemented}
1189 }
1190 \tl_if_empty:NT \l_um_tmpb_tl {
1191 \cs_set_eq:NN \um_maybe_init_alphabet:n \um_init_alphabet:n
1192 \tl_set:Nv \l_um_tmpb_tl { g_um_ \exp_after:wN \cs_to_str:N \l_um_tmpa_tl _alph_clist }
1193 }
1194 \um_setup_math_alphabet:VV \l_um_tmpa_tl \l_um_tmpb_tl
1195 }
1196 }
1197 }

```

`\um_setup_math_alphabet:Nn` #1 : Math font family name (e.g., ‘sf’)

#2 : Math alphabets, comma separated of {latin, Latin, greek, Greek, num}

First check that at least one of the alphabets for the font shape is defined, and then then loop through them defining the individual ranges.

```

1198 \cs_new:Nn \um_setup_math_alphabet:Nn {
1199 \tl_set:Nx \l_um_tmpa_tl { \cs_to_str:N #1 }
1200 \tl_set:Nx \l_um_tmpb_tl { \exp_after:wN \use_none:nnnn \l_um_tmpa_tl }
1201 \clist_map_inline:nn {#2} {
1202 \um_glyph_if_exist:cT {um@usv@ \l_um_tmpb_tl ##1}{
1203 \exp_args:NV \um_maybe_init_alphabet:n \l_um_tmpb_tl
1204 \clist_map_break:
1205 }
1206 }
1207 \clist_map_inline:nn {#2} {
1208 \um_glyph_if_exist:cTF {um@usv@ \l_um_tmpb_tl ##1}{
1209 \use:c {um_config_ \l_um_tmpa_tl _##1:}
1210 }{
1211 \PackageWarningNoLine{unicode-math}{^^J\space\space\space\space
1212 Math~ alphabet~
1213 \@backslashchar \l_um_tmpa_tl~
1214 (\tl_use:c{g_um_math_alphabet_name_##1_tl})~
1215 not~ found~ in~ font~
1216 \fontname\um@font}
1217 }
1218 }
1219 }
1220 \cs_generate_variant:Nn \um_setup_math_alphabet:Nn {NV,VV}
1221
1222 \tl_set:Nn \g_um_math_alphabet_name_latin_tl {Latin, lowercase}
1223 \tl_set:Nn \g_um_math_alphabet_name_Latin_tl {Latin, uppercase}
1224 \tl_set:Nn \g_um_math_alphabet_name_greek_tl {Greek, lowercase}
1225 \tl_set:Nn \g_um_math_alphabet_name_Greek_tl {Greek, uppercase}
1226 \tl_set:Nn \g_um_math_alphabet_name_num_tl {Numerals}

```

```

1227 \cs_new:Nn \um_setup_math_mapping:n {
1228   \cs_if_exist:cT {um_setup_math#1:} {
1229     \use:c {um_config_math#1_misc:}
1230   }
1231 }

1232 \cs_set:Nn \um_init_alphabet:n {
1233   \wlog{unicode-math:~Initialiasing~\@backslashchar math#1}
1234   \um_prepare_alph:n {#1}
1235   \cs_set_eq:cN {um_setup_math#1:} \prg_do_nothing:
1236 }

```

`\um_glyph_if_exist:nTF` : TODO: Generalise for arbitrary fonts! `\um@font` is not always the one used for a specific glyph!!

```

1237 \prg_new_conditional:Nnn \um_glyph_if_exist:n {p,TF,T,F} {
1238   \etex_iffontchar:D \um@font #1 \scan_stop: \prg_return_true: \else: \prg_return_false: \fi:
1239 }
1240 \cs_generate_variant:Nn \um_glyph_if_exist_p:n {c}
1241 \cs_generate_variant:Nn \um_glyph_if_exist:nTF {c}
1242 \cs_generate_variant:Nn \um_glyph_if_exist:nT {c}
1243 \cs_generate_variant:Nn \um_glyph_if_exist:nF {c}

```

`\um_prepare_alph:n` If `\mathXY` hasn't been (re-)declared yet, then define it in terms of `unicode-math` definitions. Use `\bgroup/\egroup` so s'scripts scan the whole thing.

```

1244 \cs_new:Nn \um_prepare_alph:n {
1245   \cs_if_exist:cF {um_math#1:n} {
1246     \cs_set:cpn {um_math#1:n} ##1 {
1247       \use:c {um_setup_math#1:} ##1 \egroup
1248     }
1249     \cs_set_protected:cpn {math#1} {
1250       \bgroup
1251       \mode_if_math:F {
1252         \egroup\expandafter
1253         \non@alpher\expandafter{\csname math#1\endcsname\space}
1254       }
1255       \use:c {um_math#1:n}
1256     }
1257   }
1258 }

```

9.1 Non-bold math alphabets

9.1.1 Upright: `\mathup`

```

1259 \cs_new:Npn \um_config_mathup_Latin: {
1260   \um_set_mathalphabet_latin:Nnn{\mathup}{\um@usv@upLatin,\um@usv@itLatin}{\um@usv@upLatin}
1261 }

```

```

1262 \cs_new:Npn \um_config_mathup_latin: {
1263   \um_set_mathalphabet_latin:Nnn{\mathup}{\um@usv@uplatin,\um@usv@itlatin}{\um@usv@uplatin}
1264   \um_set_mathalphabet_char:Nnn{\mathup}{\um@usv@ith} {\`h}
1265 }
1266 \cs_new:Npn \um_config_mathup_num: {
1267   \um_set_mathalphabet_numbers:Nnn{\mathup}{\um@usv@upnum}{\um@usv@upnum}
1268 }
1269 \cs_new:Npn \um_config_mathup_Greek: {
1270   \um_set_mathalphabet_greek:Nnn{\mathup}{\um@usv@upGreek,\um@usv@itGreek}{\um@usv@upGreek}
1271   \um_set_mathalphabet_char:Nnn{\mathup}{\um@usv@Nabla,\um@usv@itNabla}{\um@usv@Nabla}
1272   \um_set_mathalphabet_char:Nnn{\mathup}{\um@usv@varTheta,\um@usv@itvarTheta}{\um@usv@varTheta}
1273 }
1274 \cs_new:Npn \um_config_mathup_greek: {
1275   \um_set_mathalphabet_greek:Nnn{\mathup}{\um@usv@upgreek,\um@usv@itgreek}{\um@usv@upgreek}
1276   \um_set_mathalphabet_char:Nnn{\mathup}{\um@usv@partial,\um@usv@itpartial}{\um@usv@partial}
1277   \um_set_mathalphabet_char:Nnn{\mathup}{\um@usv@varepsilon,\um@usv@itvarepsilon}{\um@usv@varepsilon}
1278   \um_set_mathalphabet_char:Nnn{\mathup}{\um@usv@vartheta,\um@usv@itvartheta}{\um@usv@vartheta}
1279   \um_set_mathalphabet_char:Nnn{\mathup}{\um@usv@varkappa,\um@usv@itvarkappa}{\um@usv@varkappa}
1280   \um_set_mathalphabet_char:Nnn{\mathup}{\um@usv@varphi,\um@usv@itvarphi}{\um@usv@varphi}
1281   \um_set_mathalphabet_char:Nnn{\mathup}{\um@usv@varrho,\um@usv@itvarrho}{\um@usv@varrho}
1282   \um_set_mathalphabet_char:Nnn{\mathup}{\um@usv@varpi,\um@usv@itvarpi}{\um@usv@varpi}
1283 }

```

9.1.2 Italic: \mathit

```

1284 \cs_new:Npn \um_config_mathit_Latin: {
1285   \um_set_mathalphabet_latin:Nnn{\mathit}{\um@usv@upLatin,\um@usv@itLatin}{\um@usv@itLatin}
1286 }
1287 \cs_new:Npn \um_config_mathit_latin: {
1288   \um_set_mathalphabet_latin:Nnn{\mathit}{\um@usv@uplatin,\um@usv@itlatin}{\um@usv@itlatin}
1289   \um_set_mathalphabet_char:Nnn{\mathit}{\`h,\um@usv@ith}{\um@usv@ith}
1290 }
1291 \cs_new:Npn \um_config_mathit_Greek: {
1292   \um_set_mathalphabet_greek:Nnn{\mathit}{\um@usv@upGreek,\um@usv@itGreek}{\um@usv@itGreek}
1293   \um_set_mathalphabet_char:Nnn{\mathit}{\um@usv@varTheta,\um@usv@itvarTheta}{\um@usv@itvarTheta}
1294   \um_set_mathalphabet_char:Nnn{\mathit}{\um@usv@Nabla,\um@usv@itNabla}{\um@usv@itNabla}
1295 }
1296 \cs_new:Npn \um_config_mathit_greek: {
1297   \um_set_mathalphabet_greek:Nnn{\mathit}{\um@usv@upgreek,\um@usv@itgreek}{\um@usv@itgreek}
1298   \um_set_mathalphabet_char:Nnn{\mathit}{\um@usv@partial,\um@usv@itpartial}{\um@usv@itpartial}
1299   \um_set_mathalphabet_char:Nnn{\mathit}{\um@usv@varepsilon,\um@usv@itvarepsilon}{\um@usv@itvarepsilon}
1300   \um_set_mathalphabet_char:Nnn{\mathit}{\um@usv@vartheta,\um@usv@itvartheta}{\um@usv@itvartheta}
1301   \um_set_mathalphabet_char:Nnn{\mathit}{\um@usv@varkappa,\um@usv@itvarkappa}{\um@usv@itvarkappa}
1302   \um_set_mathalphabet_char:Nnn{\mathit}{\um@usv@varphi,\um@usv@itvarphi}{\um@usv@itvarphi}
1303   \um_set_mathalphabet_char:Nnn{\mathit}{\um@usv@varrho,\um@usv@itvarrho}{\um@usv@itvarrho}
1304   \um_set_mathalphabet_char:Nnn{\mathit}{\um@usv@varpi,\um@usv@itvarpi}{\um@usv@itvarpi}
1305 }

```

9.1.3 Blackboard or double-struck: `\mathbb` and `\mathbbi`

: TODO : make bbit work with literal input?

```
1306 \cs_new:Npn \um_config_mathbb_latin: {
1307   \um_set_mathalphabet_latin:Nnn{\mathbb}{\um@usv@uplatin,\um@usv@itlatin}{\um@usv@bblatin}
1308   \um_set_mathalphabet_char:Nnn{\mathbb}{\um@usv@ith} {"1D559}
1309 }
1310 \cs_new:Npn \um_config_mathbb_Latin: {
1311   \um_set_mathalphabet_latin:Nnn{\mathbb}{\um@usv@upLatin,\um@usv@itLatin}{\um@usv@bbLatin}
1312   \um_set_mathalphabet_char:Nnn{\mathbb}{`\C,"1D436}{ "2102}
1313   \um_set_mathalphabet_char:Nnn{\mathbb}{`\H,"1D43B}{ "210D}
1314   \um_set_mathalphabet_char:Nnn{\mathbb}{`\N,"1D441}{ "2115}
1315   \um_set_mathalphabet_char:Nnn{\mathbb}{`\P,"1D443}{ "2119}
1316   \um_set_mathalphabet_char:Nnn{\mathbb}{`\Q,"1D444}{ "211A}
1317   \um_set_mathalphabet_char:Nnn{\mathbb}{`\R,"1D445}{ "211D}
1318   \um_set_mathalphabet_char:Nnn{\mathbb}{`\Z,"1D44D} {"2124}
1319 }
1320 \cs_new:Npn \um_config_mathbb_num: {
1321   \um_set_mathalphabet_numbers:Nnn{\mathbb}{\um@usv@upnum}{\um@usv@bbnum}
1322 }
1323 \cs_new:Npn \um_config_mathbb_misc: {
1324   \um_set_mathalphabet_char:Nnn \mathbb {"03A0,"1D6F1}{ "213F} % Pi
1325   \um_set_mathalphabet_char:Nnn \mathbb {"03C0,"1D70B}{ "213C} % pi
1326   \um_set_mathalphabet_char:Nnn \mathbb {"0393,"1D6E4}{ "213E} % Gamma
1327   \um_set_mathalphabet_char:Nnn \mathbb {"03B3,"1D6FE}{ "213D} % gamma
1328   \um_set_mathalphabet_char:Nnn \mathbb {"2211}{ "2140} % summation
1329 }
1330 \cs_new:Npn \um_config_mathbbit_misc: {
1331   \um_set_mathalphabet_char:Nnn \mathbbit {\D,"1D437}{ "2145}
1332   \um_set_mathalphabet_char:Nnn \mathbbit {\d,"1D451}{ "2146}
1333   \um_set_mathalphabet_char:Nnn \mathbbit {\e,"1D452}{ "2147}
1334   \um_set_mathalphabet_char:Nnn \mathbbit {\i,"1D456}{ "2148}
1335   \um_set_mathalphabet_char:Nnn \mathbbit {\j,"1D457}{ "2149}
1336 }
```

9.1.4 Script or caligraphic: `\mathscr` and `\mathcal`

```
1337 \cs_new:Npn \um_config_mathscr_Latin: {
1338   \um_set_mathalphabet_latin:Nnn \mathscr {\um@usv@upLatin,\um@usv@itLatin}{\um@usv@scrLatin}
1339   \um_set_mathalphabet_char:Nnn \mathscr {\B,"1D435}{ "212C}
1340   \um_set_mathalphabet_char:Nnn \mathscr {\E,"1D438}{ "2130}
1341   \um_set_mathalphabet_char:Nnn \mathscr {\F,"1D439}{ "2131}
1342   \um_set_mathalphabet_char:Nnn \mathscr {\H,"1D43B}{ "210B}
1343   \um_set_mathalphabet_char:Nnn \mathscr {\I,"1D43C}{ "2110}
1344   \um_set_mathalphabet_char:Nnn \mathscr {\L,"1D43F}{ "2112}
1345   \um_set_mathalphabet_char:Nnn \mathscr {\M,"1D440}{ "2133}
1346   \um_set_mathalphabet_char:Nnn \mathscr {\R,"1D445}{ "211B}
```

```

1347 }
1348 \cs_new:Npn \um_config_mathscr_latin: {
1349   \um_set_mathalphabet_latin:Nnn \mathscr {\um@usv@uplatin,\um@usv@itlatin}{\um@usv@scrlatin}
1350   \um_set_mathalphabet_char:Nnn \mathscr {\`e,"1D452}{ "212F}
1351   \um_set_mathalphabet_char:Nnn \mathscr {\`g,"1D454}{ "210A}
1352   \um_set_mathalphabet_char:Nnn \mathscr {\`o,"1D45C}{ "2134}
1353   \um_set_mathalphabet_char:Nnn \mathscr {\um@usv@ith} {"1D4BD}
1354 }

```

9.1.5 Fraktur or fraktur or blackletter: `\mathfrak`

```

1355 \cs_new:Npn \um_config_mathfrak_Latin: {
1356   \um_set_mathalphabet_latin:Nnn \mathfrak {\um@usv@upLatin,\um@usv@itLatin}{\um@usv@frakLat
1357   \um_set_mathalphabet_char:Nnn \mathfrak {\`C,"1D436}{ "212D}
1358   \um_set_mathalphabet_char:Nnn \mathfrak {\`H,"1D43B}{ "210C}
1359   \um_set_mathalphabet_char:Nnn \mathfrak {\`I,"1D43C}{ "2111}
1360   \um_set_mathalphabet_char:Nnn \mathfrak {\`R,"1D445}{ "211C}
1361   \um_set_mathalphabet_char:Nnn \mathfrak {\`Z,"1D44D}{ "2128}
1362 }
1363 \cs_new:Npn \um_config_mathfrak_latin: {
1364   \um_set_mathalphabet_latin:Nnn \mathfrak {\um@usv@uplatin,\um@usv@itlatin}{\um@usv@fraklati
1365   \um_set_mathalphabet_char:Nnn \mathfrak {\um@usv@ith} {"1D525}
1366 }

```

9.1.6 Sans serif: `\mathsf`

```

1367 \cs_new:Npn \um_config_mathsf_Latin: {
1368   \bool_if:NTF \g_um_sfliteral_bool {
1369     \um_set_mathalphabet_latin:Nnn{\mathsf}{\um@usv@upLatin}{\um@usv@sfupLatin}
1370     \um_set_mathalphabet_latin:Nnn{\mathsf}{\um@usv@itLatin}{\um@usv@sfitLatin}
1371   }{
1372     \um_set_mathalphabet_latin:Nnn \mathsf {\um@usv@upLatin,\um@usv@itLatin}{ \um_sf_Latin_up
1373   }
1374 }
1375 \cs_new:Npn \um_config_mathsf_latin: {
1376   \bool_if:NTF \g_um_upsans_bool {
1377     \um_set_mathalphabet_latin:Nnn{\mathsf}{\um@usv@uplatin}{\um@usv@sfuplatin}
1378     \um_set_mathalphabet_latin:Nnn{\mathsf}{\um@usv@itlatin}{\um@usv@sfitlatin}
1379     \um_set_mathalphabet_char:Nnn \mathsf {\um@usv@ith} {"1D629}
1380   }{
1381     \um_set_mathalphabet_latin:Nnn \mathsf {\um@usv@uplatin,\um@usv@itlatin}{ \um_sf_latin_up
1382     \bool_if:NTF \g_um_upsans_bool {
1383       \um_set_mathalphabet_char:Nnn \mathsf {\um@usv@ith} {"1D5C1}
1384     }{
1385       \um_set_mathalphabet_char:Nnn \mathsf {\um@usv@ith} {"1D629}
1386     }
1387   }
1388 }
1389 \cs_new:Npn \um_config_mathsf_num: {

```

```

1390 \um_set_mathalphabet_numbers:Nnn{\mathsf}{\um@usv@upnum}{\um@usv@sfnum}
1391 }

```

9.1.7 Sans serif upright: `\mathsfup`

```

1392 \cs_new:Npn \um_config_mathsfup_num: {
1393   \um_set_mathalphabet_numbers:Nnn{\mathsfup}{\um@usv@upnum}{\um@usv@sfnum}
1394 }
1395 \cs_new:Npn \um_config_mathsfup_latin: {
1396   \um_set_mathalphabet_latin:Nnn{\mathsfup}{\um@usv@uplatin,\um@usv@itlatin}{\um@usv@sfuplat}
1397   \um_set_mathalphabet_char:Nnn \mathsfup {\um@usv@ith} {"1D5C1}
1398 }
1399 \cs_new:Npn \um_config_mathsfup_Latin: {
1400   \um_set_mathalphabet_latin:Nnn{\mathsfup}{\um@usv@upLatin,\um@usv@itLatin}{\um@usv@sfupLat}
1401 }

```

9.1.8 Sans serif italic: `\mathsfit`

Map the numbers like that because it seems sensible.

```

1402 \cs_new:Npn \um_config_mathsfit_num: {
1403   \um_set_mathalphabet_numbers:Nnn{\mathsfit}{\um@usv@upnum}{\um@usv@sfnum}
1404 }
1405 \cs_new:Npn \um_config_mathsfit_Latin: {
1406   \um_set_mathalphabet_latin:Nnn{\mathsfit}{\um@usv@upLatin,\um@usv@itLatin}{\um@usv@sfitLat}
1407 }
1408 \cs_new:Npn \um_config_mathsfit_latin: {
1409   \um_set_mathalphabet_latin:Nnn{\mathsfit}{\um@usv@uplatin,\um@usv@itlatin}{\um@usv@sfitlat}
1410   \um_set_mathalphabet_char:Nnn \mathsfit {\um@usv@ith} {"1D629}
1411 }

```

9.1.9 Typewriter or monospaced: `\mathtt`

```

1412 \cs_new:Npn \um_config_mathtt_num: {
1413   \um_set_mathalphabet_numbers:Nnn{\mathtt}{\um@usv@upnum}{\um@usv@ttnum}
1414 }
1415 \cs_new:Npn \um_config_mathtt_Latin: {
1416   \um_set_mathalphabet_latin:Nnn{\mathtt}{\um@usv@upLatin,\um@usv@itLatin}{\um@usv@ttLatin}
1417 }
1418 \cs_new:Npn \um_config_mathtt_latin: {
1419   \um_set_mathalphabet_latin:Nnn{\mathtt}{\um@usv@uplatin,\um@usv@itlatin}{\um@usv@ttlatin}
1420   \um_set_mathalphabet_char:Nnn \mathtt {\um@usv@ith} {"1D691}
1421 }

```

9.2 Bold math alphabets

9.2.1 Bold: `\mathbf`

```

1422 \cs_new:Npn \um_config_mathbf_num: {
1423   \um_set_mathalphabet_numbers:Nnn{\mathbf}{\um@usv@upnum}{\um@usv@bfnum}

```

```

1424 }
1425 \cs_new:Npn \um_config_mathbf_Latin: {
1426   \if@um@bfliteral
1427     \um_set_mathalphabet_latin:Nnn{\mathbf}{\um@usv@upLatin}{\um@usv@bfupLatin}
1428     \um_set_mathalphabet_latin:Nnn{\mathbf}{\um@usv@itLatin}{\um@usv@bfitLatin}
1429   \else
1430     \um_set_mathalphabet_latin:Nnn{\mathbf}{\um@usv@upLatin,\um@usv@itLatin}{\um_bf_Latin_up_
1431   \fi
1432 }
1433 \cs_new:Npn \um_config_mathbf_latin: {
1434   \if@um@bfliteral
1435     \um_set_mathalphabet_latin:Nnn{\mathbf}{\um@usv@uplatin}{\um@usv@bfuplatin}
1436     \um_set_mathalphabet_latin:Nnn{\mathbf}{\um@usv@itlatin}{\um@usv@bfitlatin}
1437     \um_set_mathalphabet_char:Nnn{\mathbf}{\um@usv@ith}{\um@usv@bfith}
1438   \else
1439     \um_set_mathalphabet_latin:Nnn{\mathbf}{\um@usv@uplatin,\um@usv@itlatin}{\um_bf_latin_up_
1440     \if@um@bfuplatin
1441       \um_set_mathalphabet_char:Nnn{\mathbf}{\um@usv@ith}{\um@usv@bfuph}
1442     \else
1443       \um_set_mathalphabet_char:Nnn{\mathbf}{\um@usv@ith}{\um@usv@bfith}
1444     \fi
1445   \fi
1446 }
1447 \cs_new:Npn \um_config_mathbf_Greek: {
1448   \um_set_mathalphabet_char:Nnn{\mathbf}{\um@usv@Digamma}{\um@usv@bfDigamma}
1449   \if@um@bfliteral
1450     \um_set_mathalphabet_greek:Nnn{\mathbf}{\um@usv@upGreek}{\um@usv@bfupGreek}
1451     \um_set_mathalphabet_greek:Nnn{\mathbf}{\um@usv@itGreek}{\um@usv@bfitGreek}
1452     \um_set_mathalphabet_char:Nnn{\mathbf}{\um@usv@varTheta}{\um@usv@bfvarTheta}
1453     \um_set_mathalphabet_char:Nnn{\mathbf}{\um@usv@itvarTheta}{\um@usv@bfitvarTheta}
1454     \um_set_mathalphabet_char:Nnn{\mathbf}{\um@usv@Nabla}{\um@usv@bfNabla}
1455     \um_set_mathalphabet_char:Nnn{\mathbf}{\um@usv@itNabla}{\um@usv@bfitNabla}
1456   \else
1457     \um_set_mathalphabet_greek:Nnn{\mathbf}{\um@usv@upGreek,\um@usv@itGreek}{\um_bf_Greek_up_
1458     \um_set_mathalphabet_char:Nnn{\mathbf}{\um@usv@Nabla,\um@usv@itNabla}{\um_bfNabla_up_or_i
1459     \if@um@bfupGreek
1460       \um_set_mathalphabet_char:Nnn{\mathbf}{\um@usv@varTheta,\um@usv@itvarTheta}{\um@usv@bfva
1461     \else
1462       \um_set_mathalphabet_char:Nnn{\mathbf}{\um@usv@varTheta,\um@usv@itvarTheta}{\um@usv@bfit
1463     \fi
1464   \fi
1465 }
1466 \cs_new:Npn \um_config_mathbf_greek: {
1467   \um_set_mathalphabet_char:Nnn{\mathbf}{\um@usv@digamma}{\um@usv@bfdigamma}
1468   \if@um@bfliteral
1469     \um_set_mathalphabet_greek:Nnn{\mathbf}{\um@usv@upgreek}{\um@usv@bfupgreek}

```

```

1470 \um_set_mathalphabet_greek:Nnn{\mathbf}{\um@usv@itgreek}{\um@usv@bfitgreek}
1471 \um_set_mathalphabet_char:Nnn{\mathbf}{\um@usv@partial}{\um@usv@bfpartial}
1472 \um_set_mathalphabet_char:Nnn{\mathbf}{\um@usv@varepsilon}{\um@usv@bfvarepsilon}
1473 \um_set_mathalphabet_char:Nnn{\mathbf}{\um@usv@vartheta}{\um@usv@bfvartheta}
1474 \um_set_mathalphabet_char:Nnn{\mathbf}{\um@usv@varkappa}{\um@usv@bfvarkappa}
1475 \um_set_mathalphabet_char:Nnn{\mathbf}{\um@usv@varphi}{\um@usv@bfvarphi}
1476 \um_set_mathalphabet_char:Nnn{\mathbf}{\um@usv@varrho}{\um@usv@bfvarrho}
1477 \um_set_mathalphabet_char:Nnn{\mathbf}{\um@usv@varpi}{\um@usv@bfvarpi}
1478 \um_set_mathalphabet_char:Nnn{\mathbf}{\um@usv@itpartial}{\um@usv@bfitpartial}
1479 \um_set_mathalphabet_char:Nnn{\mathbf}{\um@usv@itvarepsilon}{\um@usv@bfitvarepsilon}
1480 \um_set_mathalphabet_char:Nnn{\mathbf}{\um@usv@itvartheta}{\um@usv@bfitvartheta}
1481 \um_set_mathalphabet_char:Nnn{\mathbf}{\um@usv@itvarkappa}{\um@usv@bfitvarkappa}
1482 \um_set_mathalphabet_char:Nnn{\mathbf}{\um@usv@itvarphi}{\um@usv@bfitvarphi}
1483 \um_set_mathalphabet_char:Nnn{\mathbf}{\um@usv@itvarrho}{\um@usv@bfitvarrho}
1484 \um_set_mathalphabet_char:Nnn{\mathbf}{\um@usv@itvarpi}{\um@usv@bfitvarpi}
1485 \else
1486 \um_set_mathalphabet_greek:Nnn{\mathbf}{\um@usv@upgreek,\um@usv@itgreek}{\um@bf_greek_up_
1487 \if@um@bfupgreek
1488 \um_set_mathalphabet_char:Nnn{\mathbf}{\um@usv@varepsilon,\um@usv@itvarepsilon}{\um@usv@
1489 \um_set_mathalphabet_char:Nnn{\mathbf}{\um@usv@vartheta,\um@usv@itvartheta}{\um@usv@bfv
1490 \um_set_mathalphabet_char:Nnn{\mathbf}{\um@usv@varkappa,\um@usv@itvarkappa}{\um@usv@bfv
1491 \um_set_mathalphabet_char:Nnn{\mathbf}{\um@usv@varphi,\um@usv@itvarphi}{\um@usv@bfvarphi}
1492 \um_set_mathalphabet_char:Nnn{\mathbf}{\um@usv@varrho,\um@usv@itvarrho}{\um@usv@bfvarrho}
1493 \um_set_mathalphabet_char:Nnn{\mathbf}{\um@usv@varpi,\um@usv@itvarpi}{\um@usv@bfvarpi}
1494 \else
1495 \um_set_mathalphabet_char:Nnn{\mathbf}{\um@usv@varepsilon,\um@usv@itvarepsilon}{\um@usv@
1496 \um_set_mathalphabet_char:Nnn{\mathbf}{\um@usv@vartheta,\um@usv@itvartheta}{\um@usv@bfv
1497 \um_set_mathalphabet_char:Nnn{\mathbf}{\um@usv@varkappa,\um@usv@itvarkappa}{\um@usv@bfv
1498 \um_set_mathalphabet_char:Nnn{\mathbf}{\um@usv@varphi,\um@usv@itvarphi}{\um@usv@bfvarphi}
1499 \um_set_mathalphabet_char:Nnn{\mathbf}{\um@usv@varrho,\um@usv@itvarrho}{\um@usv@bfvarrho}
1500 \um_set_mathalphabet_char:Nnn{\mathbf}{\um@usv@varpi,\um@usv@itvarpi}{\um@usv@bfvarpi}
1501 \fi
1502 \um_set_mathalphabet_char:Nnn{\mathbf}{\um@usv@partial,\um@usv@itpartial}{\um@bfpartial_u
1503 \fi
1504 }

```

9.2.2 Bold Italic: `\mathbfi`

```

1505 \cs_new:Npn \um_config_mathbf_i_num: {
1506 \um_set_mathalphabet_numbers:Nnn{\mathbfi}{\um@usv@upnum}{\um@usv@bfnum}
1507 }
1508 \cs_new:Npn \um_config_mathbf_i_Latin: {
1509 \um_set_mathalphabet_latin:Nnn{\mathbfi}{\um@usv@upLatin,\um@usv@itLatin}{\um@usv@bfitLat
1510 }
1511 \cs_new:Npn \um_config_mathbf_i_latin: {
1512 \um_set_mathalphabet_latin:Nnn{\mathbfi}{\um@usv@uplatin,\um@usv@itlatin}{\um@usv@bfitlat
1513 \um_set_mathalphabet_char:Nnn{\mathbfi}{\um@usv@ith} {"1D489}

```



```

1514 }
1515 \cs_new:Npn \um_config_mathbfit_Greek: {
1516   \um_set_mathalphabet_greek:Nnn{\mathbfit}{\um@usv@upGreek,\um@usv@itGreek}{\um@usv@bfitGre
1517   \um_set_mathalphabet_char:Nnn{\mathbfit}{\um@usv@varTheta,\um@usv@itvarTheta}{\um@usv@bfit
1518   \um_set_mathalphabet_char:Nnn{\mathbfit}{\um@usv@Nabla,\um@usv@itNabla}{\um@usv@bfitNabla}
1519 }
1520 \cs_new:Npn \um_config_mathbfit_greek: {
1521   \um_set_mathalphabet_latin:Nnn{\mathbfit}{\um@usv@upLatin,\um@usv@itLatin}{\um@usv@bfitLat
1522   \um_set_mathalphabet_greek:Nnn{\mathbfit}{\um@usv@upgreek,\um@usv@itgreek}{\um@usv@bfitgre
1523   \um_set_mathalphabet_char:Nnn{\mathbfit}{\um@usv@partial,\um@usv@itpartial}{\um@usv@bfitpa
1524   \um_set_mathalphabet_char:Nnn{\mathbfit}{\um@usv@varepsilon,\um@usv@itvarepsilon}{\um@usv@
1525   \um_set_mathalphabet_char:Nnn{\mathbfit}{\um@usv@vartheta,\um@usv@itvartheta}{\um@usv@bfit
1526   \um_set_mathalphabet_char:Nnn{\mathbfit}{\um@usv@varkappa,\um@usv@itvarkappa}{\um@usv@bfit
1527   \um_set_mathalphabet_char:Nnn{\mathbfit}{\um@usv@varphi,\um@usv@itvarphi}{\um@usv@bfitvarp
1528   \um_set_mathalphabet_char:Nnn{\mathbfit}{\um@usv@varrho,\um@usv@itvarrho}{\um@usv@bfitvarr
1529   \um_set_mathalphabet_char:Nnn{\mathbfit}{\um@usv@varpi,\um@usv@itvarpi}{\um@usv@bfitvarpi}
1530 }

```

9.2.3 Bold Italic: \mathbf{fup}

```

1531 \cs_new:Npn \um_config_mathbfup_num: {
1532   \um_set_mathalphabet_numbers:Nnn{\mathbfup}{\um@usv@upnum}{\um@usv@bfnum}
1533 }
1534 \cs_new:Npn \um_config_mathbfup_Latin: {
1535   \um_set_mathalphabet_latin:Nnn{\mathbfup}{\um@usv@upLatin,\um@usv@itLatin}{\um@usv@bfupLat
1536 }
1537 \cs_new:Npn \um_config_mathbfup_latin: {
1538   \um_set_mathalphabet_latin:Nnn{\mathbfup}{\um@usv@uplatin,\um@usv@itlatin}{\um@usv@bfuplat
1539   \um_set_mathalphabet_char:Nnn{\mathbfup}{\um@usv@ith} {"1D421}
1540 }
1541 \cs_new:Npn \um_config_mathbfup_Greek: {
1542   \um_set_mathalphabet_greek:Nnn{\mathbfup}{\um@usv@upGreek,\um@usv@itGreek}{\um@usv@bfupGre
1543   \um_set_mathalphabet_char:Nnn{\mathbfup}{\um@usv@varTheta,\um@usv@itvarTheta}{\um@usv@bfva
1544   \um_set_mathalphabet_char:Nnn{\mathbfup}{\um@usv@Nabla,\um@usv@itNabla}{\um@usv@bfNabla}
1545   \um_set_mathalphabet_char:Nnn{\mathbfup}{\um@usv@Digamma}{\um@usv@bfDigamma}
1546 }
1547 \cs_new:Npn \um_config_mathbfup_greek: {
1548   \um_set_mathalphabet_greek:Nnn{\mathbfup}{\um@usv@upgreek,\um@usv@itgreek}{\um@usv@bfupgre
1549   \um_set_mathalphabet_char:Nnn{\mathbfup}{\um@usv@partial,\um@usv@itpartial}{\um@usv@bfpart
1550   \um_set_mathalphabet_char:Nnn{\mathbfup}{\um@usv@varepsilon,\um@usv@itvarepsilon}{\um@usv@
1551   \um_set_mathalphabet_char:Nnn{\mathbfup}{\um@usv@vartheta,\um@usv@itvartheta}{\um@usv@bfva
1552   \um_set_mathalphabet_char:Nnn{\mathbfup}{\um@usv@varkappa,\um@usv@itvarkappa}{\um@usv@bfva
1553   \um_set_mathalphabet_char:Nnn{\mathbfup}{\um@usv@varphi,\um@usv@itvarphi}{\um@usv@bfvarphi
1554   \um_set_mathalphabet_char:Nnn{\mathbfup}{\um@usv@varrho,\um@usv@itvarrho}{\um@usv@bfvarrho
1555   \um_set_mathalphabet_char:Nnn{\mathbfup}{\um@usv@varpi,\um@usv@itvarpi}{\um@usv@bfvarpi}
1556   \um_set_mathalphabet_char:Nnn{\mathbfup}{\um@usv@digamma}{\um@usv@bfdigamma}
1557 }

```

9.2.4 Bold fractur or fraktur or blackletter: `\mathbffrak`

```
1558 \cs_new:Npn \um_config_mathbffrak_Latin: {  
1559   \um_set_mathalphabet_latin:Nnn{\mathbffrak}{\um@usv@upLatin, \um@usv@itLatin}{\um@usv@bffrak}  
1560 }  
1561 \cs_new:Npn \um_config_mathbffrak_latin: {  
1562   \um_set_mathalphabet_latin:Nnn{\mathbffrak}{\um@usv@uplatin, \um@usv@itlatin}{\um@usv@bffrak}  
1563   \um_set_mathalphabet_char:Nnn{\mathbffrak}{\um@usv@ith} {"1D58D}  
1564 }
```

9.2.5 Bold script or calligraphic: `\mathbfscr`

```
1565 \cs_new:Npn \um_config_mathbfscr_Latin: {  
1566   \um_set_mathalphabet_latin:Nnn{\mathbfscr}{\um@usv@upLatin, \um@usv@itLatin}{\um@usv@bfscrL}  
1567 }  
1568 \cs_new:Npn \um_config_mathbfscr_latin: {  
1569   \um_set_mathalphabet_latin:Nnn{\mathbfscr}{\um@usv@uplatin, \um@usv@itlatin}{\um@usv@bfscrL}  
1570   \um_set_mathalphabet_char:Nnn{\mathbfscr}{\um@usv@ith} {"1D4F1}  
1571 }
```

9.2.6 Bold sans serif: `\mathbfsf`

These use the sans-style settings rather than bold-style. Numbers (always upright) and letters:

```
1572 \cs_new:Npn \um_config_mathbfsf_num: {  
1573   \um_set_mathalphabet_numbers:Nnn \mathbfsf {\um@usv@upnum}{\um@usv@bfsfnum}  
1574 }  
1575 \cs_new:Npn \um_config_mathbfsf_Latin: {  
1576   \bool_if:NTF \g_um_sfliteral_bool {  
1577     \um_set_mathalphabet_latin:Nnn \mathbfsf {\um@usv@upLatin}{\um@usv@bfsfupLatin}  
1578     \um_set_mathalphabet_latin:Nnn \mathbfsf {\um@usv@itLatin}{\um@usv@bfsfitLatin}  
1579   }{  
1580     \um_set_mathalphabet_latin:Nnn \mathbfsf {\um@usv@upLatin, \um@usv@itLatin}{\um@usv@bfsf_Lati}  
1581   }  
1582 }  
1583 \cs_new:Npn \um_config_mathbfsf_latin: {  
1584   \bool_if:NTF \g_um_upsans_bool {  
1585     \um_set_mathalphabet_latin:Nnn \mathbfsf {\um@usv@uplatin}{\um@usv@bfsfuplatin}  
1586     \um_set_mathalphabet_latin:Nnn \mathbfsf {\um@usv@itlatin}{\um@usv@bfsfitlatin}  
1587     \um_set_mathalphabet_char:Nnn \mathbfsf {\um@usv@ith} {"1D65D}  
1588   }{  
1589     \um_set_mathalphabet_latin:Nnn \mathbfsf {\um@usv@uplatin, \um@usv@itlatin}{\um@usv@bfsf_Lati}  
1590     \bool_if:NTF \g_um_upsans_bool {  
1591       \um_set_mathalphabet_char:Nnn \mathbfsf {\um@usv@ith} {"1D5F5}  
1592     }{  
1593       \um_set_mathalphabet_char:Nnn \mathbfsf {\um@usv@ith} {"1D65D}  
1594     }  
1595   }
```

```

1596 }
1597 \cs_new:Npn \um_config_mathbfsf_Greek: {
1598   \bool_if:NTF \g_um_sfliteral_bool {
1599     \um_set_mathalphabet_greek:Nnn \mathbfsf {\um@usv@upGreek}{\um@usv@bfsfupGreek}
1600     \um_set_mathalphabet_greek:Nnn \mathbfsf {\um@usv@itGreek}{\um@usv@bfsfitGreek}
1601     \um_set_mathalphabet_char:Nnn \mathbfsf {\um@usv@varTheta}{\um@usv@bfsfupvarTheta}
1602     \um_set_mathalphabet_char:Nnn \mathbfsf {\um@usv@itvarTheta}{\um@usv@bfsfitvarTheta}
1603     \um_set_mathalphabet_char:Nnn \mathbfsf {\um@usv@Nabla}{\um@usv@bfsfNabla}
1604     \um_set_mathalphabet_char:Nnn \mathbfsf {\um@usv@itNabla}{\um@usv@bfsfitNabla}
1605   }{
1606     \um_set_mathalphabet_greek:Nnn \mathbfsf {\um@usv@upGreek,\um@usv@itGreek}{\um_bfsf_Gree
1607     \um_set_mathalphabet_char:Nnn \mathbfsf {\um@usv@Nabla,\um@usv@itNabla}{\um_bfsfNabla_up
1608     \bool_if:NTF \g_um_upsans_bool {
1609       \um_set_mathalphabet_char:Nnn \mathbfsf {\um@usv@varTheta,\um@usv@itvarTheta}{ "1D767}
1610     }{
1611       \um_set_mathalphabet_char:Nnn \mathbfsf {\um@usv@varTheta,\um@usv@itvarTheta}{ "1D7A1}
1612     }
1613   }
1614 }
1615 \cs_new:Npn \um_config_mathbfsf_greek: {
1616   \bool_if:NTF \g_um_sfliteral_bool {
1617     \um_set_mathalphabet_greek:Nnn \mathbfsf {\um@usv@upgreek}{\um@usv@bfsfupgreek}
1618     \um_set_mathalphabet_greek:Nnn \mathbfsf {\um@usv@itgreek}{\um@usv@bfsfitgreek}
1619     \um_set_mathalphabet_char:Nnn \mathbfsf {\um@usv@partial}{\um@usv@bfsfpartial}
1620     \um_set_mathalphabet_char:Nnn \mathbfsf {\um@usv@varepsilon}{ "1D78A}
1621     \um_set_mathalphabet_char:Nnn \mathbfsf {\um@usv@vartheta}{ "1D78B}
1622     \um_set_mathalphabet_char:Nnn \mathbfsf {\um@usv@varkappa}{ "1D78C}
1623     \um_set_mathalphabet_char:Nnn \mathbfsf {\um@usv@varphi}{ "1D78D}
1624     \um_set_mathalphabet_char:Nnn \mathbfsf {\um@usv@varrho}{ "1D78E}
1625     \um_set_mathalphabet_char:Nnn \mathbfsf {\um@usv@varpi}{ "1D78F}
1626     \um_set_mathalphabet_char:Nnn \mathbfsf {\um@usv@itpartial}{\um@usv@bfsfitpartial}
1627     \um_set_mathalphabet_char:Nnn \mathbfsf {\um@usv@itvarepsilon}{ "1D7C4}
1628     \um_set_mathalphabet_char:Nnn \mathbfsf {\um@usv@itvartheta}{ "1D7C5}
1629     \um_set_mathalphabet_char:Nnn \mathbfsf {\um@usv@itvarkappa}{ "1D7C6}
1630     \um_set_mathalphabet_char:Nnn \mathbfsf {\um@usv@itvarphi}{ "1D7C7}
1631     \um_set_mathalphabet_char:Nnn \mathbfsf {\um@usv@itvarrho}{ "1D7C8}
1632     \um_set_mathalphabet_char:Nnn \mathbfsf {\um@usv@itvarpi}{ "1D7C9}
1633   }{
1634     \um_set_mathalphabet_greek:Nnn \mathbfsf {\um@usv@upgreek,\um@usv@itgreek}{\um_bfsf_gree
1635     \um_set_mathalphabet_char:Nnn \mathbfsf {\um@usv@partial,\um@usv@itpartial}{\um_bfsfpart
1636     \bool_if:NTF \g_um_upsans_bool {
1637       \um_set_mathalphabet_char:Nnn \mathbfsf {\um@usv@varepsilon,\um@usv@itvarepsilon}{ "1D7
1638       \um_set_mathalphabet_char:Nnn \mathbfsf {\um@usv@vartheta,\um@usv@itvartheta}{ "1D78B}
1639       \um_set_mathalphabet_char:Nnn \mathbfsf {\um@usv@varkappa,\um@usv@itvarkappa}{ "1D78C}
1640       \um_set_mathalphabet_char:Nnn \mathbfsf {\um@usv@varphi,\um@usv@itvarphi}{ "1D78D}
1641       \um_set_mathalphabet_char:Nnn \mathbfsf {\um@usv@varrho,\um@usv@itvarrho}{ "1D78E}

```

```

1642 \um_set_mathalphabet_char:Nnn \mathbfsf {\um@usv@varpi,\um@usv@itvarpi}{ "1D78F}
1643 }{
1644 \um_set_mathalphabet_char:Nnn \mathbfsf {\um@usv@varepsilon,\um@usv@itvarepsilon}{ "1D78A}
1645 \um_set_mathalphabet_char:Nnn \mathbfsf {\um@usv@vartheta,\um@usv@itvartheta}{ "1D7C5}
1646 \um_set_mathalphabet_char:Nnn \mathbfsf {\um@usv@varkappa,\um@usv@itvarkappa}{ "1D7C6}
1647 \um_set_mathalphabet_char:Nnn \mathbfsf {\um@usv@varphi,\um@usv@itvarphi}{ "1D7C7}
1648 \um_set_mathalphabet_char:Nnn \mathbfsf {\um@usv@varrho,\um@usv@itvarrho}{ "1D7C8}
1649 \um_set_mathalphabet_char:Nnn \mathbfsf {\um@usv@varpi,\um@usv@itvarpi}{ "1D7C9}
1650 }
1651 }
1652 }

```

9.2.7 Bold upright sans serif: `\mathbfsfup`

```

1653 \cs_new:Npn \um_config_mathbfsfup_num: {
1654 \um_set_mathalphabet_numbers:Nnn{\mathbfsfup}{\um@usv@upnum}{\um@usv@bfsfnum}
1655 }
1656 \cs_new:Npn \um_config_mathbfsfup_Latin: {
1657 \um_set_mathalphabet_latin:Nnn{\mathbfsfup}{\um@usv@upLatin,\um@usv@itLatin}{\um@usv@bfsfup}
1658 }
1659 \cs_new:Npn \um_config_mathbfsfup_latin: {
1660 \um_set_mathalphabet_latin:Nnn{\mathbfsfup}{\um@usv@uplatin,\um@usv@itlatin}{\um@usv@bfsfup}
1661 \um_set_mathalphabet_char:Nnn \mathbfsfup {\um@usv@ith} { "1D5F5}
1662 }
1663 \cs_new:Npn \um_config_mathbfsfup_Greek: {
1664 \um_set_mathalphabet_greek:Nnn{\mathbfsfup}{\um@usv@upGreek,\um@usv@itGreek}{\um@usv@bfsfup}
1665 \um_set_mathalphabet_char:Nnn{\mathbfsfup}{\um@usv@varTheta,\um@usv@itvarTheta}{ "1D767}
1666 \um_set_mathalphabet_char:Nnn{\mathbfsfup}{\um@usv@Nabla,\um@usv@itNabla}{ "1D76F}
1667 }
1668 \cs_new:Npn \um_config_mathbfsfup_greek: {
1669 \um_set_mathalphabet_greek:Nnn{\mathbfsfup}{\um@usv@upgreek,\um@usv@itgreek}{\um@usv@bfsfup}
1670 \um_set_mathalphabet_char:Nnn{\mathbfsfup}{\um@usv@partial,\um@usv@itpartial}{ "1D789}
1671 \um_set_mathalphabet_char:Nnn{\mathbfsfup}{\um@usv@varepsilon,\um@usv@itvarepsilon}{ "1D78A}
1672 \um_set_mathalphabet_char:Nnn{\mathbfsfup}{\um@usv@vartheta,\um@usv@itvartheta}{ "1D78B}
1673 \um_set_mathalphabet_char:Nnn{\mathbfsfup}{\um@usv@varkappa,\um@usv@itvarkappa}{ "1D78C}
1674 \um_set_mathalphabet_char:Nnn{\mathbfsfup}{\um@usv@varphi,\um@usv@itvarphi}{ "1D78D}
1675 \um_set_mathalphabet_char:Nnn{\mathbfsfup}{\um@usv@varrho,\um@usv@itvarrho}{ "1D78E}
1676 \um_set_mathalphabet_char:Nnn{\mathbfsfup}{\um@usv@varpi,\um@usv@itvarpi}{ "1D78F}
1677 }

```

9.2.8 Bold italic sans serif: `\mathbfsfit`

```

1678 \cs_new:Npn \um_config_mathbfsfit_num: {
1679 \um_set_mathalphabet_numbers:Nnn{\mathbfsfit}{\um@usv@upnum}{\um@usv@bfsfnum}
1680 }
1681 \cs_new:Npn \um_config_mathbfsfit_Latin: {
1682 \um_set_mathalphabet_latin:Nnn{\mathbfsfit}{\um@usv@upLatin,\um@usv@itLatin}{\um@usv@bfsfit}
1683 }

```

```

1684 \cs_new:Npn \um_config_mathbfsfit_latin: {
1685   \um_set_mathalphabet_latin:Nnn{\mathbfsfit}{\um@usv@uplatin,\um@usv@itlatin}{\um@usv@bfsfi
1686   \um_set_mathalphabet_char:Nnn \mathbfsfit {\um@usv@ith} {"1D65D}
1687 }
1688 \cs_new:Npn \um_config_mathbfsfit_greek: {
1689   \um_set_mathalphabet_greek:Nnn{\mathbfsfit}{\um@usv@upgreek,\um@usv@itgreek}{\um@usv@bfsfi
1690   \um_set_mathalphabet_char:Nnn{\mathbfsfit}{\um@usv@varTheta}{"1D7A1}
1691   \um_set_mathalphabet_char:Nnn{\mathbfsfit}{\um@usv@Nabla,\um@usv@itNabla}{\um@usv@bfsfitNa
1692 }
1693 \cs_new:Npn \um_config_mathbfsfit_greek: {
1694   \um_set_mathalphabet_greek:Nnn{\mathbfsfit}{\um@usv@upgreek,\um@usv@itgreek}{\um@usv@bfsfi
1695   \um_set_mathalphabet_char:Nnn{\mathbfsfit}{\um@usv@partial,\um@usv@itpartial}{\um@usv@bfsfi
1696   \um_set_mathalphabet_char:Nnn{\mathbfsfit}{\um@usv@varepsilon,\um@usv@itvarepsilon}{\um@usv@bfsfi
1697   \um_set_mathalphabet_char:Nnn{\mathbfsfit}{\um@usv@vartheta,\um@usv@itvartheta}{\um@usv@bfsfi
1698   \um_set_mathalphabet_char:Nnn{\mathbfsfit}{\um@usv@varkappa,\um@usv@itvarkappa}{\um@usv@bfsfi
1699   \um_set_mathalphabet_char:Nnn{\mathbfsfit}{\um@usv@varphi,\um@usv@itvarphi}{\um@usv@bfsfi
1700   \um_set_mathalphabet_char:Nnn{\mathbfsfit}{\um@usv@varrho,\um@usv@itvarrho}{\um@usv@bfsfi
1701   \um_set_mathalphabet_char:Nnn{\mathbfsfit}{\um@usv@varpi,\um@usv@itvarpi}{\um@usv@bfsfi
1702 }

```

9.3 Definitions of the math symbols

Here we define every unicode math codepoint an equivalent macro name. The two are equivalent, in a `\let\xyz=^^^1234` kind of way.

`\um@scancharlet` We need to do some trickery to transform the `\UnicodeMathSymbol` argument
`\um@scanactivedef` "ABCDEF into the X_YTeX ‘caret input’ form `^^^^abcdef`. It is *very important* that the argument has five characters. Otherwise we need to change the number of ^ chars.

To do this, turn ^ into a regular ‘other’ character and define the macro to perform the lowercasing and `\let`. `\scantokens` changes the carets back into their original meaning after the group has ended and ^’s catcode returns to normal.

```

1703 \begin{group}
1704   \char_make_other:N \^
1705   \cs_gset:Npn \um@scancharlet#1="#2\@nil {
1706     \lowercase{
1707       \scantokens{\global\let#1=^^^^#2}
1708     }
1709   }

```

Making ^ the right catcode isn’t strictly necessary right now but it helps to future proof us with, e.g., `breqn`.

```

1710 \gdef\um@scanactivedef"#1\@nil#2{
1711   \lowercase{
1712     \tl_rescan:nn{
1713       \ExplSyntaxOn
1714       \char_make_math_superscript:N\^

```

```

1715     }{
1716     \global\def^^^^^#1{#2}
1717   }
1718 }
1719 }
1720 \endgroup

```

Now give `\UnicodeMathSymbol` a definition in terms of `\um@scancharlet` and we're good to go. Make sure `#` is an 'other' so that we don't get confused with `\mathoctothorpe`.

```

1721 \begingroup
1722 \def\UnicodeMathSymbol#1#2#3#4{
1723   \um@scancharlet#2=#1\@nil
1724 }
1725 \char_make_other:N \#
1726 \@input{unicode-math-table.tex}
1727 \endgroup

```

Fix `\backslash`:

```

1728 \group_begin:
1729   \lccode`*=`\\
1730   \char_make_escape:N \|
1731   \char_make_other:N \\
1732   |lowercase{
1733   |group_end:|let|backslash=*}

```

10 Epilogue

Lots of little things to tidy up.

10.0.1 Primes

We need a new 'prime' algorithm. Unicode math has four pre-drawn prime glyphs.

```

U+2032: PRIME (\primesingle): x'
U+2033: DOUBLE PRIME (\primedouble): x''
U+2034: TRIPLE PRIME (\primetriple): x'''
U+2057: QUADRUPLE PRIME (\primequadruple): x''''

```

As you can see, they're all drawn at the correct height without being super-scripted. However, in a correctly behaviour OpenType font with the MATH table, we also see different behaviour after the `sssty` feature is applied:

```

U+2032: PRIME in the 'scriptstyle' font: x'

```

The shrinking and offsetting is done as it is turned into a superscript. This means, luckily, that by default things work nicely for single primes. We can write `x\primesingle` or `x^\primesingle` and get: x' and x' . To support single primes, then, things are easier than in \LaTeX ; we can just map ' to `\prime` and not worry about it.

However, it would be nice to use the pre-composed primes above if they exist in the font; consider x''' vs. x''' . Our algorithm is

- Prime encountered; pcount=1.
- Scan ahead; if prime: pcount:=pcount+1; repeat.
- If not prime, stop scanning.
- If pcount=1, `\prime`, end.
- If pcount=2, check `\primedouble`; if it exists, use it, end; if not, goto last step.
- Ditto pcount=3 & `\primetripel`.
- Ditto pcount=4 & `\primequadruple`.
- If pcount>4 or the glyph doesn't exist, insert pcount `\primes` with `\primekern` between each.

```

1734 \muskip_new:N \g_um_primekern_muskip
1735 \muskip_gset:Nn \g_um_primekern_muskip { -\thinmuskip/2 }% arbitrary
1736 \num_new:N \l_um_primecount_num
1737 \cs_new:Nn \um_nprimes:n {
1738   ^{
1739     \primesingle
1740     \prg_replicate:nn {#1-1} { \mskip \g_um_primekern_muskip \primesingle }
1741   }
1742 }
1743 \cs_new:Nn \um_nprimes_select:n {
1744   \prg_case_int:nnn {#1}{
1745     {1} { ^{\primesingle} }
1746     {2} {
1747       \um_glyph_if_exist:nTF {"2033} { ^{\primedouble} } {\um_nprimes:n {#1}}
1748     }
1749     {3} {
1750       \um_glyph_if_exist:nTF {"2034} { ^{\primetripel} } {\um_nprimes:n {#1}}
1751     }
1752     {4} {
1753       \um_glyph_if_exist:nTF {"2057} { ^{\primequadruple} } {\um_nprimes:n {#1}}
1754     }
1755   }{
1756     \um_nprimes:n {#1}

```

```

1757 }
1758 }

```

Scanning is more annoying than you'd think because we want to support all three of `\prime`, `'`, and the unicode prime. And `\ifx` doesn't work with mathactive chars.

```

1759 \cs_new:Nn \um_scanprime: {
1760   \num_zero:N \l_um_primecount_num
1761   \um_scanprime_collect:
1762 }
1763 \cs_new:Nn \um_scanprime_collect: {
1764   \num_incr:N \l_um_primecount_num
1765   \peek_meaning_remove:NTF ' {
1766     \um_scanprime_collect:
1767   }{
1768     \peek_meaning_remove:NTF \um_scanprime: {
1769       \um_scanprime_collect:
1770     }{
1771       \peek_meaning_remove:NTF ^^^^2032 {
1772         \um_scanprime_collect:
1773       }{
1774         \um_nprimes_select:n {\l_um_primecount_num}
1775       }
1776     }
1777   }
1778 }

1779 \cs_set_eq:NN \prime \um_scanprime:
1780 \group_begin:
1781   \char_make_active:N \'
1782   \char_make_active:n {"2032}
1783   \cs_gset_eq:NN ' \um_scanprime:
1784   \cs_gset_eq:NN ^^^^2032 \um_scanprime:
1785 \group_end:

```

10.0.2 Unicode radicals

Undo the damage made to `\sqrt`:

```

1786 \DeclareRobustCommand\sqrt{\@ifnextchar[\@sqrt\sqrtsign}

```

`\r@@t` #1 : A mathstyle (for `\mathpalette`)

#2 : Leading superscript for the sqrt sign

A re-implementation of L^AT_EX's hard-coded n-root sign using the appropriate `\fontdimens`.

```

1787 \def\r@@t#1#2{
1788   \setbox\z@\hbox{$\m@th #1\sqrtsign{#2}$}

```



```

1789 \um@scaled@apply{#1}{\kern}{\fontdimen63\um@font}
1790 \raise \dimexpr(
1791     \um@fontdimen@percent{65}{\um@font}\ht\z@-
1792     \um@fontdimen@percent{65}{\um@font}\dp\z@
1793 )\relax
1794 \copy \rootbox
1795 \um@scaled@apply{#1}{\kern}{\fontdimen64\um@font}
1796 \box \z@
1797 }

```

10.0.3 Unicode sub- and super-scripts

The idea here is to enter a scanning state after a superscript or subscript is encountered. If subsequent superscripts or subscripts (resp.) are found, they are lumped together. Each sub/super has a corresponding regular size glyph which is used by \XeTeX to typeset the results; this means that the actual subscript/superscript glyphs are never seen in the output document — they are only used as input characters.

Open question: should the superscript-like ‘modifiers’ (U+1D2C: MODIFIER CAPITAL LETTER A and on) be included here?

First, the setup of each mathactive char:

```

1798 \prop_new:N \g_um_supers_prop
1799 \prop_new:N \g_um_subs_prop
1800 \cs_generate_variant:Nn \prop_gput:Nnn {Nxn}
1801 \cs_generate_variant:Nn \prop_get:NnN {cxN}
1802 \cs_generate_variant:Nn \prop_if_in:NnTF {cx}
1803
1804 \group_begin:
1805
1806 % Populate a property list with superscript characters; their mean-
1807 % ing as their key,
1808 % for reasons that will become apparent soon, and their replace-
1809 % ment as each key's value.
1810 % Then make the superscript active and bind it to the scanning function.
1811 %
1812 % \cs{scantokens} makes this process much simpler since we can acti-
1813 % vate the char
1814 % and assign its meaning in one step.
1815 \cs_set:Nn \um_setup_active_superscript:nn {
1816     \prop_gput:Nxn \g_um_supers_prop {\meaning #1} {#2}
1817     \char_make_active:n {`#1}
1818     \global\XeTeXmathcodenum `#1 = "1FFFFF \scan_stop:
1819     \scantokens{
1820         \cs_gset:Npn #1 {
1821             \tl_set:Nn \l_um_ss_chain_tl {#2}
1822             \cs_set_eq:NN \um_sub_or_super:n \sp

```

```

1820     \tl_set:Nn \l_um_tmpa_tl {supers}
1821     \um_scan_ssript:
1822   }
1823 }
1824 }
1825
1826 \um_setup_active_superscript:nn {^^^2070} {0}
1827 \um_setup_active_superscript:nn {^^^00b9} {1}
1828 \um_setup_active_superscript:nn {^^^00b2} {2}
1829 \um_setup_active_superscript:nn {^^^00b3} {3}
1830 \um_setup_active_superscript:nn {^^^2074} {4}
1831 \um_setup_active_superscript:nn {^^^2075} {5}
1832 \um_setup_active_superscript:nn {^^^2076} {6}
1833 \um_setup_active_superscript:nn {^^^2077} {7}
1834 \um_setup_active_superscript:nn {^^^2078} {8}
1835 \um_setup_active_superscript:nn {^^^2079} {9}
1836 \um_setup_active_superscript:nn {^^^207a} {+}
1837 \um_setup_active_superscript:nn {^^^207b} {-}
1838 \um_setup_active_superscript:nn {^^^207c} {=}
1839 \um_setup_active_superscript:nn {^^^207d} {(}
1840 \um_setup_active_superscript:nn {^^^207e} {)}
1841 \um_setup_active_superscript:nn {^^^2071} {i}
1842 \um_setup_active_superscript:nn {^^^207f} {n}
1843
1844 % Ditto above.
1845 \cs_set:Nn \um_setup_active_subscript:nn {
1846   \prop_gput:Nxn \g_um_subs_prop {\meaning #1} {#2}
1847   \char_make_active:n {\`#1}
1848   \global\XeTeXmathcodenum `#1 = "1FFFFF \scan_stop:
1849   \scantokens{
1850     \cs_gset:Npn #1 {
1851       \tl_set:Nn \l_um_ss_chain_tl {#2}
1852       \cs_set_eq:NN \um_sub_or_super:n \sb
1853       \tl_set:Nn \l_um_tmpa_tl {subs}
1854       \um_scan_ssript:
1855     }
1856   }
1857 }
1858
1859 \um_setup_active_subscript:nn {^^^2080} {0}
1860 \um_setup_active_subscript:nn {^^^2081} {1}
1861 \um_setup_active_subscript:nn {^^^2082} {2}
1862 \um_setup_active_subscript:nn {^^^2083} {3}
1863 \um_setup_active_subscript:nn {^^^2084} {4}
1864 \um_setup_active_subscript:nn {^^^2085} {5}
1865 \um_setup_active_subscript:nn {^^^2086} {6}

```

```

1866 \um_setup_active_subscript:nn {^^^^2087} {7}
1867 \um_setup_active_subscript:nn {^^^^2088} {8}
1868 \um_setup_active_subscript:nn {^^^^2089} {9}
1869 \um_setup_active_subscript:nn {^^^^208a} {+}
1870 \um_setup_active_subscript:nn {^^^^208b} {-}
1871 \um_setup_active_subscript:nn {^^^^208c} {=}
1872 \um_setup_active_subscript:nn {^^^^208d} {(}
1873 \um_setup_active_subscript:nn {^^^^208e} {)}
1874 \um_setup_active_subscript:nn {^^^^2090} {a}
1875 \um_setup_active_subscript:nn {^^^^2091} {e}
1876 \um_setup_active_subscript:nn {^^^^1d62} {i}
1877 \um_setup_active_subscript:nn {^^^^2092} {o}
1878 \um_setup_active_subscript:nn {^^^^1d63} {r}
1879 \um_setup_active_subscript:nn {^^^^1d64} {u}
1880 \um_setup_active_subscript:nn {^^^^1d65} {v}
1881 \um_setup_active_subscript:nn {^^^^2093} {x}
1882 \um_setup_active_subscript:nn {^^^^1d66} {\beta}
1883 \um_setup_active_subscript:nn {^^^^1d67} {\gamma}
1884 \um_setup_active_subscript:nn {^^^^1d68} {\rho}
1885 \um_setup_active_subscript:nn {^^^^1d69} {\phi}
1886 \um_setup_active_subscript:nn {^^^^1d6a} {\chi}
1887
1888 \group_end:
1889
1890 % The scanning command, evident in its purpose:
1891 \cs_new:Nn \um_scan_sscript: {
1892   \um_scan_sscript:TF {
1893     \um_scan_sscript:
1894   }{
1895     \um_sub_or_super:n {\l_um_ss_chain_tl}
1896   }
1897 }
1898
1899 % The main theme here is stolen from the source to the vari-
1900 % ous \cs{peek_} functions.
1901 % Consider this function as simply boilerplate:
1902 \cs_new:Nn \um_scan_sscript:TF {
1903   \tl_set:Nx \l_peek_true_aux_tl { \exp_not:n{ #1 } }
1904   \tl_set_eq:NN \l_peek_true_tl \c_peek_true_remove_next_tl
1905   \tl_set:Nx \l_peek_false_tl { \exp_not:n{ \group_align_safe_end: #2 } }
1906   \group_align_safe_begin:
1907   \peek_after:NN \um_peek_execute_branches_ss:
1908 }
1909
1909 % We do not skip spaces when scanning ahead, and we explicitly wish to
1910 % bail out on encountering a space or a brace.

```

```

1911 \cs_new:Npn \um_peek_execute_branches_ss: {
1912   \bool_if:nTF {
1913     \token_if_eq_catcode_p:NN \l_peek_token \c_group_begin_token ||
1914     \token_if_eq_catcode_p:NN \l_peek_token \c_group_end_token ||
1915     \token_if_eq_meaning_p:NN \l_peek_token \c_space_token
1916   }
1917   { \l_peek_false_tl }
1918   { \um_peek_execute_branches_ss_aux: }
1919 }
1920
1921 % This is the actual comparison code.
1922 % Because the peeking has already tokenised the next token,
1923 % it's too late to extract its charcode directly. Instead,
1924 % we look at its meaning, which remains a `character' even
1925 % though it is itself math-active. If the character is ever
1926 % made fully active, this will break our assumptions!
1927 %
1928 % If the char's meaning exists as a property list key, we
1929 % build up a chain of sub-/superscripts and iterate. (If not, exit and
1930 % typeset what we've already collected.)
1931 \cs_new:Nn \um_peek_execute_branches_ss_aux: {
1932   \prop_if_in:cxTF
1933     {g_um_\l_um_tmpa_tl _prop}
1934     {\meaning\l_peek_token}
1935   {
1936     \prop_get:cxN
1937       {g_um_\l_um_tmpa_tl _prop}
1938       {\meaning\l_peek_token}
1939     \l_um_tmpb_tl
1940     \tl_put_right:NV \l_um_ss_chain_tl \l_um_tmpb_tl
1941     \l_peek_true_tl
1942   }
1943   {\l_peek_false_tl}
1944 }

```

10.0.4 Synonyms and all the rest

We need to change L^AT_EX's idea of the font used to typeset things like `\sin` and `\cos`:

```

1945 \def\operator@font{\um_setup_mathup:}
1946 \def\to{\rightarrow}
1947 \def\vec{\overrightarrow}
1948 \def\le{\leq}
1949 \def\ge{\geq}
1950 \def\neq{\neq}

```

Define `\colon` as a `mathpunct` `'\colon'`. This is wrong: it should be `U+003A: COLON` instead!

```

1951 \@@ifpackageloaded{amsmath}{
1952   % define their own colon, perhaps I should just steal it.
1953 }{
1954   \cs_set_protected:Npn \colon {
1955     \bool_if:NTF \g_um_literal_colon_bool {:\} { \mathpunct{:\} }
1956   }
1957 }

```

`\mathcal`

```

1958 \def\mathcal{\mathscr}

```

`\mathrm`

```

1959 \def\mathrm{\mathup}

```

10.0.5 Compatibility

Note that `amsmath` will always be loaded before `unicode-math`. (Conflicts occur if you try it the other way around.)

- Since the `mathcode` of `\-` is greater than eight bits, this piece of `\AtBeginDocument` code from `amsmath` dies if we try and set the `maths` font in the preamble:

```

1960 \@@ifpackageloaded{amsmath}{
1961   \tl_remove_in:Nn \@begindocumenthook {
1962     \mathchardef\std@minus\mathcode`\-\relax
1963     \mathchardef\std@equal\mathcode`\=\relax
1964   }
1965 }{}

```

- This code is to improve the output of alphabetic symbols in text of operator names (`\sin`, `\cos`, etc.). Just comment out the offending lines for now:

```

1966 \@@ifpackageloaded{amsopn}{
1967   \cs_set:Npn \newmcodes@ {
1968     \mathcode`\'39
1969     \mathcode`\*42
1970     \mathcode`\."613A%
1971     % \ifnum\mathcode`\-=45 \else
1972     %   \mathchardef\std@minus\mathcode`\-\relax
1973     % \fi
1974     \mathcode`\-45
1975     \mathcode`\ /47
1976     \mathcode`\:"603A\relax
1977   }
1978 }{}

```

Octothorpe is an odd one:

```
1979 \AtBeginDocument{
1980   \def\#{\mode_if_math:TF{\mathoctothorpe}{\char` \#}}
1981   \def\widehat{\hat}
1982   \def\widetilde{\tilde}
1983 }
```

`\digamma` I might end up just changing these in the table.

```
\Digamma 1984 \def\digamma{\updigamma}
1985 \def\Digamma{\upDigamma}
```

Overriding amsmath definitions:

```
1986 \AtBeginDocument{
1987   \def@cdots{\mathinner{\cdots}}
1988 }
```

Interaction with beamer:

```
1989 \@ifclassloaded{beamer}{
1990   \ifbeamer@suppressreplacements\else
1991     \PackageWarningNoLine{unicode-math}{
1992       Disabling~ beamer's~ math~ setup.^{^}
1993       Please~ load~ beamer~ with~ the~ [professionalfonts]~ class~ option
1994     }
1995     \beamer@suppressreplacementstrue
1996   \fi
1997 }{}
```

The end.

```
1998 \ExplSyntaxOff
```

File II

STIX table data extraction

The source for the \TeX names for the very large number of mathematical glyphs are provided via Barbara Beeton's table file for the STIX project (ams.org/STIX). A version is located at <http://www.ams.org/STIX/bnb/stix-tbl.asc> but check <http://www.ams.org/STIX/> for more up-to-date info.

This table is converted into a form suitable for reading by \XeTeX , and then hand-edited by the author; the result is `unicode-math-table.tex`.

A single file is produced containing all (more than 3298) symbols. Future optimisations might include generating various (possibly overlapping) subsets so not all definitions must be read just to redefine a small range of symbols. Performance for now seems to be acceptable without such measures.

```

1 #!/bin/sh
2
3 cat stix-tbl.txt |
4 awk '

```

If the USV isn't repeated (TODO: check this is valid!) and the entry isn't one of the weird ones in the big block at the end of the STIX table (TODO: check that out!)...

```

5 {if (usv != substr($0,2,5) && substr($0,2,1) != " ")
6     {usv = substr($0,2,5);
7       texname = substr($0,84,25);
8       class = substr($0,57,1);
9       description = tolower(substr($0,233,350));

```

If the USV has a macro name, which isn't `\text...`, and isn't a single character macro (e.g., `\#`, `\S`, ...), and has a class, and it isn't reserved (*i.e.*, doubled up with a previously assigned glyph):

```

10     if (texname ~ /[\\]/ &&
11         substr(texname,0,5) != "\\text" &&
12         substr(texname,0,4) != "\\ipa" &&
13         substr(texname,0,5) != "\\tone" &&
14         substr(texname,3,1) != " " &&
15         class != " " &&
16         description !~ /<reserved>/ )

```

Print the actual entry corresponding to the unicode character:

```

17     print "\\UnicodeMathSymbol{" \" \" \
18         usv "}{" \" \
19         texname "}{" \" \
20         class "}{" \" \
21         description "%";
22     } }' - |

```

Now replace the STIX class abbreviations with their \TeX macro names.

```

23 sed -e ' s/{N}/{\\mathord}/ ' ' \

```

A 'fence' defined by the STIX table is something like `\vert`; in \XeTeX this is just a `\mathord` that will grow with the magic of `\XeTeXmathchardef`.

```

24     -e ' s/{F}/{\\mathord}/ ' ' \
25     -e ' s/{A}/{\\mathalpha}/ ' ' \
26     -e ' s/{D}/{\\mathaccent}/ ' ' \
27     -e ' s/{P}/{\\mathpunct}/ ' ' \
28     -e ' s/{B}/{\\mathbin}/ ' ' \
29     -e ' s/{R}/{\\mathrel}/ ' ' \
30     -e ' s/{L}/{\\mathop}/ ' ' \
31     -e ' s/{O}/{\\mathopen}/ ' ' \
32     -e ' s/{C}/{\\mathclose}/ ' ' \

```

Fixing up a couple of things in the STIX table.

```

33     -e ' s/^\nstring^/ ' ' > unicode-math.tex

```

A Documenting maths support in the NFSS

A.1 Overview

In the following, $\langle NFSS\ decl. \rangle$ stands for something like $\{T1\}\{lmr\}\{m\}\{n\}$.

Maths symbol fonts Fonts for symbols: α , \leq , \rightarrow

$\backslash\text{DeclareSymbolFont}\{\langle name \rangle\}\langle NFSS\ decl. \rangle$

Declares a named maths font such as operators from which symbols are defined with $\backslash\text{DeclareMathSymbol}$.

Maths alphabet fonts Fonts for $ABC-xyz$, $\mathfrak{ABC}-\mathcal{XYZ}$, etc.

$\backslash\text{DeclareMathAlphabet}\{\langle cmd \rangle\}\langle NFSS\ decl. \rangle$

For commands such as $\backslash\text{mathbf}$, accessed through maths mode that are unaffected by the current text font, and which are used for alphabetic symbols in the ASCII range.

$\backslash\text{DeclareSymbolFontAlphabet}\{\langle cmd \rangle\}\{\langle name \rangle\}$

Alternative (and optimisation) for $\backslash\text{DeclareMathAlphabet}$ if a single font is being used for both alphabetic characters (as above) and symbols.

Maths ‘versions’ Different maths weights can be defined with the following, switched in text with the $\backslash\text{mathversion}\{\langle maths\ version \rangle\}$ command.

$\backslash\text{SetSymbolFont}\{\langle name \rangle\}\{\langle maths\ version \rangle\}\langle NFSS\ decl. \rangle$

$\backslash\text{SetMathAlphabet}\{\langle cmd \rangle\}\{\langle maths\ version \rangle\}\langle NFSS\ decl. \rangle$

Maths symbols Symbol definitions in maths for both characters (=) and macros ($\backslash\text{eqdef}$): $\backslash\text{DeclareMathSymbol}\{\langle symbol \rangle\}\{\langle type \rangle\}\{\langle named\ font \rangle\}\{\langle slot \rangle\}$ This is the macro that actually defines which font each symbol comes from and how they behave.

Delimiters and radicals use wrappers around $\text{T}_{\text{E}}\text{X}$ ’s $\backslash\text{delimiter}/\backslash\text{radical}$ primitives, which are re-designed in $\text{X}_{\text{E}}\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$. The syntax used in $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ ’s NFSS is therefore not so relevant here.

Delimiters A special class of maths symbol which enlarge themselves in certain contexts.

$\backslash\text{DeclareMathDelimiter}\{\langle symbol \rangle\}\{\langle type \rangle\}\{\langle sym.\ font \rangle\}\{\langle slot \rangle\}\{\langle sym.\ font \rangle\}\{\langle slot \rangle\}$

Radicals Similar to delimiters ($\backslash\text{DeclareMathRadical}$ takes the same syntax) but behave ‘weirdly’. $\backslash\text{sqrt}$ might very well be the only one.

In those cases, glyph slots in *two* symbol fonts are required; one for the small (‘regular’) case, the other for situations when the glyph is larger. This is not the case in $\text{X}_{\text{E}}\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$.

Accents are not included yet.

Summary For symbols, something like:

```
\def\DeclareMathSymbol#1#2#3#4{
  \global\mathchardef#1"\mathchar@type#2
  \expandafter\hexnumber@\csname sym#2\endcsname
  {\hexnumber@{\count\z@}\hexnumber@{\count\tw@}}}
```

For characters, something like:

```
\def\DeclareMathSymbol#1#2#3#4{
  \global\mathcode`#1"\mathchar@type#2
  \expandafter\hexnumber@\csname sym#2\endcsname
  {\hexnumber@{\count\z@}\hexnumber@{\count\tw@}}}
```

File III

X_YTeX math font dimensions

These are the extended `\fontdimen`s available for suitable fonts in X_YTeX. Note that LuaTeX takes an alternative route, and this package will eventually provide a wrapper interface to the two (I hope).

<code>\fontdimen</code>	Dimension name	Description
10	<code>SCRIPTPERCENTSCALEDOWN</code>	Percentage of scaling down for script level 1. Suggested value: 80%.
11	<code>SCRIPTSCRIPTPERCENTSCALEDOWN</code>	Percentage of scaling down for script level 2 (ScriptScript). Suggested value: 60%.
12	<code>DELIMITEDSUBFORMULAMINHEIGHT</code>	Minimum height required for a delimited expression to be treated as a subformula. Suggested value: normal line height \times 1.5.
13	<code>DISPLAYOPERATORMINHEIGHT</code>	Minimum height of n-ary operators (such as integral and summation) for formulas in display mode.
14	<code>MATHLEADING</code>	White space to be left between math formulas to ensure proper line spacing. For example, for applications that treat line gap as a part of line ascender, formulas with ink going above (<code>os2.sTypoAscender</code> + <code>os2.sTypoLineGap</code> – <code>MathLeading</code>) or with ink going below <code>os2.sTypoDescender</code> will result in increasing line height.

\fontdimen	Dimension name	Description
15	AxisHEIGHT	Axis height of the font.
16	ACCENTBASEHEIGHT	Maximum (ink) height of accent base that does not require raising the accents. Suggested: x-height of the font (os2.sxHeight) plus any possible overshots.
17	FLATTENEDACCENTBASE-HEIGHT	Maximum (ink) height of accent base that does not require flattening the accents. Suggested: cap height of the font (os2.sCapHeight).
18	SUBSCRIPTSHIFTDOWN	The standard shift down applied to subscript elements. Positive for moving in the downward direction. Suggested: os2.ySubscriptYOffset.
19	SUBSCRIPTTOPMAX	Maximum allowed height of the (ink) top of subscripts that does not require moving subscripts further down. Suggested: $\frac{1}{5}$ x-height.
20	SUBSCRIPTBASELINEDROPMIN	Minimum allowed drop of the baseline of subscripts relative to the (ink) bottom of the base. Checked for bases that are treated as a box or extended shape. Positive for subscript baseline dropped below the base bottom.
21	SUPERSCRIPSHIFTUP	Standard shift up applied to superscript elements. Suggested: os2.ySuperscriptYOffset.
22	SUPERSCRIPSHIFTUPCRAMPED	Standard shift of superscripts relative to the base, in cramped style.
23	SUPERSCRIPBOTTOMMIN	Minimum allowed height of the (ink) bottom of superscripts that does not require moving subscripts further up. Suggested: $\frac{1}{4}$ x-height.
24	SUPERSCRIPBASELINEDROP-MAX	Maximum allowed drop of the baseline of superscripts relative to the (ink) top of the base. Checked for bases that are treated as a box or extended shape. Positive for superscript baseline below the base top.
25	SUBSUPERSCRIPGAPMIN	Minimum gap between the superscript and subscript ink. Suggested: 4×default rule thickness.

\fontdimen	Dimension name	Description
26	SUPERSCRIPBTOTTOMMAX- WITHSUBSCRIPT	The maximum level to which the (ink) bottom of superscript can be pushed to increase the gap between superscript and subscript, before subscript starts being moved down. Suggested: $\frac{1}{5}$ x-height.
27	SPACEAFTERScript	Extra white space to be added after each subscript and superscript. Suggested: 0.5pt for a 12 pt font.
28	UPPERLIMITGAPMIN	Minimum gap between the (ink) bottom of the upper limit, and the (ink) top of the base operator.
29	UPPERLIMITBASELINERISEMIN	Minimum distance between baseline of upper limit and (ink) top of the base operator.
30	LOWERLIMITGAPMIN	Minimum gap between (ink) top of the lower limit, and (ink) bottom of the base operator.
31	LOWERLIMITBASELINE DROP- MIN	Minimum distance between baseline of the lower limit and (ink) bottom of the base operator.
32	STACKTOPSHIFTUP	Standard shift up applied to the top element of a stack.
33	STACKTOPDISPLAYSTYLESHIFT- UP	Standard shift up applied to the top element of a stack in display style.
34	STACKBOTTOMSHIFTDOWN	Standard shift down applied to the bottom element of a stack. Positive for moving in the downward direction.
35	STACKBOTTOMDISPLAYSTYLE- SHIFTDOWN	Standard shift down applied to the bottom element of a stack in display style. Positive for moving in the downward direction.
36	STACKGAPMIN	Minimum gap between (ink) bottom of the top element of a stack, and the (ink) top of the bottom element. Suggested: $3 \times$ default rule thickness.
37	STACKDISPLAYSTYLEGAPMIN	Minimum gap between (ink) bottom of the top element of a stack, and the (ink) top of the bottom element in display style. Suggested: $7 \times$ default rule thickness.
38	STRETCHSTACKTOPSHIFTUP	Standard shift up applied to the top element of the stretch stack.

\fontdimen	Dimension name	Description
39	STRETCHSTACKBOTTOMSHIFT- DOWN	Standard shift down applied to the bottom element of the stretch stack. Positive for moving in the downward direction.
40	STRETCHSTACKGAPABOVEMIN	Minimum gap between the ink of the stretched element, and the (ink) bottom of the element above. Suggested: UpperLimitGapMin
41	STRETCHSTACKGAPBELOWMIN	Minimum gap between the ink of the stretched element, and the (ink) top of the element below. Suggested: LowerLimitGapMin.
42	FRACTIONNUMERATORSHIFTUP	Standard shift up applied to the numerator.
43	FRACTIONNUMERATOR- DISPLAYSTYLESHIFTUP	Standard shift up applied to the numerator in display style. Suggested: StackTopDisplayStyleShiftUp.
44	FRACTIONDENOMINATORSHIFT- DOWN	Standard shift down applied to the denominator. Positive for moving in the downward direction.
45	FRACTIONDENOMINATOR- DISPLAYSTYLESHIFTDOWN	Standard shift down applied to the denominator in display style. Positive for moving in the downward direction. Suggested: StackBottomDisplayStyleShiftDown.
46	FRACTIONNUMERATORGAP- MIN	Minimum tolerated gap between the (ink) bottom of the numerator and the ink of the fraction bar. Suggested: default rule thickness
47	FRACTIONNUMDISPLAYSTYLE- GAPMIN	Minimum tolerated gap between the (ink) bottom of the numerator and the ink of the fraction bar in display style. Suggested: 3×default rule thickness.
48	FRACTIONRULETHICKNESS	Thickness of the fraction bar. Suggested: default rule thickness.
49	FRACTIONDENOMINATORGAP- MIN	Minimum tolerated gap between the (ink) top of the denominator and the ink of the fraction bar. Suggested: default rule thickness

\fontdimen	Dimension name	Description
50	FRACTIONDENOMDISPLAY- STYLEGAPMIN	Minimum tolerated gap between the (ink) top of the denominator and the ink of the fraction bar in display style. Suggested: 3×default rule thickness.
51	SKEWEDFRACTION- HORIZONTALGAP	Horizontal distance between the top and bottom elements of a skewed fraction.
52	SKEWEDFRACTIONVERTICAL- GAP	Vertical distance between the ink of the top and bottom elements of a skewed fraction.
53	OVERBARVERTICALGAP	Distance between the overbar and the (ink) top of the base. Suggested: 3×default rule thickness.
54	OVERBARRULETHICKNESS	Thickness of overbar. Suggested: default rule thickness.
55	OVERBAREXTRAASCENDER	Extra white space reserved above the overbar. Suggested: default rule thickness.
56	UNDERBARVERTICALGAP	Distance between underbar and (ink) bottom of the base. Suggested: 3×default rule thickness.
57	UNDERBARRULETHICKNESS	Thickness of underbar. Suggested: default rule thickness.
58	UNDERBAREXTRADESCENDER	Extra white space reserved below the underbar. Always positive. Suggested: default rule thickness.
59	RADICALVERTICALGAP	Space between the (ink) top of the expression and the bar over it. Suggested: 1¼ default rule thickness.
60	RADICALDISPLAYSTYLE- VERTICALGAP	Space between the (ink) top of the expression and the bar over it. Suggested: default rule thickness + ¼ x-height.
61	RADICALRULETHICKNESS	Thickness of the radical rule. This is the thickness of the rule in designed or constructed radical signs. Suggested: default rule thickness.
62	RADICALEXTRAASCENDER	Extra white space reserved above the radical. Suggested: RadicalRuleThickness.
63	RADICALKERNBEFOREDEGREE	Extra horizontal kern before the degree of a radical, if such is present. Suggested: 5/18 of em.

\fontdimen	Dimension name	Description
64	RADICALKERNAFTERDEGREE	Negative kern after the degree of a radical, if such is present. Suggested: –10/18 of em.
65	RADICALDEGREEBOTTOM- RAISEPERCENT	Height of the bottom of the radical degree, if such is present, in proportion to the ascender of the radical sign. Suggested: 60%.

Index

Numbers written in *italic* refer to the page where the corresponding entry is described; numbers underlined refer to the code line of the definition; numbers in *roman* refer to the code lines where the entry is used.

Symbols	
\ "	17
\ #	1725, 1980
\ '	1781, 1968
\ *	545, 1729, 1969
\ -	544, 1962, 1971, 1972, 1974
\	1970
\ /	594, 1975
\ :	547, 1976
\ ::	846
\ ::f	846
\ ::n	846
\ <	598
\ =	1963
\ >	599
\ @DeclareMathDelimiter	382
\ @DeclareMathSizes	347
\ @backslashchar	1213, 1233
\ @begindocumenthook	1961
\ @cclvi	407
\ @cdots	1987
\ @elt	680–684, 687, 691, 693
\ @empty	460, 769, 784, 805, 810
\ @ifclassloaded	1989
\ @ifnextchar	1786
\ @ifpackageloaded ...	1951, 1960, 1966
\ @ii	768, 769, 771, 773, 776, 781
\ @input	520, 1726
\ @marker	781, 800
\ @nil	416,
	575, 781, 794–797, 1705, 1710, 1723
\ @preamblecmds	391
\ @sqrt	1786
\ @tempa	151, 219, 252, 280, 298,
	316, 324, 331, 726, 772, 796, 798, 810
\ @tempb ..	151, 152, 219, 220, 252, 253,
	280, 281, 298, 299, 316, 317, 324,
	325, 331, 332, 726, 727, 799, 800, 805
\ @tempswafalse	770
\ @tempswatruetrue	774, 777, 802, 807, 812, 817
\ @um@bfliteraltrue	206, 239
\ @um@bfupGreekfalse	157, 221
\ @um@bfupGreektrue	170, 183, 196, 227, 233
\ @um@bfupLatinfalse	160, 223
\ @um@bfupLatintrue	173, 186, 199, 229, 235
\ @um@bfupgreekfalse	158, 171, 222, 228
\ @um@bfupgreektrue	184, 197, 234
\ @um@bfuplatinfalse	161, 224
\ @um@bfuplatintrue	174, 187, 200, 230, 236
\ @um@fontspec@featuretrue	461
\ @um@literaltrue	205
\ @um@ot@math@true	495
\ @um@upGreekfalse	153
\ @um@upGreektrue	166, 179, 192
\ @um@upLatinfalse	155, 168
\ @um@upLatintrue	181, 194
\ @um@upNablafalse	162, 284
\ @um@upNablatrue	175, 188, 201, 282
\ @um@upgreekfalse	154, 167
\ @um@upgreektrue	180, 193
\ @um@uplatinfalse	156, 169, 182
\ @um@uplatintrue	195
\ @um@uppartialfalse	
	159, 172, 225, 231, 302
\ @um@uppartialtrue ..	185, 198, 237, 300
\ @xDeclareMathDelimiter	383
\ @xxDeclareMathDelimiter	381
\ \	10–13, 17, 23–33, 1729, 1731
\ ^	33, 1704, 1714
\ 	1730
Numbers	
\ 0	1136
\ 9	1136
\ _	17–20, 23–32
A	
\ A	1133
\ a	1132

\addnolimits	686	\chi	925, 1886
\addtoversion	354	\cirfnint	682
\alloc@	407	\clist_map_break:	1204
\Alpha	874	\clist_map_inline:Nn	666
\alpha	899	\clist_map_inline:nn	
\alpha@elt	358	580, 661, 668, 734, 1201, 1207
\alpha@list	357	\clist_map_variable:nNn ..	825, 848, 854
\AtBeginDocument	872, 1979, 1986	\colon	1954
\awint	682	\copy	1794
B		\cs	1810, 1899
\B	1339	\cs_generate_variant:Nn	
\beamer@suppressreplacementstrue	1995	..	742, 1220, 1240–1243, 1800–1802
\begingroup	413, 690, 1703, 1721	\cs_gset:cpn	421, 431
\Beta	875	\cs_gset:Npn ..	434, 439, 1705, 1817, 1850
\beta	900, 1882	\cs_gset:Npx	444
\bgroup	1250	\cs_gset_eq:NN	1783, 1784
\bool_if:NF	546	\cs_if_exist:cF	1245
\bool_if:NTF		\cs_if_exist:cT	1228
	262, 503, 631, 649, 904, 923, 929,	\cs_new:Nn	528, 543, 574, 579, 584, 587,
	934, 1368, 1376, 1382, 1576, 1584,		593, 624, 627, 847, 853, 863, 866,
	1590, 1598, 1608, 1616, 1636, 1955		869, 1159, 1198, 1227, 1244, 1737,
\bool_if:nTF	1912		1743, 1759, 1763, 1891, 1901, 1931
\bool_new:N	10, 23, 249, 250, 323	\cs_new:Npn ..	1259, 1262, 1266, 1269,
\bool_set_false:N	163, 164,		1274, 1284, 1287, 1291, 1296,
	177, 190, 203, 208, 254, 318, 328, 731		1306, 1310, 1320, 1323, 1330,
\bool_set_true:N	176, 189,		1337, 1348, 1355, 1363, 1367,
	202, 207, 256, 258, 320, 326, 458, 728		1375, 1389, 1392, 1395, 1399,
\box	1796		1402, 1405, 1408, 1412, 1415,
C			1418, 1422, 1425, 1433, 1447,
\C	1312, 1357		1466, 1505, 1508, 1511, 1515,
\c_group_begin_token	1913		1520, 1531, 1534, 1537, 1541,
\c_group_end_token	1914		1547, 1558, 1561, 1565, 1568,
\c_peek_true_remove_next_tl ...	1903		1572, 1575, 1583, 1597, 1615,
\c_space_token	1915		1653, 1656, 1659, 1663, 1668,
\cdots	1987		1678, 1681, 1684, 1688, 1693, 1911
\cdp@elt	345	\cs_set:cpn	1246
\cdp@list	344	\cs_set:Nn	211, 242, 261,
\char	1980		287, 305, 453, 535, 538, 630, 660,
\char_make_active:N	1781		665, 824, 834, 837, 840, 843, 939,
\char_make_active:n	414, 1782, 1814, 1847		963, 987, 1015, 1018, 1022, 1026,
\char_make_escape:N	1730		1044, 1047, 1050, 1054, 1072,
\char_make_math_superscript:N ..	1714		1076, 1080, 1084, 1232, 1812, 1845
\char_make_other:N ..	1704, 1725, 1731	\cs_set:Npn	759, 763, 846, 1967
\chardef	407	\cs_set:Npx	470, 474
\Chi	896	\cs_set_eq:cN	1235

<code>\cs_set_eq:NN</code>	506–509, 513–516, 1182, 1191, 1779, 1819, 1852
<code>\cs_set_protected:cpn</code>	1249
<code>\cs_set_protected:Npn</code>	1954
<code>\cs_to_str:N</code> 416, 421, 431, 675, 1192, 1199	
<code>\csname</code>	410, 416, 419, 422, 454, 462, 589, 625, 628, 1253
D	
<code>\D</code>	1331
<code>\d</code>	1332
<code>\DeclareDocumentCommand</code>	147, 456
<code>\DeclareMathAccent</code>	375
<code>\DeclareMathAlphabet</code>	371
<code>\DeclareMathDelimiter</code>	380
<code>\DeclareMathRadical</code>	386
<code>\DeclareMathSizes</code>	346
<code>\DeclareMathSymbol</code>	377
<code>\DeclareMathVersion</code>	351, 464
<code>\DeclareRobustCommand</code>	1786
<code>\DeclareSymbolFont</code>	364, 518
<code>\DeclareSymbolFontAlphabet</code>	388
<code>\DeclareSymbolFontAlphabet@</code>	389
<code>\def</code>	24–146, 393, 396, 407, 409, 411, 463, 679, 687, 689, 691, 698, 700, 794, 796–799, 874–903, 906–922, 925–928, 931–933, 936, 937, 1716, 1722, 1787, 1945–1950, 1958, 1959, 1980–1982, 1984, 1985, 1987
<code>\define@choicekey</code>	150, 219, 251, 280, 298, 316, 324, 331, 726
<code>\define@cmdkey</code>	719–722
<code>\define@key</code>	702
<code>\define@mathalphabet</code>	352
<code>\define@mathgroup</code>	353
<code>\Delta</code>	877
<code>\delta</code>	902
<code>\Digamma</code>	<u>1984</u>
<code>\digamma</code>	<u>1984</u>
<code>\dimexpr</code>	394, 494, 1790
<code>\do</code>	391
<code>\dorestore@version</code>	361
<code>\dp</code>	1792
E	
<code>\E</code>	1340
<code>\e</code>	1333, 1350
<code>\edef</code>	795, 796
<code>\egroup</code>	1247, 1252
<code>\else</code> 82–85, 212–217, 243–247, 292, 310, 399, 402, 428, 432, 437, 442, 445, 496, 554, 567, 635, 643, 692, 705, 775, 780, 786, 804, 809, 814, 1035, 1063, 1093, 1429, 1438, 1442, 1456, 1461, 1485, 1494, 1971, 1990	
<code>\else:</code>	1238
<code>\encodingdefault</code>	519
<code>\endcsname</code>	410, 416, 419, 422, 454, 462, 589, 625, 628, 1253
<code>\endgroup</code>	417, 696, 1720, 1727
<code>\Epsilon</code>	878
<code>\epsilon</code>	903
<code>\errorcontextlines</code>	855
<code>\Eta</code>	880
<code>\eta</code>	907
<code>\etex_iffontchar:D</code>	1238
<code>\ExecuteOptionsX</code>	340
<code>\exp_after:wN</code>	748, 751, 1192, 1200
<code>\exp_args:Nnff</code>	846, 849, 857
<code>\exp_args:No</code>	662, 669
<code>\exp_args:NV</code>	1203
<code>\exp_not:N</code>	471, 475
<code>\exp_not:n</code>	1902, 1904
<code>\exp_not:V</code>	471, 736
<code>\expandafter</code>	410, 418, 423, 425, 429, 687, 771, 773, 776, 781, 795, 796, 800, 1252, 1253
<code>\ExplSyntaxOff</code>	1998
<code>\ExplSyntaxOn</code>	6, 1713
F	
<code>\F</code>	1341
<code>\f@size</code>	462
<code>\fi</code> 82–85, 209, 212–217, 240, 243–247, 259, 285, 296, 303, 314, 321, 329, 338, 404, 405, 426, 447–451, 502, 557, 572, 640, 648, 671, 694, 710, 729, 778, 779, 782, 788, 790, 791, 803, 808, 813, 818–822, 1042, 1070, 1100, 1431, 1444, 1445, 1463, 1464, 1501, 1503, 1973, 1996	
<code>\fi:</code>	1238
<code>\fint</code>	682
<code>\font</code>	493

\fontdimen	394, 494, 1789, 1795	\g_um_mathsfitt_alpha_clist ..	1124, 1171
\fontname	1216	\g_um_mathsfitt_usv	1151
G			
\g	1351	\g_um_mathsfup_alpha_clist ..	1123, 1170
\g@addto@macro	785, 787	\g_um_mathsfup_usv	1150
\g_um_literal_colon_bool		\g_um_mathtt_alpha_clist ...	1121, 1172
.....	323, 326, 328, 546, 1955	\g_um_mathtt_usv	1148
\g_um_math_alphabet_name_Greek_tl		\g_um_mathup_alpha_clist ...	1114, 1161
.....	1225	\g_um_mathup_Greek_usv	1135
\g_um_math_alphabet_name_greek_tl		\g_um_mathup_greek_usv	1134
.....	1224	\g_um_mathup_Latin_usv	1133
\g_um_math_alphabet_name_Latin_tl		\g_um_mathup_latin_usv	1132
.....	1223	\g_um_mathup_num_usv	1136
\g_um_math_alphabet_name_latin_tl		\g_um_primekern_muskip	1734, 1735, 1740
.....	1222	\g_um_sfliteral_bool	
\g_um_math_alphabet_name_num_tl	1226	207, 250, 258, 649,
\g_um_mathalph_seq	723, 753, 1102	1368, 1376, 1576, 1584, 1598, 1616
\g_um_mathbb_alpha_clist ...	1120, 1163	\g_um_slash_delimiter_usv	
\g_um_mathbb_usv	1147	333, 335, 337, 594–596
\g_um_mathbf_alpha_clist ...	1125, 1173	\g_um_subs_prop	1799, 1846
\g_um_mathbf_usv	1152	\g_um_supers_prop	1798, 1813
\g_um_mathbffrak_alpha_clist	1119, 1177	\g_um_texgreek_bool	23, 164, 177, 190,
\g_um_mathbffrak_usv	1146	203, 208, 318, 320, 904, 923, 929, 934
\g_um_mathbfit_alpha_clist ..	1127, 1175	\g_um_upsans_bool	
\g_um_mathbfit_usv	1154	163, 176, 189, 202, 249,
\g_um_mathbfscr_alpha_clist	1118, 1176	254, 256, 262, 1382, 1590, 1608, 1636
\g_um_mathbfscr_usv	1145	\Gamma	876
\g_um_mathbfssf_alpha_clist ..	1128, 1178	\gamma	901, 1883
\g_um_mathbfssf_usv	1155	\gdef	1710
\g_um_mathbfssfit_alpha_clist	1130, 1180	\ge	1949
\g_um_mathbfssfit_usv	1157	\geq	1949
\g_um_mathbfssfup_alpha_clist	1129, 1179	\get@cdp	370
\g_um_mathbfssfup_usv	1156	\glb@currsiz	457
\g_um_mathbfup_alpha_clist ..	1126, 1174	\global	415, 418, 435, 436,
\g_um_mathbfup_usv	1153	440, 441, 446, 1707, 1716, 1815, 1848
\g_um_mathfrak_alpha_clist ..	1117, 1168	\group@elt	366
\g_um_mathfrak_usv	1144	\group@list	365
\g_um_mathit_alpha_clist ...	1115, 1162	\group_align_safe_begin:	1905
\g_um_mathit_Greek_usv	1141	\group_align_safe_end:	1904
\g_um_mathit_greek_usv	1140	\group_begin:	1728, 1780, 1804
\g_um_mathit_Latin_usv	1139	\group_end:	1785, 1888
\g_um_mathit_latin_usv	1138	H	
\g_um_mathscr_alpha_clist ..	1116, 1167	\H	1313, 1342, 1358
\g_um_mathscr_usv	1143	\h	217, 1020, 1264, 1289
\g_um_mathsf_alpha_clist ...	1122, 1169	\hat	1981
\g_um_mathsf_usv	1149	\hbox	1788
		\ht	1791

I		K	
\I	1343, 1359	\Kappa	883
\i	1334	\kappa	910
\if@tempswa	783	\kern	1789, 1795
\if@um@bfliteral	16, 558, 641, 1426, 1434, 1449, 1468	L	
\if@um@bfupGreek	17, 243, 247, 1459	\L	1344
\if@um@bfupgreek	18, 244, 1056, 1086, 1487	\l_peek_false_tl	1904, 1917, 1943
\if@um@bfupLatin	19, 245	\l_peek_token	1913–1915, 1934, 1938
\if@um@bfuplatin	20, 246, 1440	\l_peek_true_aux_tl	1902
\if@um@fontspec@feature	8, 703	\l_peek_true_tl	1903, 1941
\if@um@literal	11, 549, 633	\l_um_char_range_seq	459, 725, 732, 738, 768
\if@um@ot@math@	9	\l_um_inc_num	856, 858, 859
\if@um@upGreek	12, 84, 212, 213	\l_um_incr_num	826, 828, 830
\if@um@upgreek	13, 85, 214, 1028	\l_um_init_bool	10, 458, 503, 631, 728, 731
\if@um@upLatin	14, 82, 215	\l_um_input_num	825, 828, 848, 850, 854, 858
\if@um@uplatin	15, 83, 216, 217	\l_um_mathalph_seq	724, 733, 736, 1160, 1183
\if@um@upNabla	21, 288	\l_um_primecount_num	1736, 1760, 1764, 1774
\if@um@uppartial	22, 306	\l_um_script_features_tl	465, 482
\ifbeamer@suppressreplacements	1990	\l_um_script_font_tl	467, 481
\ifcase	152, 220, 253, 281, 299, 317, 325, 332, 727	\l_um_ss_chain_tl	1818, 1851, 1895, 1940
\ifdim	494	\l_um_sscript_features_tl	466, 486
\ifin@	424, 430	\l_um_sscript_font_tl	468, 485
\ifnum	667, 801, 806, 811, 815, 816, 1971	\l_um_tmpa_tl	736, 744, 747, 748, 750, 751, 753, 760, 764, 1184, 1192, 1194, 1199, 1200, 1209, 1213, 1820, 1853, 1933, 1937
\ifx	397, 400, 412, 433, 438, 443, 692, 769, 772, 773, 776, 784, 800, 805, 810	\l_um_tmpe_tl	736, 745, 765, 1185, 1190, 1192, 1194, 1200, 1202, 1203, 1208, 1939, 1940
\iiint	680	\Lambda	884
\iint	680	\lambda	911
\iint	680	\lccode	1729
\in@	423, 429	\le	1948
\init@restore@version	360	\left	699
\int	680	\left@primitive	699, 700
\intBar	682	\leq	1948
\intbar	682	\let	408, 457, 460, 699, 1707
\intcap	684	\lowercase	1706, 1711
\intclockwise	681	\lowint	684
\intcup	684	J	
\intlarhk	683	\j	1335
\intx	683		
\Iota	882		
\iota	909		

M	
<code>\M</code>	1345
<code>\math@th</code>	1788
<code>\mathaccent</code>	443
<code>\mathalpha</code>	676, 829
<code>\mathbb</code>	1105, 1163, 1307, 1308, 1311–1318, 1321, 1324–1328
<code>\mathbbbit</code>	1331–1335
<code>\mathbf</code>	1107, 1173, 1423, 1427, 1428, 1430, 1435–1437, 1439, 1441, 1443, 1448, 1450–1455, 1457, 1458, 1460, 1462, 1467, 1469–1484, 1486, 1488–1493, 1495–1500, 1502
<code>\mathbffrac</code>	1108, 1177, 1559, 1562, 1563
<code>\mathbf fit</code> ...	1107, 1175, 1506, 1509, 1512, 1513, 1516–1518, 1521–1529
<code>\mathbfscr</code>	1108, 1176, 1566, 1569, 1570
<code>\mathbfsf</code> ...	1109, 1178, 1573, 1577, 1578, 1580, 1585–1587, 1589, 1591, 1593, 1599–1604, 1606, 1607, 1609, 1611, 1617–1632, 1634, 1635, 1637–1642, 1644–1649
<code>\mathbfsf fit</code>	1109, 1180, 1679, 1682, 1685, 1686, 1689–1691, 1694–1701
<code>\mathbfsfup</code>	1109, 1179, 1654, 1657, 1660, 1661, 1664–1666, 1669–1676
<code>\mathbfup</code> ...	1107, 1174, 1532, 1535, 1538, 1539, 1542–1545, 1548–1556
<code>\mathbin</code>	544, 545
<code>\mathcal</code>	1958
<code>\mathchar@type</code>	387, 420, 434, 436, 439, 441, 444, 446, 454, 588
<code>\mathchardef</code>	1962, 1963, 1972
<code>\mathclose</code>	438
<code>\mathcode</code>	415, 1962, 1963, 1968–1972, 1974–1976
<code>\mathfrac</code>	1105, 1168, 1356–1361, 1364, 1365
<code>\mathgroup</code>	407
<code>\mathinner</code>	1987
<code>\mathit</code>	1104, 1162, 1285, 1288, 1289, 1292–1294, 1297–1304
<code>\mathoctothorpe</code>	1980
<code>\mathop</code>	412
<code>\mathopen</code>	433, 700
<code>\mathord</code>	550–553, 555, 556, 559–566, 568–571, 585
<code>\mathpunct</code>	1955
<code>\mathrel</code>	547
<code>\mathrm</code>	1959
<code>\mathscr</code>	1105, 1167, 1338–1346, 1349–1353, 1958
<code>\mathsf</code>	1106, 1169, 1369, 1370, 1372, 1377–1379, 1381, 1383, 1385, 1390
<code>\mathsf fit</code>	1106, 1171, 1403, 1406, 1409, 1410
<code>\mathsfup</code>	1106, 1170, 1393, 1396, 1397, 1400
<code>\mathtt</code>	1105, 1172, 1413, 1416, 1419, 1420
<code>\mathup</code>	1104, 1161, 1260, 1263, 1264, 1267, 1270–1272, 1275–1282, 1959
<code>\mddefault</code>	519
<code>\meaning</code>	1813, 1846, 1934, 1938
<code>\mitAlpha</code>	874
<code>\mitalpha</code>	899
<code>\mitBeta</code>	875
<code>\mitbeta</code>	900
<code>\mitChi</code>	896
<code>\mitchi</code>	925
<code>\mitDelta</code>	877
<code>\mitdelta</code>	902
<code>\mitEpsilon</code>	878
<code>\mitepsilon</code>	904, 929
<code>\mitEta</code>	880
<code>\miteta</code>	907
<code>\mitGamma</code>	876
<code>\mitgamma</code>	901
<code>\mitIota</code>	882
<code>\mitiota</code>	909
<code>\mitKappa</code>	883
<code>\mitkappa</code>	910
<code>\mitLambda</code>	884
<code>\mitlambda</code>	911
<code>\mitMu</code>	885
<code>\mitmu</code>	912
<code>\mitNu</code>	886
<code>\mitnu</code>	913
<code>\mitOmega</code>	898
<code>\mitomega</code>	927
<code>\mitOmicron</code>	888
<code>\mitomicron</code>	915
<code>\mitPhi</code>	895
<code>\mitphi</code>	923, 934

<code>\mitPi</code>	889	<code>\newmathalphabet@@</code>	349
<code>\mitpi</code>	916	<code>\newmathalphabet@@@</code>	350
<code>\mitPsi</code>	897	<code>\newmcodes@</code>	1967
<code>\mitpsi</code>	926	<code>\noexpand</code>	693
<code>\mitRho</code>	890	<code>\nolimits</code>	425
<code>\mitrho</code>	917	<code>\non@alpherr</code>	1253
<code>\mitSigma</code>	892	<code>\npolint</code>	683
<code>\mitsigma</code>	919	<code>\Nu</code>	886
<code>\mitTau</code>	893	<code>\nu</code>	913
<code>\mittau</code>	920	<code>\num_incr:N</code>	1764
<code>\mitTheta</code>	881	<code>\num_new:N</code>	1736
<code>\mittheta</code>	908	<code>\num_zero:N</code>	1760
<code>\mitUpsilon</code>	894	<code>\number</code>	217, 850, 858, 859
<code>\mitupsilon</code>	921	<code>\numexpr</code>	806, 811, 815, 816, 828, 830, 850, 858, 859
<code>\mitvarepsilon</code>	904, 929		
<code>\mitvarkappa</code>	932		
<code>\mitvarphi</code>	923, 934		
<code>\mitvarpi</code>	937		
<code>\mitvarrho</code>	936		
<code>\mitvarsigma</code>	918		
<code>\mitvarTheta</code>	891		
<code>\mitvartheta</code>	931		
<code>\mitXi</code>	887		
<code>\mitxi</code>	914		
<code>\mitZeta</code>	879		
<code>\mitzeta</code>	906		
<code>\mode_if_math:F</code>	1251		
<code>\mode_if_math:TF</code>	1980		
<code>\mskip</code>	1740		
<code>\Mu</code>	885		
<code>\mu</code>	912		
<code>\muskip_gset:Nn</code>	1735		
<code>\muskip_new:N</code>	1734		
N			
<code>\N</code>	1314		
<code>\ne</code>	1950		
<code>\neq</code>	1950		
<code>\new@mathalphabet</code>	372		
<code>\new@mathgroup</code>	343, 407, 408		
<code>\new@mathversion</code>	363		
<code>\new@symbolfont</code>	367		
<code>\newcommand</code>	674, 686, 701, 767, 873		
<code>\newcounter</code>	7		
<code>\newfam</code>	408		
<code>\newif</code>	8, 9, 11–22		
<code>\newmathalphabet</code>	348		
		O	
		<code>\o</code>	1352
		<code>\oiint</code>	680
		<code>\oint</code>	680
		<code>\ointctrclockwise</code>	681
		<code>\Omega</code>	898
		<code>\omega</code>	927
		<code>\Omicron</code>	888
		<code>\omicron</code>	915
		<code>\operator@font</code>	1945
		<code>\or</code>	165, 178, 191, 204, 226, 232, 238, 255, 257, 283, 301, 319, 327, 334, 336
		<code>\overrightarrow</code>	1947
		P	
		<code>\P</code>	1315
		<code>\PackageError</code>	706
		<code>\PackageInfo</code>	505
		<code>\PackageWarning</code>	1188
		<code>\PackageWarningNoLine</code>	497, 1211, 1991
		<code>\peek_after:NN</code>	1906
		<code>\peek_meaning_remove:NTF</code>	1765, 1768, 1771
		<code>\Phi</code>	895
		<code>\phi</code>	922, 1885
		<code>\Pi</code>	889
		<code>\pi</code>	916
		<code>\pointint</code>	683
		<code>\prg_case_int:nnn</code>	1744
		<code>\prg_do_nothing:</code>	1235

\prg_new_conditional:Nnn .. 743, 1237	S
\prg_replicate:nn 1740	\sb 1852
\prg_return_false: 756, 1238	\scan_stop: .. 590, 591, 1238, 1815, 1848
\prg_return_true: 754, 1238	\scantokens 1707, 1816, 1849
\prg_stepwise_variable:nnnNn 826, 856	\scpolint 683
\prime 1779	\scriptscriptstyle 400
\primedouble 1747	\scriptstyle 397
\primequadruple 1753	\seq_clear:N 459, 732, 733, 1102
\primesingle 585, 1739, 1740, 1745	\seq_if_empty:NTF 1160
\primetriples 1750	\seq_if_in:NnTF 742
\process@table 362	\seq_if_in:NVTF 753
\ProcessOptionsX 341	\seq_map_inline:Nn 1183
\prop_get:cxN 1936	\seq_map_variable:Nnn 768
\prop_get:NnN 1801	\seq_new:N 723–725
\prop_gput:Nnn 1800	\seq_put_right:Nn 738, 1111
\prop_gput:Nxn 1813, 1846	\seq_put_right:Nx 736
\prop_if_in:cxTF 1932	\set@@mathdelimiter 385
\prop_if_in:NnTF 1802	\set@mathaccent 376
\prop_new:N 1798, 1799	\set@mathchar 378
\protect 709	\set@mathdelimiter 384
\ProvidesPackage 1	\set@mathsymbol 379
\Psi 897	\setbox 1788
\psi 926	\setkeys 148, 469, 471
Q	\SetMathAlphabet 373
\Q 1316	\SetMathAlphabet@ 374
\q_nil 748, 751, 759, 763	\setmathfont 456, 709
R	\SetSymbolFont 368
\R 1317, 1346, 1360	\SetSymbolFont@ 369
\r@@t 1787	\sf@size 480, 484
\raise 1790	\Sigma 892
\relax 152, 220, 253, 281, 299, 317, 325, 332, 394, 412, 415, 420, 431, 433–436, 438–441, 443, 444, 446, 454, 457, 493, 494, 667, 727, 773, 776, 800, 801, 806, 811, 815, 816, 828, 830, 850, 858, 859, 1793, 1962, 1963, 1972, 1976	\sigma 919
\removenolimits 689	\sp 1819
\RequirePackage 3–5	\space 1211, 1253
\restore@mathversion 359	\sqint 683
\Rho 890	\sqrt 698, 1786
\rho 917, 1884	\sqrtsign 1786, 1788
\rightarrow 1946	\std@equal 1963
\rootbox 1794	\std@minus 1962, 1972
\rppolint 682	\stepcounter 511
	\string 419, 422, 795, 796
	\strip@pt 394
	\sumint 681
	T
	\Tau 893
	\tau 920
	\tf@size 479, 480

\backslash Theta	881	\backslash um@scaled@apply	396, 1789, 1795
\backslash theta	908	\backslash um@scanactivedef	416, 1703
\backslash theum@fam	512	\backslash um@scancharlet	1703, 1723
\backslash thinmuskip	1735	\backslash um@set@mathsymbol	410, 411
\backslash tilde	1982	\backslash um@usv@bbLatin	34, 1311
\backslash tl_if_empty:Nf	1187	\backslash um@usv@bblatin	35, 1307
\backslash tl_if_empty:NT	1190	\backslash um@usv@bbnum	33, 1321
\backslash tl_if_in:NnT	747, 750	\backslash um@usv@bfDigamma	96, 1448, 1545
\backslash tl_map_inline:nn	342, 1103	\backslash um@usv@bfdigamma	103, 1467, 1556
\backslash tl_new:Nn	1114–1130, 1132–1136, 1138–1141, 1143–1157	\backslash um@usv@bffrakLatin	67, 1559
\backslash tl_put_right:cn	675	\backslash um@usv@bffraklatin	68, 1562
\backslash tl_put_right:Nv	1940	\backslash um@usv@bfGreek	57
\backslash tl_remove_in:Nn	391, 1961	\backslash um@usv@bfgreek	58
\backslash tl_rescan:nn	1712	\backslash um@usv@bfitGreek	65, 243, 970, 1051, 1451, 1516
\backslash tl_set:Nn	263–269, 271–277, 289–291, 293–295, 307–309, 311–313, 333, 335, 337, 465–468, 504, 744–746, 760, 761, 764, 765, 1222–1226, 1818, 1820, 1851, 1853	\backslash um@usv@bfitgreek	66, 244, 971, 1055, 1470, 1522
\backslash tl_set:No	1184–1186	\backslash um@usv@bfith	113, 1437, 1443
\backslash tl_set:Nv	1192	\backslash um@usv@bfitLatin	63, 245, 966, 1045, 1428, 1509, 1521
\backslash tl_set:Nx	212–217, 243–247, 512, 1199, 1200, 1902, 1904	\backslash um@usv@bfitlatin	64, 246, 967, 1048, 1436, 1512
\backslash tl_set_eq:NN	1903	\backslash um@usv@bfitNabla	138, 294, 560, 568, 1455, 1518
\backslash tl_use:c	1214	\backslash um@usv@bfitnum	54
\backslash to	1946	\backslash um@usv@bfitpartial	144, 312, 564, 570, 1478, 1523
\backslash token_if_eq_catcode_p:NN	1913, 1914	\backslash um@usv@bfitvarepsilon	115, 980, 1057, 1064, 1479, 1495, 1524
\backslash token_if_eq_meaning_p:NN	1915	\backslash um@usv@bfitvarkappa	117, 982, 1059, 1066, 1481, 1497, 1526
U			
\backslash um@addto@mathmap	662, 669, 674	\backslash um@usv@bfitvarphi	118, 983, 1060, 1067, 1482, 1498, 1527
\backslash um@backslash	772, 795	\backslash um@usv@bfitvarpi	120, 985, 1062, 1069, 1484, 1500, 1529
\backslash um@char@num@range	460, 666, 784, 785, 787	\backslash um@usv@bfitvarrho	119, 984, 1061, 1068, 1483, 1499, 1528
\backslash um@firstchar	771, 796	\backslash um@usv@bfitvarTheta	114, 247, 973, 1052, 1453, 1462, 1517
\backslash um@firsttof	794–796	\backslash um@usv@bfitvartheta	116, 981, 1058, 1065, 1480, 1496, 1525
\backslash um@font	398, 401, 493, 494, 1216, 1238, 1789, 1791, 1792, 1795	\backslash um@usv@bfLatin	55
\backslash um@fontdimen@percent	393, 398, 401, 1791, 1792	\backslash um@usv@bflatin	56
\backslash um@mathsymbol	409, 536	\backslash um@usv@bfNabla	137, 290, 559, 568, 1454, 1544
\backslash um@mversion	463, 464	\backslash um@usv@bfnum	52, 1423, 1506, 1532
\backslash um@nolimits	423, 679, 687, 695		
\backslash um@parse@range	781, 797		
\backslash um@parse@term	539, 575, 767		
\backslash um@radicals	429, 698		
\backslash um@resolve@greek	872		

$\backslash\mathrm{um@usv@bfpartial}$	$\backslash\mathrm{um@usv@bfsfupnum}$
..... 143, 308, 563, 570, 1471, 1549	72
$\backslash\mathrm{um@usv@bfscrLatin}$	$\backslash\mathrm{um@usv@bfsfupvarepsilon}$
69, 1566 122, 1002, 1087, 1094
$\backslash\mathrm{um@usv@bfscrLatin}$	$\backslash\mathrm{um@usv@bfsfupvarkappa}$
70, 1569 124, 1004, 1089, 1096
$\backslash\mathrm{um@usv@bfsfGreek}$	$\backslash\mathrm{um@usv@bfsfupvarphi}$
84 125, 1005, 1090, 1097
$\backslash\mathrm{um@usv@bfsfgreek}$	$\backslash\mathrm{um@usv@bfsfupvarpi}$ 127, 1007, 1092, 1099
85	$\backslash\mathrm{um@usv@bfsfupvarrho}$
$\backslash\mathrm{um@usv@bfsfitGreek}$ 126, 1006, 1091, 1098
.. 80, 84, 275, 998, 1081, 1600, 1689	$\backslash\mathrm{um@usv@bfsfupvarTheta}$
$\backslash\mathrm{um@usv@bfsfitgreek}$ 121, 269, 1000, 1082, 1601
.. 81, 85, 276, 999, 1085, 1618, 1694	$\backslash\mathrm{um@usv@bfsfupvarthetaeta}$
$\backslash\mathrm{um@usv@bfsfitLatin}$ 123, 1003, 1088, 1095
.. 78, 82, 273, 994, 1074, 1578, 1682	$\backslash\mathrm{um@usv@bfupGreek}$
$\backslash\mathrm{um@usv@bfsfitlatin}$ 61, 243, 968, 1051, 1450, 1542
.. 79, 83, 274, 995, 1078, 1586, 1685	$\backslash\mathrm{um@usv@bfupgreek}$
$\backslash\mathrm{um@usv@bfsfitNabla}$ 62, 244, 969, 1055, 1469, 1548
..... 140, 295, 562, 569, 1604, 1691	$\backslash\mathrm{um@usv@bfuph}$
$\backslash\mathrm{um@usv@bfsfitnum}$	112, 1441
73	$\backslash\mathrm{um@usv@bfupLatin}$
$\backslash\mathrm{um@usv@bfsfitpartial}$ 59, 245, 964, 1045, 1427, 1535
..... 146, 313, 566, 571, 1626, 1695	$\backslash\mathrm{um@usv@bfuplatin}$
$\backslash\mathrm{um@usv@bfsfitvarepsilon}$ 60, 246, 965, 1048, 1435, 1538
..... 129, 1008, 1087, 1094	$\backslash\mathrm{um@usv@bfupnum}$
$\backslash\mathrm{um@usv@bfsfitvarkappa}$	53
..... 131, 1010, 1089, 1096	$\backslash\mathrm{um@usv@bfvarepsilon}$
$\backslash\mathrm{um@usv@bfsfitvarphi}$	97, 974, 1057, 1064, 1472, 1488, 1550
..... 132, 1011, 1090, 1097	$\backslash\mathrm{um@usv@bfvarkappa}$
$\backslash\mathrm{um@usv@bfsfitvarpi}$ 134, 1013, 1092, 1099	99, 976, 1059, 1066, 1474, 1490, 1552
$\backslash\mathrm{um@usv@bfsfitvarrho}$	$\backslash\mathrm{um@usv@bfvarphi}$
..... 133, 1012, 1091, 1098	100, 977, 1060, 1067, 1475, 1491, 1553
$\backslash\mathrm{um@usv@bfsfitvarTheta}$	$\backslash\mathrm{um@usv@bfvarpi}$
..... 128, 277, 1001, 1082, 1602	102, 979, 1062, 1069, 1477, 1493, 1555
$\backslash\mathrm{um@usv@bfsfitvarthetaeta}$	$\backslash\mathrm{um@usv@bfvarrho}$
..... 130, 1009, 1088, 1095	101, 978, 1061, 1068, 1476, 1492, 1554
$\backslash\mathrm{um@usv@bfsfLatin}$	$\backslash\mathrm{um@usv@bfvarTheta}$
82	95, 247, 972, 1052, 1452, 1460, 1543
$\backslash\mathrm{um@usv@bfsflatIn}$	$\backslash\mathrm{um@usv@bfvarthetaeta}$
83	98, 975, 1058, 1065, 1473, 1489, 1551
$\backslash\mathrm{um@usv@bfsfNabla}$ 139, 291, 561, 569, 1603	$\backslash\mathrm{um@usv@Digamma}$
$\backslash\mathrm{um@usv@bfsfnum}$... 71, 1573, 1654, 1679	87, 1448, 1545
$\backslash\mathrm{um@usv@bfsfpartial}$	$\backslash\mathrm{um@usv@digamma}$
..... 145, 309, 565, 571, 1619	94, 1467, 1556
$\backslash\mathrm{um@usv@bfsfupGreek}$	$\backslash\mathrm{um@usv@frakLatin}$
.. 76, 84, 267, 996, 1081, 1599, 1664	38, 1356
$\backslash\mathrm{um@usv@bfsfupgreek}$	$\backslash\mathrm{um@usv@fraklatin}$
.. 77, 85, 268, 997, 1085, 1617, 1669	39, 1364
$\backslash\mathrm{um@usv@bfsfupLatin}$	$\backslash\mathrm{um@usv@itGreek}$
.. 74, 82, 265, 992, 1074, 1577, 1657	31, 212,
$\backslash\mathrm{um@usv@bfsfuplatin}$	948, 1023, 1270, 1292, 1451, 1457,
.. 75, 83, 266, 993, 1078, 1585, 1660	1516, 1542, 1600, 1606, 1664, 1689

$\backslash\mathrm{um@usv@itgreek}$	32, 214, 1027, 1275, 1297, 1470, 1486, 1522, 1548, 1618, 1634, 1669, 1694
$\backslash\mathrm{um@usv@ith}$	104, 217, 943, 1020, 1138, 1264, 1289, 1308, 1353, 1365, 1379, 1383, 1385, 1397, 1410, 1420, 1437, 1441, 1443, 1513, 1539, 1563, 1570, 1587, 1591, 1593, 1661, 1686
$\backslash\mathrm{um@usv@itLatin}$	29, 215, 941, 1016, 1260, 1285, 1311, 1338, 1356, 1370, 1372, 1400, 1406, 1416, 1428, 1430, 1509, 1521, 1535, 1559, 1566, 1578, 1580, 1657, 1682
$\backslash\mathrm{um@usv@itlatin}$	30, 216, 942, 1019, 1263, 1288, 1307, 1349, 1364, 1378, 1381, 1396, 1409, 1419, 1436, 1439, 1512, 1538, 1562, 1569, 1586, 1589, 1660, 1685
$\backslash\mathrm{um@usv@itNabla}$	136, 293, 551, 555, 1271, 1294, 1455, 1458, 1518, 1544, 1604, 1607, 1666, 1691
$\backslash\mathrm{um@usv@itpartial}$	142, 311, 553, 556, 1276, 1298, 1478, 1502, 1523, 1549, 1626, 1635, 1670, 1695
$\backslash\mathrm{um@usv@itvarepsilon}$	106, 956, 1029, 1036, 1277, 1299, 1479, 1488, 1495, 1524, 1550, 1627, 1637, 1644, 1671, 1696
$\backslash\mathrm{um@usv@itvarkappa}$	108, 958, 1031, 1038, 1279, 1301, 1481, 1490, 1497, 1526, 1552, 1629, 1639, 1646, 1673, 1698
$\backslash\mathrm{um@usv@itvarphi}$	109, 959, 1032, 1039, 1280, 1302, 1482, 1491, 1498, 1527, 1553, 1630, 1640, 1647, 1674, 1699
$\backslash\mathrm{um@usv@itvarpi}$	111, 961, 1034, 1041, 1282, 1304, 1484, 1493, 1500, 1529, 1555, 1632, 1642, 1649, 1676, 1701
$\backslash\mathrm{um@usv@itvarrho}$	110, 960, 1033, 1040, 1281, 1303, 1483, 1492, 1499, 1528, 1554, 1631, 1641, 1648, 1675, 1700
$\backslash\mathrm{um@usv@itvarTheta}$	105, 213, 947, 1024, 1272, 1293, 1453, 1460, 1462, 1517, 1543, 1602, 1609, 1611, 1665
$\backslash\mathrm{um@usv@itvartheta}$	107, 957, 1030, 1037, 1278, 1300, 1480, 1489, 1496, 1525, 1551, 1628, 1638, 1645, 1672, 1697
$\backslash\mathrm{um@usv@Nabla}$	135, 289, 550, 555, 1271, 1294, 1454, 1458, 1518, 1544, 1603, 1607, 1666, 1691
$\backslash\mathrm{um@usv@partial}$	141, 307, 552, 556, 1276, 1298, 1471, 1502, 1523, 1549, 1619, 1635, 1670, 1695
$\backslash\mathrm{um@usv@scrLatin}$	36, 1338
$\backslash\mathrm{um@usv@scrlatin}$	37, 1349
$\backslash\mathrm{um@usv@sfitLatin}$	47, 271, 990, 1073, 1370, 1406
$\backslash\mathrm{um@usv@sfitlatin}$	48, 272, 991, 1077, 1378, 1409
$\backslash\mathrm{um@usv@sfitnum}$	42
$\backslash\mathrm{um@usv@sflatin}$	44, 263
$\backslash\mathrm{um@usv@sflatin}$	46, 264
$\backslash\mathrm{um@usv@sfnun}$	40, 1390, 1393, 1403
$\backslash\mathrm{um@usv@sfulatin}$	43, 988, 1073, 1369, 1400
$\backslash\mathrm{um@usv@sfulatin}$	45, 989, 1077, 1377, 1396
$\backslash\mathrm{um@usv@sfulnum}$	41
$\backslash\mathrm{um@usv@ttLatin}$	50, 1416
$\backslash\mathrm{um@usv@ttlLatin}$	51, 1419
$\backslash\mathrm{um@usv@ttnum}$	49, 1413
$\backslash\mathrm{um@usv@upGreek}$	27, 212, 945, 1023, 1270, 1292, 1450, 1457, 1516, 1542, 1599, 1606, 1664, 1689
$\backslash\mathrm{um@usv@upgreek}$	28, 214, 949, 1027, 1275, 1297, 1469, 1486, 1522, 1548, 1617, 1634, 1669, 1694
$\backslash\mathrm{um@usv@upLatin}$	25, 215, 940, 1016, 1260, 1285, 1311, 1338, 1356, 1369, 1372, 1400, 1406, 1416, 1427, 1430, 1509, 1521, 1535, 1559, 1566, 1577, 1580, 1657, 1682
$\backslash\mathrm{um@usv@uplatin}$	26, 216, 944, 1019, 1263, 1288, 1307, 1349, 1364, 1377, 1381, 1396, 1409,

1419, 1435, 1439, 1512, 1538, 1562, 1569, 1585, 1589, 1660, 1685 308, 312, 570, 1502
\um@usv@upnum 24, 632, 1267, 1321, 1390, 1393, 1403, 1413, 1423, 1506, 1532, 1573, 1654, 1679	\um_bfsf_Greek_up_or_it_usv 267, 275, 1081, 1606
\um@usv@varepsilon 88, 950, 1029, 1036, 1277, 1299, 1472, 1488, 1495, 1524, 1550, 1620, 1637, 1644, 1671, 1696	\um_bfsf_greek_up_or_it_usv 268, 276, 1085, 1634
\um@usv@varkappa 90, 952, 1031, 1038, 1279, 1301, 1474, 1490, 1497, 1526, 1552, 1622, 1639, 1646, 1673, 1698	\um_bfsf_Latin_up_or_it_usv 265, 273, 1074, 1580
\um@usv@varphi 91, 953, 1032, 1039, 1280, 1302, 1475, 1491, 1498, 1527, 1553, 1623, 1640, 1647, 1674, 1699	\um_bfsf_latin_up_or_it_usv 266, 274, 1078, 1589
\um@usv@varpi 93, 955, 1034, 1041, 1282, 1304, 1477, 1493, 1500, 1529, 1555, 1625, 1642, 1649, 1676, 1701	\um_bfsf_varTheta_up_or_it_usv 269, 277, 1082
\um@usv@varrho 92, 954, 1033, 1040, 1281, 1303, 1476, 1492, 1499, 1528, 1554, 1624, 1641, 1648, 1675, 1700	\um_bfsfNabla_up_or_it_usv 291, 295, 569, 1607
\um@usv@varTheta 86, 213, 946, 1024, 1272, 1293, 1452, 1460, 1462, 1517, 1543, 1601, 1609, 1611, 1665, 1690	\um_bfsfpartial_up_or_it_usv 309, 313, 571, 1635
\um@usv@vartheta 89, 951, 1030, 1037, 1278, 1300, 1473, 1489, 1496, 1525, 1551, 1621, 1638, 1645, 1672, 1697	\um_config_mathbb_Latin: 1310
\um@zf@feature 701, 713, 716	\um_config_mathbb_latin: 1306
\um_bf_Greek_up_or_it_usv 243, 1051, 1457	\um_config_mathbb_misc: 1323
\um_bf_greek_up_or_it_usv 244, 1055, 1486	\um_config_mathbb_num: 1320
\um_bf_Latin_up_or_it_usv 245, 1045, 1430	\um_config_mathbbbit_misc: 1330
\um_bf_latin_up_or_it_usv 246, 1048, 1439	\um_config_mathbbf_Greek: 1447
\um_bf_varTheta_up_or_it_usv 247, 1052	\um_config_mathbbf_greek: 1466
\um_bfNabla_up_or_it_usv 290, 294, 568, 1458	\um_config_mathbbf_Latin: 1425
\um_bfpartial_up_or_it_usv	\um_config_mathbbf_latin: 1433
	\um_config_mathbbf_num: 1422
	\um_config_mathbbffrak_Latin: ... 1558
	\um_config_mathbbffrak_latin: ... 1561
	\um_config_mathbbfit_Greek: 1515
	\um_config_mathbbfit_greek: 1520
	\um_config_mathbbfit_Latin: 1508
	\um_config_mathbbfit_latin: 1511
	\um_config_mathbbfit_num: 1505
	\um_config_mathbbfscr_Latin: ... 1565
	\um_config_mathbbfscr_latin: ... 1568
	\um_config_mathbbfsf_Greek: 1597
	\um_config_mathbbfsf_greek: 1615
	\um_config_mathbbfsf_Latin: 1575
	\um_config_mathbbfsf_latin: 1583
	\um_config_mathbbfsf_num: 1572
	\um_config_mathbbfsfit_Greek: ... 1688
	\um_config_mathbbfsfit_greek: ... 1693
	\um_config_mathbbfsfit_Latin: ... 1681
	\um_config_mathbbfsfit_latin: ... 1684
	\um_config_mathbbfsfit_num: 1678
	\um_config_mathbbfsfup_Greek: ... 1663
	\um_config_mathbbfsfup_greek: ... 1668

\um_config_mathbfup_Latin: ...	1656	\um_Latin_up_or_it_usv	215, 1016
\um_config_mathbfup_latin: ...	1659	\um_latin_up_or_it_usv	216, 1019
\um_config_mathbfup_num:	1653	\um_latinh_up_or_it_usv ...	217, 1020
\um_config_mathbfup_Greek:	1541	\um_make_mathactive:nnn	585, <u>587</u>
\um_config_mathbfup_greek:	1547	\um_map_char:nn	
\um_config_mathbfup_Latin:	1534	... 843, 943, 946, 947, 950–961,	
\um_config_mathbfup_latin:	1537	972–985, 1000–1013, 1020,	
\um_config_mathbfup_num:	1531	1024, 1029–1034, 1036–1041,	
\um_config_mathfrak_Latin:	1355	1052, 1057–1062, 1064–1069,	
\um_config_mathfrak_latin:	1363	1082, 1087–1092, 1094–1099	
\um_config_mathit_Greek:	1291	\um_map_char:nn□	<u>824</u>
\um_config_mathit_greek:	1296	\um_map_chars_greek:nn	837,
\um_config_mathit_Latin:	1284	945, 948, 949, 968–971, 996–999,	
\um_config_mathit_latin:	1287	1023, 1027, 1051, 1055, 1081, 1085	
\um_config_mathscr_Latin:	1337	\um_map_chars_latin:nn	
\um_config_mathscr_latin:	1348 834, 940–942, 944,	
\um_config_mathsf_Latin:	1367	964–967, 988–995, 1016, 1019,	
\um_config_mathsf_latin:	1375	1045, 1048, 1073, 1074, 1077, 1078	
\um_config_mathsf_num:	1389	\um_map_chars_numbers:nn ...	632, 840
\um_config_mathsf_Latin:	1405	\um_map_chars_range:nnn	
\um_config_mathsf_latin:	1408 824, 835, 838, 841, 844	
\um_config_mathsf_num:	1402	\um_mathmap:Nnn 507, 514, 849, 857, 1182	
\um_config_mathsfup_Latin:	1399	\um_mathmap_noparse:Nnn 507, <u>660</u> , 1182	
\um_config_mathsfup_latin:	1395	\um_mathmap_parse:Nnn	514, <u>665</u>
\um_config_mathsfup_num:	1392	\um_maybe_init_alphabet:n	
\um_config_mathtt_Latin:	1415 509, 516, 1165, 1191, 1203	
\um_config_mathtt_latin:	1418	\um_Nabla_up_or_it_usv ..	289, 293, 555
\um_config_mathtt_num:	1412	\um_nprimes:n 1737, 1747, 1750, 1753, 1756	
\um_config_mathup_Greek:	1269	\um_nprimes_select:n	1743, 1774
\um_config_mathup_greek:	1274	\um_partial_up_or_it_usv 307, 311, 556	
\um_config_mathup_Latin:	1259	\um_peek_execute_branches_ss: ...	
\um_config_mathup_latin:	1262 1906, 1911	
\um_config_mathup_num:	1266	\um_peek_execute_branches_ss_aux:	
\um_glyph_if_exist:cT	1202 1918, 1931	
\um_glyph_if_exist:cTF	1208	\um_prepare_alph:n	1234, <u>1244</u>
\um_glyph_if_exist:n	1237	\um_process_symbol_noparse:nnnn .	
\um_glyph_if_exist:nF	1243 506, <u>535</u>	
\um_glyph_if_exist:nT	1242	\um_process_symbol_parse:nnnn 513, <u>535</u>	
\um_glyph_if_exist:nTF		\um_remap_symbol:nnn	
..... <u>1237</u> , 1747, 1750, 1753	 508, 515, 544, 545, 547,	
\um_glyph_if_exist_p:n	1240	550–553, 555, 556, 559–566, 568–571	
\um_Greek_up_or_it_usv	212, 1023	\um_remap_symbol_noparse:nnn 508, <u>543</u>	
\um_greek_up_or_it_usv	214, 1027	\um_remap_symbol_parse:nnn	
\um_if_mathalph_decl:n	743 515, <u>543</u> , 574	
\um_if_mathalph_decl:nTF	735	\um_remap_symbols:	522, <u>543</u>
\um_init_alphabet:n ..	509, 1191, 1232	\um_scan_sscript: 1821, 1854, 1891, 1893	

\um_scan_sscript:TF 1892, 1901	\um_set_mathalphabet_numbers:Nnn 863, 1267,
\um_scanprime:		1321, 1390, 1393, 1403, 1413,
.....	1759, 1768, 1779, 1783, 1784		1423, 1506, 1532, 1573, 1654, 1679
\um_scanprime_collect:	\um_set_mathcode:nnnn	<u>453</u> , 581, 676, 827
.....	1761, 1763, 1766, 1769, 1772	\um_setup_active_subscript:nn	...
\um_set_delcode:n	... 597, 600–622, 627	1845, 1859–1886
\um_set_delcode:nn	594–596, 598, 599, 624	\um_setup_active_superscript:nn	...
\um_set_mathalph_range:Nnn <u>853</u>	1812, 1826–1842
\um_set_mathalph_range:nNnn	\um_setup_all_shapes: 521, 528
.....	853, 864, 867, 870	\um_setup_alphabets: 526, <u>1113</u>
\um_set_mathalphabet_char:Nnn	<u>846</u> ,	\um_setup_alphanum: 525, <u>630</u>
	1264, 1271, 1272, 1276–1282,	\um_setup_bf_Greek: 646, <u>1050</u>
	1289, 1293, 1294, 1298–1304,	\um_setup_bf_greek: 647, <u>1054</u>
	1308, 1312–1318, 1324–1328,	\um_setup_bf_Latin: 644, <u>1044</u>
	1331–1335, 1339–1346, 1350–1353,	\um_setup_bf_latin: 645, <u>1047</u>
	1357–1361, 1365, 1379, 1383,	\um_setup_bf_literals: 642, <u>963</u>
	1385, 1397, 1410, 1420,	\um_setup_bfshapes: <u>242</u> , 533
	1437, 1441, 1443, 1448,	\um_setup_delcodes: 524, <u>593</u> , <u>624</u>
	1452–1455, 1458, 1460, 1462,	\um_setup_Greek: 638, <u>1022</u>
	1467, 1471–1484, 1488–1493,	\um_setup_greek: 639, <u>1026</u>
	1495–1500, 1502, 1513, 1517,	\um_setup_Latin: 636, <u>1015</u>
	1518, 1523–1529, 1539,	\um_setup_latin: 637, <u>1018</u>
	1543–1545, 1549–1556, 1563,	\um_setup_literals: 634, <u>939</u>
	1570, 1587, 1591, 1593,	\um_setup_math_alphabet:Nn <u>1198</u>
	1601–1604, 1607, 1609,	\um_setup_math_alphabet:NV
	1611, 1619–1632, 1635,	1161–1163, 1167–1180
	1637–1642, 1644–1649, 1661,	\um_setup_math_alphabet:WV 1194
	1665, 1666, 1670–1676,	\um_setup_math_mapping:n
	1686, 1690, 1691, 1695–1701	1164, 1166, 1227
\um_set_mathalphabet_greek:Nnn	..	\um_setup_mathactives: 523, <u>584</u>
.....	869, 1270, 1275, 1292,	\um_setup_mathup: 1945
	1297, 1450, 1451, 1457, 1469,	\um_setup_nabla: 287, 529
	1470, 1486, 1516, 1522, 1542,	\um_setup_partial: 305, 530
	1548, 1599, 1600, 1606, 1617,	\um_setup_sf_Greek: 654, <u>1080</u>
	1618, 1634, 1664, 1669, 1689, 1694	\um_setup_sf_greek: 655, <u>1084</u>
\um_set_mathalphabet_latin:Nnn	..	\um_setup_sf_Latin: 652, <u>1072</u>
.....	866, 1260,	\um_setup_sf_latin: 653, <u>1076</u>
	1263, 1285, 1288, 1307, 1311,	\um_setup_sf_literals: 650, <u>987</u>
	1338, 1349, 1356, 1364, 1369,	\um_setup_sfshapes: <u>261</u> , 532
	1370, 1372, 1377, 1378, 1381,	\um_setup_shapes: <u>211</u> , 531
	1396, 1400, 1406, 1409, 1416,	\um_sf_Latin_up_or_it_usv
	1419, 1427, 1428, 1430, 1435,	263, 271, 1073, 1372
	1436, 1439, 1509, 1512, 1521,	\um_sf_latin_up_or_it_usv
	1535, 1538, 1559, 1562, 1566,	264, 272, 1077, 1381
	1569, 1577, 1578, 1580, 1585,	\um_split_arrow:w 748, 759
	1586, 1589, 1657, 1660, 1682, 1685		

<code>\um_split_slash:w</code>	751, 763	<code>\vartheta</code>	931
<code>\um_sub_or_super:n</code> ..	1819, 1852, 1895	<code>\vec</code>	1947
<code>\um_symfont_tl</code>	504, 512, 518, 536, 581, 589, 625, 628, 662, 669, 829	<code>\version@elt</code>	356
<code>\um_tmp:</code>	470, 473, 474, 492	<code>\version@list</code>	355
<code>\um_varTheta_up_or_it_usv</code> .	213, 1024	W	
<code>\UnicodeMathSymbol</code>	506, 513, 1722	<code>\widehat</code>	1981
<code>\unimathsetup</code>	147	<code>\widetilde</code>	1982
<code>\unless</code>	769	<code>\wlog</code>	1233
<code>\updefault</code>	519	X	
<code>\upDigamma</code>	1985	<code>\xdef</code>	695
<code>\updigamma</code>	1984	<code>\XeTeXdelcode</code>	435, 440, 625, 628
<code>\upint</code>	684	<code>\XeTeXdelimiter</code>	434, 439
<code>\Upsilon</code>	894	<code>\XeTeXmathaccent</code>	444
<code>\upsilon</code>	921	<code>\XeTeXmathchardef</code>	418, 588
<code>\use:c</code>	1209, 1229, 1247, 1255	<code>\XeTeXmathcode</code>	436, 441, 446, 454
<code>\use_i:nnn</code>	1184	<code>\XeTeXmathcodenum</code>	591, 1815, 1848
<code>\use_ii:nnn</code>	1185	<code>\XeTeXradical</code>	431
<code>\use_iii:nnn</code>	1186	<code>\Xi</code>	887
<code>\use_none:n</code>	516	<code>\xi</code>	914
<code>\use_none:nnnn</code>	1200	<code>\XKV@rm</code>	471, 489
V		Z	
<code>\varepsilon</code>	928	<code>\Z</code>	1133, 1318, 1361
<code>\varkappa</code>	932	<code>\z</code>	1132
<code>\varointclockwise</code>	681	<code>\z@</code>	1788, 1791, 1792, 1796
<code>\varphi</code>	933	<code>\Zeta</code>	879
<code>\varpi</code>	937	<code>\zeta</code>	906
<code>\varrho</code>	936	<code>\zf@family</code>	519
<code>\varsigma</code>	918	<code>\zf@fontspec</code>	475
<code>\varTheta</code>	891	<code>\zf@update@ff</code>	714, 717