Towards a new Math Font Encoding for (LA)TEX

Matthias CLASEN[1] & Ulrik VIETH[2]

[1] *Albert-Ludwigs-Universität Freiburg*
Institut für Mathematische Logik
D-79104 Freiburg, Germany
clasen@mathematik.uni-freiburg.de

[2] *Heinrich-Heine-Universität Düsseldorf*
Institut für Theoretische Physik II
D-40225 Düsseldorf, Germany
vieth@thphy.uni-duesseldorf.de

Abstract. *This paper presents a snapshot of ongoing work towards a prototype implementation of new 8-bit math font encodings for (LA)TEX, based on the 'Aston' proposal presented at the TUG '93 conference. The design goals and technical considerations that have led to the present font table layouts are summarized and the contents and organization of the individual encodings are presented in detail. Finally, some alternative approaches and some remaining open problems are discussed.*

Keywords: mathematical typesetting, font encodings, symbol fonts

1. Introduction and Background

When the arrival of TEX 3.0 [32] made it possible to overcome the limitations of 7-bit fonts, it did not take long until an encoding for 8-bit text fonts to support the needs of European languages was developed and agreed at the European TEX conference in September 1990 [2,3,6]. When the first implementations of these 'Cork' encoded fonts became available during 1991, it was soon recognized that new 8-bit math font encodings were also eventually needed to overcome the inter-dependencies between text fonts and math fonts, and the first informal discussions on math font encoding issues began taking place at workshops at TEX conferences and on the TEX-FONTS mailing lists.

To focus these efforts, a working group on extended math font encodings was initiated in 1992 and established as a joint LATEX3 Project/TUG Technical Working Group, officially chaired by Barbara Beeton of the American Mathematical Society (AMS). In the summer of 1993 GUTenberg sponsored a three month research activity of Justin Ziegler at the University of Mainz to work on this subject together with members of the LATEX3 Project team. During this time a number of technical studies were conducted and an outline of a proposed new math font encoding was put together and presented in a workshop session at the TUG '93 conference [17]. Following the conference an electronic mailing list was established, and a number of details of this 'Aston' proposal were worked out during six weeks of very active discussions. Unfortunately, work came to a halt shortly after Justin Ziegler had to leave the scene, and the only actual result was a LATEX3 Technical Report [42] published in mid-1994, which went largely unnoticed among the TEX community despite several public announcements [36, 37].

After this first effort was left unfinished, it somehow took until early 1997 before discussions on math font encoding issues were revived on the `LATEX-L` mailing list and work towards an implementation of the 'Aston' proposal as outlined in Justin Ziegler's report was resumed. As a first step, a `fontinst`-based prototype implementation [5] was developed by Matthias Clasen with contributions from Ulrik Vieth and others. At the time of writing (in early 1998) preliminary implementations of the Computer Modern, Concrete, and AMS Euler versions are near complete, but they may still require further testing and fine-tuning. Additional versions based on standard PostScript fonts (Times Roman with Adobe Symbol), the commercial MathTime and Lucida New Math font sets, or the Mathematica symbol fonts are under development and will hopefully be completed by the time of the conference.

In the following sections of this paper we shall present an overview of the ongoing work on new math font encodings and their implementations. We shall begin with a review of the motivation behind the Math Font Group's activities and the situation of math fonts vs. text fonts in general, followed by a brief summary of the contents of TEX's old math font encodings, and an outline of the proposed new encodings. After a discussion of the design goals and design choices influencing the organization of the font table layouts, we shall review the contents of the individual font encodings in detail. Finally, we shall consider the user interface and some implementation issues, concluding with a discussion of some alternative approaches and some open questions.

It should be noted that the details of the new math font encodings and the font table layouts presented in this paper are to be considered as preliminary and subject to change, depending on the feedback we might get. For further information about submitting suggestions or comments see section 10.

2. Motivation

Based on the number of assigned `UniqueIDs` for Adobe Type 1 fonts it was estimated in 1997 that there were more than 90,000 text fonts available, while there were only relatively few math fonts, and even fewer of these were usable with TEX [15]. Among the available math fonts for TEX we have Computer Modern [29], Concrete with AMS Euler [31], Concrete Math [40], Times Roman with Adobe Symbol [21], the commercial MathTime and Lucida New Math font sets, and the Mathematica symbol fonts [34, 41]. Each of these font sets uses a different encoding, and each requires its own selection of (LA)TEX macros.

At the time when the Math Font Group began its work in 1993, setting up LATEX (actually LATEX 2.09) to use any other math fonts than 7-bit Computer Modern was a very difficult task and often required writing complex *ad hoc* macro packages for each font set. In the meantime, the introduction of LATEX 2$_\varepsilon$ as the new standard version of LATEX has eased the situation a bit, but some of the problems remain.

With the introduction of the New Font Selection Scheme (NFSS2) as the new standard font selection interface [8], using a different text font family, either employing the old (OT1) or the 'Cork' (T1) encoding, has become almost trivial. Setting up a new text font for use with LATEX is no longer a big deal either since almost any PostScript font can be set up with little effort using the widely-established `fontinst` utility [7, 16, 20, 21, 38] or recently developed front-ends to `fontinst` such as `VFInst` [11, 14]. In addition, the text companion encoding (TS1), originally suggested as part of the 'Aston' proposal, has also become a reality [27, 28], providing a place for non-mathematical symbols that will ultimately be taken out of the math fonts.

While it is true that using a different math font family has become a little easier with LATEX 2$_\varepsilon$, this is still far from trivial and a dedicated LATEX package is still required in each case. (Fortunately, such support packages are available in most cases nowadays, either contributed by volunteers [23, 24, 34, 40] or commissioned by font suppliers [35, 39].) Setting up new math font sets for use with LATEX also remains a difficult task due to the diversity of different encodings and the extra complications arising from inter-dependencies between text fonts and math fonts in the existing encodings.

Some front-end utilities to simplify the installation of new math fonts have been developed recently, but the scope of utilities such as `MathInst` or `MathKit` [12–14] remains limited, and a certain amount of manual intervention is still required for fine-tuning and the finishing touches. Moreover, the proper choice of matching text and math typefaces that mix well is a delicate issue that requires taste and typographical experience [4, 14].

3. Summary of the old math font encodings

Before we go into the details of the proposed new math font encodings, it may be a good idea to review how the old math font encodings are organized and which problems are caused by this setup. In the default (LA)TEX math setup (using Computer Modern fonts) the mathematical glyphs are arranged into four families, whose layout is explained in Appendix F of *The TEXbook*.

Family 0 contains an OT1-encoded upright roman font (e. g. `cmr`), which is used for a variety of purposes: In its primary function it serves as the operator font for typesetting multi-letter operators such as 'log' or 'lim', and as the `\mathrm` math alphabet, which is used by default for digits and the uppercase Greek letters, while the Latin alphabet and the lowercase Greek are taken from the math italic font. In addition to that, it also provides access to various symbols such as the math accents, some delimiters, and the '+' and '=' signs. Since the OT1 encoding includes the Greek capitals, it is not possible simply to substitute it by a T1-encoded text font. Moreover, since the '=' sign also serves as the extension module for double arrows, this font also has to match the size and shape of the math symbol font containing the arrowheads, which prohibits using another a text typeface in its place.

Family 1 contains an OML-encoded math italic font (e. g. `cmmi`), which is used as the default font for most single-letter identifiers. Even though this font contains a complete Latin and Greek alphabet in italic shape, the Greek capitals are traditionally typeset in upright roman, unless the `\mathnormal` math alphabet is selected explicitly.[1] Apart from the alphabets, this font also contains a few mathematical symbols and, strangely, the oldstyle digits '0123456789' which are not needed in math typesetting. Most of the symbols, however, such as the slash or the punctuation marks, are needed for kerning with the Latin or Greek letters, and therefore have to live in the same font table. Interestingly, the MathTime MY1 font layout derived from OML replaces the oldstyle digits and a few symbols by the upright Greek capitals, so that these may be kerned with the italic lowercase Greek letters as well.

Family 2 contains an OMS-encoded math symbol font (e. g. `cmsy`), which provides the Calligraphic alphabet (`\mathcal`) and most of the mathematical symbols, but also some text symbols that do not really belong in a math font such as '¶' and '§' or the playing card symbols. Among the mathematical glyphs there are a few symbols such as '∇' or '∐' that strongly depend on the design of the Greek letters in the text roman or math italic fonts and therefore will

1 This setup matches the American typesetting tradition, but typesetting rules applicable in certain fields of science may call for a different setup [1,25]. Moreover, still other conventions apply in the French typesetting tradition, which used to prefer upright shape not only for Greek capitals, but also for lowercase Greek letters.

be affected by font changes, while most of the geometric symbols could be left unchanged. Finally, this font also contains the small radical sign ('$\surd$'), which is likely to cause problems for non-TeXnical software since it happens to hang below the baseline for obscure technical reasons.

Family 3 contains an OMX-encoded math extension font (e.g. `cmex`), which holds all the big operators, growing delimiters and radicals, and the wide accents. These glyphs are all specific to TeX in that they have very unusual font metrics. Traditionally, all the big operators, delimiters and radicals are positioned to hang below the baseline, which is due to technical requirements for the radicals, but otherwise seems to be just an odd coincidence that might be the outcome of historical circumstances.[2] Apart from the unusual glyph placement, most symbols in this font also make use of special TFM features, such as `charlist` links between glyphs or `extensible` recipes.

The first four math families are the very minimum required for typesetting any math formula with TeX. In particular, families 2 and 3 are special in that they are expected to have a certain number of `\fontdimen` parameters, mostly dealing with the placement of sub- and superscripts, numerators and denominators in fractions, and limits on big operators.

Plain TeX and LaTeX 2.09 allocate a number of additional families for extra math alphabets such as text italic, bold, typewriter, or sans serif, some of which may be of interest only in typesetting computer programs while others may be needed in mathematics as well. In LaTeX 2_ε these math families are no longer preloaded; instead they are allocated dynamically when they are used.

If the latexsym package is used to access the extra LaTeX symbols that used to be built into LaTeX 2.09, one extra math family is added, which yields a total of five families. If the amssymb package is used instead, providing access to the AMS-LaTeX symbol complement and a Blackboard Bold (or 'open face') math alphabet, two extra math families are needed, resulting in a total of six families. In addition, a Fraktur (or 'black letter') math alphabet is also provided, which is allocated dynamically only when it is actually used.

Since there is a significant overlap between the LaTeX symbols and the AMS symbols, both sets of symbols as well as the Blackboard Bold and Fraktur alphabets can be allocated in the new math font encodings within six families. Some non-mathematical symbols such as the Yen sign, the Maltese cross, or the circled 'R' and 'S' will be taken out, and a new place them will have to be found, either in the TS1 text symbol encoding or in a dingbats font.

2 The Error Log of TeX [30] seems to indicate that TeX78 used to center big delimiters on the math axis with respect to their depth only while TeX82 uses their height and depth, which would allow placing the glyphs in a vertically centered position, better suited for use with non-TeXnical typesetting systems. (Log entry #168, dated 21 March 1978.)

4. Outline of the new math font encodings

The primary goal of the Math Font Group is to develop a comprehensive new math font encoding for use with (LA)TEX (and possibly other typesetting systems as well), consisting of some six 256-character font tables that fulfill certain requirements. They should contain all symbols needed to ensure compatibility with the existing encodings and provide enough room for new symbols that have been requested or suggested during the group's discussions.

The original 'Aston' proposal [17] consisted of six encodings (not counting the TS1 text symbol encoding), which were organized as follows:

T1	math operators ('Cork' text encoding)
MC	math core encoding
MX	math extension encoding
MSP	math symbol primary encoding
MS1	math symbol extra 1 encoding
MS2	math symbol extra 2 encoding

In this proposal the MC-encoded math core font more or less assumes the role of the OML-encoded math letters font (`cmmi`), the MX encoding replaces OMX (`cmex`), and the three MSn encodings replace OMS (`cmsy`) and the additional AMS symbol fonts (`msam` and `msbm`).

Each of the new encodings will provide room for up to 256 glyphs, which may seem like a lot given only 128 glyphs in the old encodings, but even that may still represent a severe constraint if all the requests for new symbols are taken into account. A particular problematic case in this regard is the math extension encoding (MX), where the addition of each new pair of big delimiters easily consumes between 8 and 16 slots for the usual four sizes and an extensible version. The situation only gets worse if additional intermediate or extra-big sizes of delimiters and radicals were to be included as well, following an idea first implemented in Yannis Haralambous' yhmath package [9].

As a result of all this, the original proposal was modified and the MX encoding was split into two MXn encodings, so that we end up with:

MXP	math extension primary encoding
MX1	math extension extra 1 encoding

In the basic version, using no more than the traditional sizes, both of these font layouts will contain a number of empty slots in a so-called "variable area", which may be filled in an extended version at the font designer's disposal without requiring changes to the (LA)TEX macros. There may even be several extended versions, adding either the intermediate or the extra-big sizes or both.

5. Design goals and design choices

The actual font layouts of the new math font encodings represent the result of a trade-off between a number of factors, such as technical constraints, the requirement for compatibility with the old encodings, the desire for simplicity and orthogonality, and the availability of symbols in various existing font sets. Most importantly, the new font layouts should be designed to avoid the most irritating problems of the old encodings arising from the fact that some math symbols are taken from text fonts or used for several different purposes.

Compatibility: The most important design goal will be to ensure compatibility with the symbol complement provided in the old encodings. In practice, this means that all existing symbols must be represented somewhere in the new encodings, with the exception of a few non-mathematical glyphs that will be removed explicitly. In addition, the new encodings should also provide extra slots for composite glyphs that are presently constructed by combining several glyphs and for glyphs that are used for multiple purposes.

Since the number of TeX's math families is limited to a total of 16, the new math setup should be implemented within the same number of families as the old setup. In particular, it should be possible to achieve compatibility with Plain TeX or LaTeX within four families (encoded as T1, MC, MSP, MXP) and with AMS-TeX or AMS-LaTeX within six families (adding MS1 and MS2). If there is enough room, it may be a good idea to include the LaTeX symbols in the first four families as well, so that compatibility with the LaTeX 2.09 symbol complement could be achieved with one family less than previously required.

Technical Constraints: Apart from compatibility issues, technical requirements are another important factor to consider, since they often lead to implications that some groups of symbols must be kept together in the same font table if they are linked by means of kerning, ligaturing, or other special TFM features.

Kerning will play an important role when it comes to letters and letter-like symbols. In the old OML encoding only the letters from the italic Latin and Greek alphabets could be kerned with each other and the punctuation symbols, while this wasn't possible for the upright Greek capitals taken from OT1. The MathTime MY1 encoding did a little better in this regard and replaced some of the non-mathematical glyphs from OML by the upright Greek capitals to enable kerning at the expense of using a non-standard encoding. In the proposed new encodings the math core font (MC) will carry this idea further and provide room for two complete sets of Greek (upright and italic) along with the default Latin alphabet. Furthermore, it will include the basic-size delimiters to allow kerning between the letters and the delimiters as well if required.

Another area where kerning will play a role is the construction of extensible horizontal arrows. In the old encodings building blocks for long arrows used to come from several different font tables (OT1, OML, OMS) and putting them together had to be done using a `\joinrel` macro. In the new encodings care will be taken to put all the building blocks into one font table (MSP), so that the same effect may be achieved automatically through kerning.

Apart from the implications of kerning or ligaturing, other technical constraints arise when it comes to extensible symbols linked through special TFM features. All the big delimiters of increasing height or wide accents of increasing width which are part of a chain must live together. The positive side-effect of this is that it allows all TEX-specific symbols to be kept together in the MXn encodings, so that the remaining MSn symbol fonts will be unaffected.

Simplicity and Orthogonality: As already mentioned, the new encodings should be designed to avoid the problems of the old encodings concerning the organization of font tables. For instance, to avoid inter-dependencies between text fonts and math fonts, the T1-encoded operator font in family 0 should not be used for any other purpose than typesetting upright roman letters. Similarly, glyphs serving multiple purpose should be given separate slots for each of their roles, e. g. for use as binary operators or as arrow extension modules.

Apart from avoiding unwanted inter-dependencies, orthogonality also means keeping all glyphs of similar design together and keeping glyphs of different designs in separate font tables (e. g. geometric vs. non-geometric symbols). As for the distribution of symbols among the MSn encodings, design similarity will be used as a guideline, but a trade-off will ultimately have to be made between following the principle of design similarity and taking the availability of symbols in various font sets into account.

Slots: Most of new encodings (with the exception of MXn) will provide room for a complete math alphabet, each in a different style, consisting of digits, uppercase and lowercase letters, and the dotless 'i' and 'j'. To allow switching math alphabets, these glyphs will have to share the same slots in each font table, preferably using the same encoding as T1. If some characters are unavailable in some styles, the corresponding slots should be left empty and not be allocated otherwise to avoid surprises.

As another common feature, each of the new encodings (including MXn) will reserve slot 32 as a 'space' glyph, which may help to make the fonts usable with other software, even if such a glyph may not be needed for math typesetting with TEX. Finally, slot 127 (ASCII `DEL`) will be used as the 'skewchar', which only serves to encode accent positioning information in fake kern pairs and will never appear on the output side for actual typesetting.

6. Details of the new font encodings

6.1. The T1-encoded math operator font

Family 0 of the new math setup will contain a T1-encoded upright roman math operator font. This font will only be used for typesetting multi-letter operators such as 'log' or 'lim', and to access the Latin letters of the `\mathrm` math alphabet. Unlike its OT1-encoded counterpart, it will not be used for anything else, so as not to create any unwanted inter-dependencies between text fonts and math fonts.

The decision to use the T1 encoding is based on the assumption that this encoding will be used as the default encoding for text fonts, which would allow arbitrary text fonts to be used for the multi-letter operators. As long as only the ASCII letters are used (excluding the dotless 'i' and 'j'), any other encodings with an ASCII-subset will do, including OT1, LY1 (Y&Y's TeX and ANSI), `8a` (Adobe Standard), `8r` (TeX Base1), or even T4 (African FC-fonts [26]).

Although the math operator font will use the same encoding as a text font (T1 by default), it does not necessarily have to be a text font. In particular, it may have the glyph widths and italic corrections adjusted to produce good subscript and superscript positioning, as long as this is not to the detriment of setting multi-letter operators or textual subscripts.

6.2. The MC-encoded math core font

The MC-encoded math core font (see Tables 1 and 2) will be used as the real work-horse of the new math font encoding. It will include two complete Greek alphabets (upright and italic) along with the default Latin alphabet (italic) and the default numerals (upright). In addition to the slash and the punctuation marks already included in OML for kerning reasons, the MC encoding will also include the basic-size delimiters (parentheses, square brackets, curly braces, etc.), so as to allow kerning between the letters and the delimiters.

Other letter-like glyphs that strongly depend on the font design such as the Fraktur letters ('ℜ', ℑ'), the inverted Greek letters ('∇', '∐', '℧', etc.), the Hebrew letters ('ℵ', 'ℶ', 'ℷ', 'ℸ'), and the so-called Humanist ("shapy") glyphs ('∗', '†', '‡', etc.) will also be included in the MC encoding, but the motivation for this is orthogonality of the font layouts. As long as the symbol complement in the MSn encodings can be restricted to geometric ("non-shapy") symbols, the symbol fonts will not be directly affected from changes of the design, and it may be possible to use the same version of MSn in combination with several different versions of MC (e. g. Computer Modern, Concrete, and Euler).

Unlike its OML-encoded counterpart in family 1, the math core font will be equipped with enough `\fontdimen` parameters to be used as family 2, since the positioning of subscripts and superscripts depends much more on the letters and letter-like symbols than on the geometric symbols. An additional benefit of this arrangement is that everything special to TEX such as extra font dimensions or adjustments to the glyph widths and italic corrections may be confined to the MC font, while all the MSn fonts may be implemented as normal fonts.

Since the MC encoding will contain two complete sets of Greek in different font positions, the implementation of font changing commands such as `\mathrm` or `\mathnormal` will have to be reconsidered. To facilitate switching between the two sets, the math core font will include a so-called Greek "control glyph" and a special ligaturing program, which maps ligatures between the control glyph and italic Greek letters to upright Greek letters and vice versa.

As for bold, bold italic, or sans serif glyphs, it is assumed that these will be taken from separate bold or sans serif versions of the math fonts sharing the same encoding. If a complete math formula is to be set in a bold context such as a section heading, a complete set of bold fonts will be needed in all encodings. However, if only the letters and letter-like symbols are used in a bold or sans serif math alphabet, it may be sufficient to provide only a bold/bold italic or sans/sans oblique version of the T1/MC encodings. Furthermore, if only a few bold glyphs are requested, these can be set using macros similar to `\boldsymbol` from AMS-LATEX or the bm package from the LATEX tools bundle.

6.3. The MS-encoded math symbol fonts

The MSP, MS1, and MS2 encodings (see Tables 3 to 5) will contain all the geometric symbols from OT1, OML, OMS, and the AMS symbol fonts, plus a number of frequently requested new glyphs. Among them there will be some obvious counterparts to existing glyphs, such as a right-to-left `\mapsfrom` glyph or some ready-made negated versions. In addition, some glyphs that were used for several different purposes (such as the '$-$' and '$=$' signs serving as extension modules for horizontal arrows) will be assigned separate slots. Finally, the upper half of the MS2 encoding will provide room for a new development: an "arrow construction kit", first suggested by Alan Jeffrey [18].

In addition to the geometric symbols, each symbol font will contain one additional math alphabet, distributed as follows:

MSP	will contain Calligraphic or Script
MS1	will contain Blackboard Bold ('open face')
MS2	will contain Fraktur ('black letter')

As to whether a Calligraphic or Script alphabet should be included in MSP, the decision will be left to the font designer and may depend on the choice of available font sets. If both are needed or if the default is not suitable for a particular application, it is always possible to load another font and set up another math alphabet, e. g. to replace the Calligraphic letters from Computer Modern by Euler Script or Ralph Smith's Formal Script font (`rsfs`).

To achieve compatibility with Plain TEX or LATEX within four families, all the geometric symbols from OT1, OML, and OMS will have to be kept in the MSP encoding. As long as there is room, a selection of additional symbols from the LATEX or AMS symbol fonts may be added, while the remaining symbols will end up in the MS1 or MS2 encodings.

As for the distribution of symbols among the MSP and MS1 or MS2 encodings, the 'Aston' proposal originally gave a higher priority to the design similarity argument, keeping symbols of similar design together regardless of whether or not these were available in some font sets. During the implementation work, however, it was recognized that much better results could be achieved if the availability of symbols was taken into account. Consequently, the MSP and MS1 font tables were reshuffled so that a near-complete version of MSP combined with a near-empty version of MS1 could be achieved, instead of having two half-complete font tables in the implementation based on the Mathematica symbol fonts. It remains to be seen if further reshuffling will be needed to achieve the best possible results with the MathTime or Lucida New Math font sets.

The arrow kit. The upper half of the MS2 encoding table is reserved for the arrow construction kit, first suggested in [18]. This means on the one hand that a lot of pieces for building arrows — heads, stems, and tails — are collected here. On the other hand, it means that TEX's ligature mechanism is used to automatically select the proper glyphs. We have implemented the following syntax for arrows, which is inspired by, but different from the syntax proposed in [19]:

$$\langle \textit{left\_head} \rangle [\langle \textit{none} \rangle][\langle \textit{stem\_modifier} \rangle]\langle \textit{extension} \rangle{*}\langle \textit{right\_head} \rangle[\langle \textit{none} \rangle]$$

Here ⟨*left_head*⟩ and ⟨*right_head*⟩ can be chosen among 24 arrow heads or ends. The ⟨*extension*⟩ pieces can specify a negation, a gap, or an extension, which may also be gaped or negated. The ⟨*stem_modifier*⟩ can be used to produce dotted or squiggly arrows.

An additional benefit of this approach to arrow construction is that only the head glyphs and the control glyphs used for the ⟨*stem_modifier*⟩ and ⟨*extension*⟩ pieces are directly accessed through macros. The rest of the arrow kit is a variable area that may vary from font to font without sacrificing compatibility. (Each font has to provide a proper ligature program though!)

6.4. The MX-encoded math extension fonts

As mentioned earlier, the original 'Aston' proposal was modified, splitting the MX math extension encoding into MXP and MX1 (see Tables 6 to 8) in order to provide more room for additional sizes. The MXP encoding will be assigned to family 3, taking the place of OMX, while MX1 will be optional.

To achieve compatibility with Plain TEX or LATEX within four families, the MXP encoding will have to contain all the existing TEX-specific glyphs from OMX (`cmex`), as well as the small radical sign from OMS (`cmsy`) and the extra sizes of wide accents from the AMS symbol fonts (`msbm`). Additional intermediate or extra-big sizes of delimiters and radicals as implemented in the `yhcmex` font [9] may be provided at the font designer's disposal, filling the empty slots in the variable area as appropriate.

In addition to the standard glyphs, a number of new extensible symbols have been suggested, including obvious counterparts to existing symbols as well as a variety of new operators and delimiters. Most of these new symbols will have to be assigned to the MX1 encoding due to space limitations in MXP. However, there may be good reasons to include at least a few of the most frequently suggested ones in MXP, especially if they are already implemented in some existing font sets. One candidate we have included are the semantic brackets, which are provided in the St. Mary's Road symbol font [22] as well as in the Mathematica and Lucida New Math font sets.

7. The User Interface

Together with the font layouts and the virtual font implementations, we have developed a set of macros that make the new math font encodings accessible from within LATEX, consisting of the following components:

`fontmath.cfg` is a configuration file that may be used to generate a LATEX format with the new math setup preloaded. It provides all the macros needed for compatibility with the standard LATEX math setup, consisting mostly of `\DeclareMathSymbol` and similar statements that establish the control sequence names to access the math symbols.

`newmath.sty` is a LATEX package file that provides some additional macros which may be enabled through package options such as amssymb (extra symbols in the MS1 and MS2 encodings), extraops (extra operators and delimiters in the MX1 encoding), or accents (special macros for over- and underaccents). It also provides the infrastructure for changes to the math layouts through so-called `*.mfd` files (see below).

`oldmath.sty` is a LaTeX package file that reverts to the traditional LaTeX math setup. This can be used to process old files which require the traditional math setup, even if the LaTeX format includes the new math setup.

`*.mfd` **files** (math font definitions) are used to describe the setup of the individual math layouts. Each '`.mfd`' file establishes a mapping between the symbolic names of math families and the LaTeX font attributes that correspond to one particular set of virtual fonts implementing the new encodings. In addition, the '`.mfd`' files may also be used to introduce adjustments to the relative sizes of subscripts and superscripts and the various parameters influencing TeX's math typesetting algorithms, such as `\delimiterfactor` or `\thinmuskip`, `\medmuskp`, `\thickmuskip`.

`*.fd` **files** (font definitions) are used as usual to establish a mapping between symbolic LaTeX font attributes and external font file names. The eventual font names of the virtual fonts and the base fonts providing extra symbols have not been finalized yet and may be subject to change.

The packages newmath and oldmath have been designed to work equally well with standard LaTeX formats and modified LaTeX formats containing the new math setup. With the new math setup, changing the math layout no longer requires loading a special-purpose macro package and simply means adding the appropriate option (cm, concrete, euler, mathptm, mathematica, mathtime, lucida, etc.) to the `\usepackage{newmath}` statement.

So far, the development of the user interface has focused exclusively on LaTeX. However, a Plain TeX interface also needs to be developed eventually, so as to fulfill the original goal of providing a new encoding for use with any common TeX format. For most of the code, preparing a Plain TeX version will simply amount to rewriting the `\DeclareMathSymbol` statements into `\mathchardef` or `\mathcode` assignments, which should not be too difficult.

8. Implementation Issues

8.1. Virtual Fonts

As a primary product of our implementation work, we have developed a number of sets of virtual fonts implementing the new encodings for each of the math layouts. The purpose of these virtual fonts primarily consists in remapping glyphs taken from several existing or newly-developed base fonts. In addition, they might also be used to construct some composite glyphs as a substitute for symbols which are not readily available from a given set of base fonts.

In the case of METAFONT fonts (Computer Modern, Concrete, Euler, etc.) some of the missing symbols have been supplied by means of relatively trivial METAFONT work, especially in simple cases of reflected or inverted symbols. Some more extensive METAFONT work is required, however, in the design of the arrow construction kit and the upright lowercase Greek alphabet.

In the case of PostScript fonts (Times, MathTime, Mathematica, Lucida, etc.) only the available glyphs from a given set of base fonts have been used, so as to avoid mixing different typefaces or introducing undesirable inter-dependencies between PostScript and METAFONT fonts. Unfortunately, this means that a number of symbols will be missing, especially in the case of MathTime which doesn't cover the AMS and LATEX symbols.

A particular interesting question arises when it comes to the choice of math alphabets to be used in the MSn encodings. While there might be only one sensible choice in the Computer Modern math layout for the Fraktur alphabet in MS2 (`eufm`), there are quite a number of different alternatives that may be used for the Calligraphic or Script alphabet in MSP (`cmsy`, `eusm`, `rsfs`) and for the Blackboard Bold alphabet in MS1 (`msbm`, `bbold`, `bbm`). Similarly, in the MathTime version, the Script alphabet in MSP may be chosen from MathScript (`MTMS`) or Adobe MathPi, while Adobe MathPi might be the only choice for the Blackboard Bold and Fraktur alphabets in MS1 and MS2.

If all possible combinations of these choices were to be supported, the number of virtual fonts required to implement the math layouts would soon become uncomfortably large. As an alternative approach, it might be worthwhile to consider implementing the MSn encodings as doubly-virtual fonts, composed of two subset encodings containing only the alphanumerical glyphs or the geometric symbols, the latter of which, in turn, may be implemented as a virtual font drawing characters from a variety of base fonts.

Concerning the choice of the default font sets to be used in the reference implementation of the Computer Modern math layout, sticking to widely-available base fonts (CM, Euler, AMS symbol) may be preferable, since this will allow to provide an implementation in Adobe Type 1 format, which is becoming more and more important in view of the recent development of PDFTEX.

8.2. METAFONT work

In addition to the virtual fonts implementing the new encodings, a number of METAFONT fonts have also been developed, serving as base fonts that are used to supply some extra symbols. Apart from some simple cases of reflected or inverted symbols, the METAFONT work has concentrated on the design of the arrow construction kit and the upright lowercase Greek alphabet.

Arrow Kit: The arrow construction kit described in section 6.3 consists of some 120 glyphs, containing all sorts of left and right heads, tails, extension pieces, gaps, negation slashes, gaped and negated extension pieces, as well as building blocks for dotted or squiggly arrows. The designs of all these glyphs were heavily borrowed from Computer Modern and the AMS fonts, but some effort was taken to make them consistent and to adapt the AMS glyphs to take into account the design changes that were introduced by Don Knuth in 1992. Since the designs of the arrows only depend on the *rule_thickness* parameters, the arrow kit should be usable with each of the Computer Modern, Concrete, and Euler math layouts without any changes.

Upright Greek: Probably the most interesting application of METAFONT was the design of an upright lowercase Greek alphabet. Although it may appear straight-forward to start out with the Computer Modern `greekl.mf` [29] and a driver file which sets $slant := 0$, it quickly turned out that the design of a nice-looking upright Greek alphabet was quite a bit more complicated than that. Since the character programs in `greekl.mf` apparently have been designed with $slant := 1/4$ in mind, a number of letters look unbalanced without the slant. In some cases this could be repaired with minor adjustments to the character programs. In other cases some more extensive changes were needed, amounting to a complete redesign of the character for use in the upright version.

Still another problem related to METAFONT fonts showed up when characters from different base fonts were combined in the same font table, revealing some inconsistencies or deficiencies of the character designs. For instance, the Hebrew letters ('$\beth$', '$\gimel$', '$\daleth$') taken from the AMS fonts turned out to be a little too heavy compared with '$\aleph$' from `cmsy`, while they are fine compared with '**ℵ**' from `eusm`. As another example, the variant 'kappa' ('$\varkappa$') and the 'digamma' ('$\digamma$') from the AMS fonts turned out to produce poor results with Concrete parameters.

9. Discussion

9.1. Is there a better way?

Replacing T1/MC? In the present proposal, the MC font is supposed to include the default Latin alphabet (in italic) combined with two sets of Greek (in italic and upright), while the T1 font provides the upright Latin alphabet. Since this may cause complications when trying to switch math alphabets for Greek letters, an alternative idea was brought up: This was to replace the T1/MC combo by a set of two fonts in a math-alternate encoding, each containing only one set of Latin and Greek. Although this approach may seem attractive at first sight, it loses the ability to have kerning between the upright and italic Greek letters, so it provides no advantage over OML and even falls behind what is already possible in MY1. In addition, it also loses the ability to use a T1-encoded text font directly as the math operator font.

Splitting MX? In the present proposal, it was decided to split the originally proposed MX encoding into MXP and MX1, providing room for additional sizes at the font designer's disposal. For compatibility, all existing symbols from OMX must be included in MXP, which leaves MX1 for the newly introduced symbols, except perhaps for a few of the most frequently requested ones that might be accepted in MXP. The question remains whether it would be preferable to have only a single MX font by default containing all extensible symbols, which would be replaced by two MXn fonts only when an optional LaTeX package is loaded. Given the number of proposed new delimiters, it appears that an extra MX1 encoding may be unavoidable anyhow. It just remains to be seen how many font sets will actually provide the necessary glyphs to implement them.

Design similarity or availability? In the original 'Aston' proposal, design similarity was considered to be one of the most important factors determining the distribution of symbols among the MSP and MSn encodings. While this is not a bad idea by itself, there is a danger that the argument may be taken too far, leading to the inclusion of exotic symbols that would be absent is most implementations. If the availability of symbols is taken into account, a different arrangement may be prefered, so that MSP will be as complete as possible while MSn might remain almost empty. From a practical point of view, availability may be more important when implementing virtual fonts based on characters taken from existing font sets, while design similarity might be prefered when designing new fonts from scratch.

9.2. More Symbols?

Apart from organizational questions, the decision as to which of the many proposed new symbols to include or to leave out has been one of most debated issues. In the end, only a few of these symbols have been implemented, while others have been put on hold until a need for them can be confirmed.

Among the newly-included symbols we have introduced some new variants of uppercase Greek letters. We have followed Adobe Symbol in having two variants of 'Upsilon', a straight one that looks like 'Y' and a curly one that looks like 'Υ'. Similarly, we have introduced two variants of 'Chi', a straight one that looks like 'X', and a curly one that has yet to be designed. In doing so, we followed a request by some high-energy physicists who wanted to have a variant Greek 'Chi' that could be clearly distinguished from a Latin 'X'.

Among the proposed symbols not currently implemented, we had suggestions for the variant lowercase Greek 'beta' (Unicode `U+03D0`), as well as the archaic Greek numerals 'Stigma' (`U+03DA`), 'Digamma' (`U+03DC`), 'Koppa' (`U+03DE`), and 'Sampi' (`U+03E0`). While some of them appear to be used in classical Greek text typesetting, their role in math typesetting remains unclear, except for the 'Digamma' ('$\digamma$') which seems to have a well-established meaning and needs to be kept for compatibility with the AMS symbol fonts. If needed, these symbols may be found in Unicode fonts such as Yannis Haralambous' OmegaTimes [10], but it seems unreasonable to include them in the new math fonts since they would be missing in nearly all implementations. The only exception are the Mathematica symbol fonts [41], which happen to provide the archaic Greek numerals both in uppercase and lowercase.

10. Further Information

The public discussions of the Math Font Group (MFG) have been taking place since August 1993 on an electronic mailing list `math-font-request`, which may be subscribed to by sending a message to the list owner at the address

`math-font-request@cogs.susx.ac.uk`

Archives of discussion papers, technical reports, preliminary test implementations, as well as complete mail archives in hypertext form are available at the MFG home page on the World Wide Web at

`http://www.tug.org/twg/mfg/`

There are plans to publish a more detailed report, once the remaining issues have been sorted out, either as a *TUGboat* article or as a LaTeX3 Report.

Acknowledgements

We wish to express our special thanks to Frank Mittelbach of the LaTeX3 Project for his encouragement during the early phase of the project and to Berthold Horn of Y&Y Inc. for supporting the project by kindly providing us with "evaluation copies" of the MathTime and Lucida New Math font sets. Thanks are also due to several working group members who helped to improve this article, among them Thierry Bouche, David Carlisle, Berthold Horn, and Chris Rowley. Finally, we wish to thank Karl Berry for providing the infrastructure on `www.tug.org` to establish a home page and an archive site for the Math Font Group.

Bibliography

[1] Claudio BECCARI, "Typesetting mathematics for science an technology according to ISO 31/XI", *TUGboat*, Vol. 18, No. 1, 39–48, 1997.

[2] Nelson BEEBE, "Character set encoding", *TUGboat*, Vol. 11, No. 2, 171–175, 1990.

[3] Janusz S. BIEŃ, "On standards for computer modern font extensions", *TUGboat*, Vol. 11, No. 2, 175–183, 1990.

[4] Thierry BOUCHE, "Sur la diversité des fontes mathématiques", *Cahiers GUTenberg*, , No. 25, 1–24, 1996.

[5] Matthias CLASEN, "An implementation of new math fonts", Technical Report, 1997–98, in preparation.

[6] Michael FERGUSON, "Report on multilingual activities", *TUGboat*, Vol. 11, No. 4, 514–516, 1990.

[7] Michel GOOSSENS, Frank MITTELBACH, and Sebastian RAHTZ, *The LATEX Graphics Companion*, Addison-Wesley, 1997.

[8] Michel GOOSSENS, Frank MITTELBACH, and Alexander SAMARIN, *The LATEX Companion*, Addison-Wesley, 1994.

[9] Yannis HARALAMBOUS, "My humble addtions to (LA)TEX mathematics", 1996, available from `CTAN:macros/latex/contrib/supported/yhmath/`.

[10] Yannis HARALAMBOUS, "ΩTimes and ΩHelvetica fonts under development: Step One", *TUGboat*, Vol. 17, No. 2, 126–146, 1996.

[11] Alan HOENIG, "The `VFinst` virtual font installer", Apr. 1997, available from `CTAN:fonts/utilities/vfinst/`.

[12] Alan HOENIG, "`MathInst`: New math fonts for TEX", May 1997, available from `CTAN:fonts/utilities/mathinst/`.

[13] Alan HOENIG, "Hundreds of new math fonts with `MathKit`", May 1997, available from `CTAN:fonts/utilities/mathkit/`.

[14] Alan HOENIG, *TEX Unbound: LATEX and TEX Stategies for Fonts, Graphics, and More*, Oxford University Press, 1998.

[15] Berthold HORN, "Where are the math fonts?" *TUGboat*, Vol. 14, No. 3, 282–284, 1993.

[16] Alan JEFFREY, "A PostScript font installation package written in TeX", *TUGboat*, Vol. 14, No. 3, 285–292, 1993.

[17] Alan JEFFREY, "Math font encodings: A workshop summary", *TUGboat*, Vol. 14, No. 3, 293–295, 1993.

[18] Alan JEFFREY, "Requirements for setting arrows", Aug. 1993, available from `CTAN:info/ltx3pub/arrowreq.tex`.

[19] Alan JEFFREY, "Control glyphs for arrows", Aug. 1993, available from `CTAN:info/ltx3pub/arrocont.tex`.

[20] Alan JEFFREY, "The `fontinst` package", Jun. 1994, available from `CTAN:fonts/utilities/fontinst/doc/fontinst.tex`.

[21] Alan JEFFREY, "PostScript font support in LaTeX 2_ε", *TUGboat*, Vol. 15, No. 3, 263–268, 1994.

[22] Alan JEFFREY and Jeremy GIBBONS, "The `stmaryrd` symbol font", 1994, available from `CTAN:fonts/stmaryrd/`.

[23] Frank JENSEN, "The `beton` package", 1992–95, available from `CTAN:macros/latex/contrib/supported/beton/`.

[24] Frank JENSEN, "The `euler` package", 1992–95, available from `CTAN:macros/latex/contrib/supported/euler/`.

[25] Jörg KNAPPEN, "Changing the appearance of math", in *Proceedings of the 7th European TeX Conference, September 14–18, 1992, Prague, Czechoslovakia*, edited by Jiří ZLATUŠKA, $\mathcal{CS}$TUG, 212–216, Sep. 1992.

[26] Jörg KNAPPEN, "Fonts for Africa: The FC-fonts", *TUGboat*, Vol. 14, No. 2, 104–106, 1993.

[27] Jörg KNAPPEN, "The release 1.2 of the Cork encoded DC fonts and the text companion symbol fonts", in *Proceedings of the 9th European TeX Conference, September 4–8, 1995, Arnheim, The Netherlands*, edited by Wietse DOL, $\mathcal{NTG}$, 239–254, Sep. 1995.

[28] Jörg KNAPPEN, "The DC fonts 1.3: Move towards stability and completeness", *TUGboat*, Vol. 17, No. 2, 99–101, 1996.

[29] Donald E. KNUTH, *Computer Modern Typefaces*, Vol. E of *Computers & Typesetting*, Addison-Wesley, 1986.

[30] Donald E. KNUTH, "The Errors of TeX", *Software — Practice and Experience*, Vol. 19, No. 7, 607–681, 1989, reprinted as Chapters 9–10 of [33].

[31] Donald E. KNUTH, “Typesetting *Concrete Mathematics*”, *TUGboat*, Vol. 10, No. 1, 31–36, 1989.

[32] Donald E. KNUTH, “The new versions of TEX and METAFONT”, *TUGboat*, Vol. 10, No. 3, 325–328, 1989.

[33] Donald E. KNUTH, *Literate Programming*, CSLI Lecture Notes No. 27, Stanford University Center for the Study of Language and Information/ Cambridge University Press, 1992.

[34] Jens-Peer KUSKA, “The Mathematica virtual font package”, Sep. 1997, available from `CTAN:fonts/psfonts/Mathematica3.0/`.

[35] Frank MITTELBACH and David CARLISLE, “The `mathtime` and `mathpi` packages”, 1997, available from `CTAN:macros/latex/packages/psnfss/`.

[36] Frank MITTELBACH and Chris ROWLEY, “LATEX3 News in 1993”, *TEX and TUG News*, Vol. 3, No. 1, 7–11, 1994.

[37] Frank MITTELBACH and Chris ROWLEY, “Math Font Encoding”, *TEX and TUG News*, Vol. 4, No. 2, 17–18, 1995.

[38] Sebastian RAHTZ, “Implementing the extended TEX layout using PostScript fonts”, *TUGboat*, Vol. 14, No. 2, 107–117, 1993.

[39] Sebastian RAHTZ and David CARLISLE, “The `lucidabr` package”, 1997, available from `CTAN:macros/latex/packages/psnfss/`.

[40] Ulrik VIETH, “The `concmath` package”, Oct. 1997, available from `CTAN:macros/latex/contrib/supported/concmath/`.

[41] Stephen WOLFRAM, *The Mathematica Book*, Wolfram Media/Cambridge University Press, 3rd ed., 1996.

[42] Justin ZIEGLER, “Technical Report on Math Font Encodings”, Technical Report, Jun. 1994, available from `CTAN:info/ltx3pub/l3d007.tex`.

Appendix: Font Tables

In the following font tables, the symbol ‘!’ is used to indicate slots reserved for special purposes, such as the skewchar in slot 127 and the Greek control glyph in slot 128 in the MC encoding. Another symbol ‘?’ is used to indicate slots reserved for unavailable symbols. Empty slots can be assumed to be unassigned, with the exception of slot 32, which is a space glyph in all encodings.

Table 1 – Math Core Encoding, CM version

	´0	*´1*	*´2*	*´3*	*´4*	*´5*	*´6*	*´7*	
´00x	⌊	⌋	⌈	⌉	⟮	⟯	∫	⎱	″0x
´01x	|	‖	⦀	♭	♮	♯	†	‡	
´02x	ℵ	ℶ	ℷ	ℸ	ℜ	ℑ	þ	ð	″1x
´03x	∅	ı	ȷ	≬	∝	∞	′	‵	
´04x		!	⟦	#	⟧	%	&	⅋	″2x
´05x	(	)	∗	⋆	,	·	.	/	
´06x	0	1	2	3	4	5	6	7	″3x
´07x	8	9	:	;	⟨	√	⟩	?	
´10x	@	A	B	C	D	E	F	G	″4x
´11x	H	I	J	K	L	M	N	O	
´12x	P	Q	R	S	T	U	V	W	″5x
´13x	X	Y	Z	[	\	]	ħ	ℏ	
´14x	d	a	b	c	d	e	f	g	″6x
´15x	h	i	j	k	l	m	n	o	
´16x	p	q	r	s	t	u	v	w	″7x
´17x	x	y	z	{	℘	}	ℓ	!	
´20x	!	Α	Β	Γ	Δ	Ε	Ζ	Η	″8x
´21x	Θ	Ι	Κ	Λ	Μ	Ν	Ξ	Ο	
´22x	Π	Ρ	Σ	Τ	ϒ	Φ	Χ	Ψ	″9x
´23x	Ω	ϒ	?	∇	∐	℧			
´24x	∂	α	β	γ	δ	ϵ	ζ	η	″Ax
´25x	θ	ι	κ	λ	μ	ν	ξ	ο	
´26x	π	ρ	σ	τ	υ	ϕ	χ	ψ	″Bx
´27x	ω	ε	ϑ	ϰ	ϖ	ϱ	ς	φ	
´30x		A	B	$\mathit{\Gamma}$	$\mathit{\Delta}$	E	Z	H	″Cx
´31x	$\mathit{\Theta}$	I	K	$\mathit{\Lambda}$	M	N	$\mathit{\Xi}$	O	
´32x	$\mathit{\Pi}$	P	$\mathit{\Sigma}$	T	$\mathit{\Upsilon}$	$\mathit{\Phi}$	X	$\mathit{\Psi}$	″Dx
´33x	$\mathit{\Omega}$	Y	?	Ϝ	э	℩	ƛ	ƛ	
´34x	∂	α	β	γ	δ	ϵ	ζ	η	″Ex
´35x	θ	ι	κ	λ	μ	ν	ξ	o	
´36x	π	ρ	σ	τ	υ	ϕ	χ	ψ	″Fx
´37x	ω	ε	ϑ	$\varkappa$	ϖ	ϱ	ς	φ	
	″8	″9	″A	″B	″C	″D	″E	″F	

Table 2 – Math Core Encoding, Euler version

	´0	´1	´2	´3	´4	´5	´6	´7	
´00x	$\lfloor$	$\rfloor$	$\lceil$	$\rceil$	$\lgroup$	$\rgroup$	$\int$	$\rmoustache$	″0x
´01x	$\vert$	$\Vert$	$\vert\vert\vert$	$\flat$	$\natural$	$\sharp$	$\dagger$	$\ddagger$	
´02x	$\aleph$	$\beth$	$\gimel$	$\daleth$	$\Re$	$\Im$	þ	ð	″1x
´03x	$\emptyset$	$\imath$	$\jmath$	()	$\propto$	∞	$\prime$	$\backprime$	
´04x		!	$\llbracket$	#	$\rrbracket$	%	&	&	″2x
´05x	(	)	$*$	$\star$	,	$\cdot$	.	/	
´06x	0	1	2	3	4	5	6	7	″3x
´07x	8	9	:	;	$\langle$	$\surd$	$\rangle$	?	
´10x	@	A	B	C	D	E	F	G	″4x
´11x	H	I	J	K	L	M	N	O	
´12x	P	Q	R	S	T	U	V	W	″5x
´13x	X	Y	Z	[	$\backslash$	]	$\hbar$	$\hslash$	
´14x	d	a	b	c	d	e	f	g	″6x
´15x	h	i	j	k	l	m	n	o	
´16x	p	q	r	s	t	u	v	w	″7x
´17x	x	y	z	{	$\wp$	}	ℓ		
´20x	(!)	A	B	Γ	Δ	E	Z	H	″8x
´21x	Θ	I	K	Λ	M	N	Ξ	O	
´22x	Π	P	Σ	T	Υ	Φ	X	Ψ	″9x
´23x	Ω	Y	(?)	∇	$\amalg$	$\mho$			
´24x	∂	α	β	γ	δ	ϵ	ζ	η	″Ax
´25x	θ	ι	κ	λ	μ	ν	ξ	o	
´26x	π	ρ	σ	τ	υ	ϕ	χ	ψ	″Bx
´27x	ω	ε	ϑ	$\varkappa$	ϖ	ϱ	ς	φ	
´30x		A	B	Γ	Δ	E	Z	H	″Cx
´31x	Θ	I	K	Λ	M	N	Ξ	O	
´32x	Π	P	Σ	T	Υ	Φ	X	Ψ	″Dx
´33x	Ω	Y	(?)	(?)	э	ı	ƛ	ƛ	
´34x	∂	α	β	γ	δ	ϵ	ζ	η	″Ex
´35x	θ	ι	κ	λ	μ	ν	ξ	o	
´36x	π	ρ	σ	τ	υ	ϕ	χ	ψ	″Fx
´37x	ω	ε	ϑ	$\varkappa$	ϖ	ϱ	ς	φ	
	″8	″9	″A	″B	″C	″D	″E	″F	

Table 3 – Math Symbol Primary Encoding

	´0	´1	´2	´3	´4	´5	´6	´7	
´00x	ˋ	ˊ	ˆ	˜	¨	̑	˚	ˇ	″0x
´01x	˘	¯	˙	⋯	⁞	→	←	↔	
´02x	⎵	⎴	̣̄	̇̄	̤̄	̈̄	̂̂	̂̄	″1x
´03x	̿	?	?	´	ˋ	$\smile$	$\frown$	$\asymp$	
´04x		-	$-$	$=$	$\equiv$	$+$	$\pm$	$\mp$	″2x
´05x		$\doteq$	$\div$	$\divideontimes$	$\times$	$\ltimes$	$\rtimes$	$\bowtie$	
´06x	?	?	?	?	?	?	?	?	″3x
´07x	?	?	$\forall$	$\exists$	$\nexists$	$\Finv$	$\Game$	℧	
´10x	$\neg$	$\mathcal{A}$	$\mathcal{B}$	$\mathcal{C}$	$\mathcal{D}$	$\mathcal{E}$	$\mathcal{F}$	$\mathcal{G}$	″4x
´11x	$\mathcal{H}$	$\mathcal{I}$	$\mathcal{J}$	$\mathcal{K}$	$\mathcal{L}$	$\mathcal{M}$	$\mathcal{N}$	$\mathcal{O}$	
´12x	$\mathcal{P}$	$\mathcal{Q}$	$\mathcal{R}$	$\mathcal{S}$	$\mathcal{T}$	$\mathcal{U}$	$\mathcal{V}$	$\mathcal{W}$	″5x
´13x	$\mathcal{X}$	$\mathcal{Y}$	$\mathcal{Z}$	$\int$	$\iint$	$\iiint$	$\oint$	$\oiint$	
´14x		?	?	?	?	?	?	?	″6x
´15x	?	?	?	?	?	?	?	?	
´16x	?	?	?	?	?	?	?	?	″7x
´17x	?	?	?					!	
´20x	$\cup$	$\cap$	$\sqcup$	$\sqcap$	$\wedge$	$\vee$	$\triangle$	$\triangledown$	″8x
´21x	$\uplus$	$\oplus$	$\ominus$	$\otimes$	$\oslash$	$\odot$	$\circ$	$\bullet$	
´22x	$\subseteq$	$\supseteq$	$\subset$	$\supset$	$\sqsubseteq$	$\sqsupseteq$	$\sqsubset$	$\sqsupset$	″9x
´23x	$\triangleleft$	$\triangleright$	$\trianglelefteq$	$\trianglerighteq$	$\ntriangleleft$	$\ntriangleright$	$\ntrianglelefteq$	$\ntrianglerighteq$	
´24x	$<$	$>$	$\leq$	$\geq$	$\leqslant$	$\geqslant$	$\lesssim$	$\gtrsim$	″Ax
´25x	$\nless$	$\ngtr$	$\nleq$	$\ngeq$	$\nleqslant$	$\ngeqslant$	$\not\lesssim$	$\not\gtrsim$	
´26x	$\prec$	$\succ$	$\preceq$	$\succeq$	$\preccurlyeq$	$\succcurlyeq$	$\precsim$	$\succsim$	″Bx
´27x	$\nprec$	$\nsucc$	$\npreceq$	$\nsucceq$	$\not\preccurlyeq$	$\not\succcurlyeq$	$\not\precsim$	$\not\succsim$	
´30x	$\ll$	$\gg$	$\lll$	$\ggg$	$\bigcirc$	$\square$	$\diamond$	$\Diamond$	″Cx
´31x	$\sim$	$\approx$	$\cong$	$\simeq$	$\backsim$	$\approxeq$	$\angle$	$\vert$	
´32x	$\nsim$	$\not\approx$	$\ncong$	$\eqsim$	$\backsimeq$	$\wr$	$/$	$\rightsquigarrow$	″Dx
´33x	$\leftarrow$	$\rightarrow$	$-$	$\leftrightarrow$	ˈ	ˈ	$\uparrow$	$\Uparrow$	
´34x	$\Leftarrow$	$\Rightarrow$	$=$	$\Leftrightarrow$	$\vert$	$\vert$	$\downarrow$	$\Downarrow$	″Ex
´35x	$\leftharpoonup$	$\rightharpoonup$	$\leftharpoondown$	$\rightharpoondown$	›	‹	$\updownarrow$	$\Updownarrow$	
´36x	$\vdash$	$\dashv$	$\in$	$\ni$	$\top$	╱	$\nwarrow$	$\nearrow$	″Fx
´37x	$\nvdash$	$\not\dashv$	$\notin$	$\not\ni$	$\bot$	╲	$\swarrow$	$\searrow$	
	″8	″9	″A	″B	″C	″D	″E	″F	

Table 4 – Math Symbol 1 Encoding

	´0	´1	´2	´3	´4	´5	´6	´7	
´00x									″0x
´01x									
´02x									″1x
´03x		ı	ȷ						
´04x		⊖	⊛	⊚	↻	↺	↶	↷	″2x
´05x	⇈	⇊	↿	⇃	↾	⇂	↰	↱	
´06x	?	𝟙	𝟚	?	?	?	?	?	″3x
´07x	?	?	⌞	⌟	⌜	⌝	⋋	⋌	
´10x		𝔸	𝔹	ℂ	𝔻	𝔼	𝔽	𝔾	″4x
´11x	ℍ	𝕀	𝕁	𝕂	𝕃	𝕄	ℕ	𝕆	
´12x	ℙ	ℚ	ℝ	𝕊	𝕋	𝕌	𝕍	𝕎	″5x
´13x	𝕏	𝕐	ℤ						
´14x		𝕒	𝕓	𝕔	𝕕	𝕖	𝕗	𝕘	″6x
´15x	𝕙	𝕚	𝕛	𝕜	𝕝	𝕞	𝕟	𝕠	
´16x	𝕡	𝕢	𝕣	𝕤	𝕥	𝕦	𝕧	𝕨	″7x
´17x	𝕩	𝕪	𝕫					!	
´20x	⩽	⩾	⪉	⪊	⪅	⪆	⪉	⪊	″8x
´21x	⋞	⋟	⪹	⪺	⪷	⪸	⪹	⪺	
´22x	≦	≧	≰	≱	≨	≩	≨	≩	″9x
´23x	⋖	⋗	≶	≷	⋚	⋛	⪋	⪌	
´24x	⪇	⪈	⊈	⊉	⊊	⊋	⊊	⊋	″Ax
´25x	⫅	⫆	⊈	⊉	⫋	⫌	⫋	⫌	
´26x	⋐	⋑	⋓	⋒	∴	∵	⋔	∔	″Bx
´27x	≓	≑	≒	≜	≗	≖	≏	≎	
´30x	⋏	⋎	⌣	⌢	⊻	⊼	⩞		″Cx
´31x	∡	∢	∤	∦	∣	∥	∤	∦	
´32x	⊩	⊪	⊨	⊫	⊮	⊯	⊭	⊯	″Dx
´33x	⊡	⊞	⊟	⊠	■	◊	⧫	★	
´34x	◀	▶	▲	▼	△	▽	▫	▵	″Ex
´35x	·	C	T	✓	∅	∝	⟍		
´36x									″Fx
´37x									
	″8	″9	″A	″B	″C	″D	″E	″F	

Table 5 – Math Symbol 2 Encoding

	´0	´1	´2	´3	´4	´5	´6	´7	
´00x									″0x
´01x									
´02x									″1x
´03x		ı	ȷ						
´04x					⌢	⌣	▵	▿	″2x
´05x	(	)	⟪	⟫	⦃	⦄	⟦	⟧	
´06x	0	1	2	3	4	5	6	7	″3x
´07x	8	9	⊍	⩀	□	▯	‖	⦀	
´10x	⊕	𝔄	𝔅	ℭ	𝔇	𝔈	𝔉	𝔊	″4x
´11x	ℌ	ℑ	𝔍	𝔎	𝔏	𝔐	𝔑	𝔒	
´12x	𝔓	𝔔	ℜ	𝔖	𝔗	𝔘	𝔙	𝔚	″5x
´13x	𝔛	𝔜	ℨ						
´14x		𝔞	𝔟	𝔠	𝔡	𝔢	𝔣	𝔤	″6x
´15x	𝔥	𝔦	𝔧	𝔨	𝔩	𝔪	𝔫	𝔬	
´16x	𝔭	𝔮	𝔯	𝔰	𝔱	𝔲	𝔳	𝔴	″7x
´17x	𝔵	𝔶	𝔷					!	
´20x	−	!	/	−̸		– –	〰	┈	″8x
´21x	←	⇐	⇚	←	↞	⊢	⊨	⫢	
´22x	↩	↼	↼	↽	⥪	⥫	⇇	⥢	″9x
´23x	⇆	⬱	⤚	⤜	⫰	⫱	⟜	●−	
´24x	→	⇒	⇛	→	↠	⊣	⫤	⫣	″Ax
´25x	↪	⇀	⇀	⇁	⥬	⥭	⇉	⥤	
´26x	⇄	⇶	⤙	⤛	⫯	⫰	⊸	−●	″Bx
´27x	⇠	⇠	⇠	⇠	⇢	⇢	⇢	⇢	
´30x		=	=	≡	∼	┈	┈	┈	″Cx
´31x	−	=	=	≡	∼	┈	┈	┈	
´32x	—	═	═	≣	∼	∼	┈	┈	″Dx
´33x		/	/	/	/	/	/	/	
´34x		≠	≠	≢	≁	≁	≠	≢	″Ex
´35x									
´36x		==	==	≡≡	∼∼	┈┈	┈┈	┈┈	″Fx
´37x									
	″8	″9	″A	″B	″C	″D	″E	″F	

Table 6 – Math Extension Primary Encoding (lower half)

	´0	*´1*	*´2*	*´3*	*´4*	*´5*	*´6*	*´7*	
´00x	$[$	$]$	$[\![$	$]\!]$	$($	$)$	$\{$	$\}$	″0x
´01x	$\lfloor$	$\rfloor$	$\lceil$	$\rceil$	$\langle$	$\rangle$	$/$	$\backslash$	
´02x	$\widehat{}$	╭	⏞	╮	-	⌐	$\uparrow$	$\Uparrow$	″1x
´03x	$\widetilde{}$	╰	⏟	╯	-	$\vert$	$\vert$	$\Vert$	
´04x		$\vert$	$\Vert$	⦀	$\surd$	$\surd$	$\downarrow$	$\Downarrow$	″2x
´05x	⎡	⎤	$[\![$	$]\!]$	⎛	⎞	⎧	⎫	
´06x	⎢	⎥	$\Vert$	$\Vert$	⎜	⎟	⎪	⎪	″3x
´07x	⎣	⎦	$[\![$	$]\!]$	⎝	⎠	⎨	⎬	
´10x	$\sum$	$\prod$	$\coprod$	$\odot$	$\oplus$	$\otimes$	⎩	⎭	″4x
´11x	$\displaystyle\sum$	$\displaystyle\prod$	$\displaystyle\coprod$	$\bigodot$	$\bigoplus$	$\bigotimes$			
´12x	$\int$	$\iint$	$\iiint$	$\oint$	∯	$\cup$	$\cap$	$\uplus$	″5x
´13x	$\displaystyle\int$	$\displaystyle\iint$	$\displaystyle\iiint$	$\displaystyle\oint$	∯	$\bigcup$	$\bigcap$	$\biguplus$	
´14x	$\vee$	$\wedge$	$\sqcup$	$\sqcap$	$\bigvee$	$\bigwedge$	$\bigsqcup$	⨅	″6x
´15x	$\widehat{}$	$\widehat{}$	$\widehat{}$	$\widehat{}$					
´16x	$\widetilde{}$	$\widetilde{}$	$\widetilde{}$	$\widetilde{}$					″7x
´17x	$\surd$	$\sqrt{}$	$\sqrt{}$	$\sqrt{}$				⚠	
	″8	″9	″A	″B	″C	″D	″E	″F	

Table 7 – Math Extension Primary Encoding (upper half)

	´0	´1	´2	´3	´4	´5	´6	´7	
´20x	$\big[$	$\big]$	$\big\llbracket$	$\big\rrbracket$	$\big($	$\big)$	$\big\{$	$\big\}$	″8x
´21x	$\Big[$	$\Big]$	$\Big\llbracket$	$\Big\rrbracket$	$\Big($	$\Big)$	$\Big\{$	$\Big\}$	
´22x	$\bigg[$	$\bigg]$	$\bigg\llbracket$	$\bigg\rrbracket$	$\bigg($	$\bigg)$	$\bigg\{$	$\bigg\}$	″9x
´23x	$\big\lfloor$	$\big\rfloor$	$\big\lceil$	$\big\rceil$	$\big\langle$	$\big\rangle$	$\big/$	$\big\backslash$	
´24x	$\Big\lfloor$	$\Big\rfloor$	$\Big\lceil$	$\Big\rceil$	$\Big\langle$	$\Big\rangle$	$\Big/$	$\Big\backslash$	″Ax
´25x	$\bigg\lfloor$	$\bigg\rfloor$	$\bigg\lceil$	$\bigg\rceil$	$\bigg\langle$	$\bigg\rangle$	$\bigg/$	$\bigg\backslash$	
	″8	″9	″A	″B	″C	″D	″E	″F	

Table 8 – Math Extension 1 Encoding