Technical Reports

(2nd) Proposed Update Unicode Technical Report #25

UNICODE SUPPORT FOR MATHEMATICS

Authors	Barbara Beeton (bnb@ams.org), Asmus Freytag (asmus@unicode.org), Murray Sargent III (murrays@microsoft.com)
Date	2007-2-1
This Version	http://www.unicode.org/reports/tr25/tr25-8.html
Previous Version	http://www.unicode.org/reports/tr25/tr25-6.html
Latest Version	http://www.unicode.org/reports/tr25
Revision	8

Summary

The Unicode Standard includes virtually all of the standard characters used in mathematics. This set supports a wide variety of math usage on computers, including in document presentation languages like TeX, in math markup languages like MathML and OpenMath, in internal representations of mathematics for applications like Mathematica, Maple, and MathCAD, in computer programs, and in plain text. This technical report describes the Unicode mathematics character groups and gives some of their imputed default math properties.

NOTE TO REVIEWERS:

Significant changes to the text are marked. Extensive copy editing was applied to this document compared to the latest published version, but most of those text changes have not been marked in order to keep the text readable.

A number of sections are marked *[proposed]*. The issues addressed in these sections have proposed solutions submitted separately to the UTC. The text in these proposed sections as written would apply if these proposals are adopted. The plan is to update these sections to match the outcome of the disposition of such proposals. The authors feel that capturing the ongoing work in these instances is beneficial to the user community in order to provide more meaningful reviews. Where pending character proposals are mentioned, code points are not given as they are tentative until characters are finally encoded.

Status

This is a draft document which may be updated, replaced, or superseded by other documents at any time. Publication does not imply endorsement by the Unicode Consortium. This is not a stable document; it is inappropriate to cite this document as other than a work in progress.

A Unicode Technical Report (UTR) contains informative material. Conformance to the Unicode Standard does not imply conformance to any UTR. Other specifications, however, are free to make

normative references to a UTR.

Please submit corrigenda and other comments with the online reporting form [Feedback]. Related information that is useful in understanding this document is found in the References. For the latest version of the Unicode Standard see [Unicode]. For a list of current Unicode Technical Reports see [Reports]. For more information about versions of the Unicode Standard, see [Versions].

Contents

1 Overview

All of science and technology uses formulas, equations, and mathematical notation as part of the language of the subject. This report presents a discussion of the mathematics character repertoire of the Unicode Standard [Unicode] as used for mathematics, but this discussion is intended apply to mathematical notation in general.

Mathematical documents using the Arabic script use additional conventions, in particular when typesetting mathematics from right to left. Such conventions are not documented here. This report also does not discuss mathematical symbols of purely historical or local interest, such as symbols found in ancient mathematical texts or digits used in script specific systems for writing numeric quantities.

As described in the Unicode Character Property Model [PropMod], each Unicode character has associated character properties. This report describes the properties relevant to the mathematics character repertoire, including a number of properties that are not yet part of the Unicode Standard, and details character classifications by usage and by typography. In addition, this report gives some implementation guidelines for input methods and use of Unicode math characters in programming languages.

Some of the text of the character block descriptions in the Unicode Standard was based on early drafts of this report; as a result there is significant overlap, although the focus of the presentation is different. As always, wherever there is a discrepancy, the text of the Standard has precedence.

The notational conventions follow the use in [Unicode]. Due to limitations of the plain HTML format of this report, examples of mathematical formulas are shown in larger size than would be typical for a mathematical paper, and their layout, spacing and vertical alignment are merely approximations of the correct appearance.

2 Mathematical Character Repertoire

The Unicode Standard provides a quite complete set of standard math characters to support publication of mathematics on and off the web. The early versions of Unicode, through version 3.0 already included over three hundred math–specific symbols. Unicode 3.1 introduced almost a thousand new alphanumeric symbols, and Unicode 3.2 introduced six hundred new characters for operators, arrows, and delimiters for a total of around 2000 mathematical symbols. The more limited additions to the repertoire in the versions since then have filled some gaps in coverage, in particular for mapping existing ISO entity sets for publishing [ISO9573].

The repertoire of mathematical characters in [Unicode] is the result of input from many sources, notably from the STIX Project (Scientific and Technical Information Exchange) [STIX], a collaborative project of scientific and technical publishers. The STIX collection includes, but is not limited to, symbols gleaned from mathematical publications by experts from the American Mathematical Society (AMS), and symbol sets provided by Elsevier Publishing and by the American Physical Society. This repertoire enables the display of virtually all standard mathematical symbols. Nevertheless no collection of mathematical symbols can ever be considered complete; mathematicians and other scientists are continually inventing new mathematical symbols, which will be considered for addition as they become widely accepted in the scientific communities.

Mathematical Markup Language (MathML™) [MathML], an XML application [XML], is a major beneficiary of the increased repertoire for mathematical symbols. The W3C Math Working Group, which developed MathML, lobbied in favor of the inclusion of the new characters. In addition, the new characters lend themselves to direct plain text encoding of mathematics for various purposes which can be much more

compact than MathML or T_EX, the typesetting language and program designed by Donald Knuth [TeX] (see *Section 4, Implementation Guidelines*).

2.1 Mathematical Alphanumeric Symbols Block

The Mathematical Alphanumeric Symbols block (U+1D400—U+1D7FF) contains a large collection of letterlike symbols for use in mathematical notation, typically for variables. The characters in this block are intended for use only in mathematical or technical notation; they are not intended for use in non-technical text. When used with markup languages, for example with MathML the characters are expected to be used directly, instead of indirectly via entity references or by composing them from base letters and style markup.

Words Used as Variables. In some specialties, whole words are used as variables, not just single letters. For these cases, style markup is preferred because the juxtaposition of variables generally implies multiplication, or some other composition, in ordinary mathematical notation, not word formation as in ordinary text. Markup not only provides the necessary scoping in these cases, it also allows the use of a more extended alphabet.

2.2 Mathematical Alphabets

Basic Set of Alphanumeric Characters. Mathematical notation uses a basic set of mathematical alphanumeric characters which consists of:

- the set of basic Latin digits (0 – 9) (U+0030..U+0039)
- the set of basic upper– and lowercase Latin letters (a – z, A – Z)
- the uppercase Greek letters A – Ω (U+0391..U+03A9), plus the nabla ∇ (U+2207), digamma (U+03DC), and the variant of theta Θ given by U+03F4
- the lowercase Greek letters α – ω (U+03B1..U+03C9), plus the partial differential sign ∂ (U+2202), digamma (U+03DD), and the six glyph variants of ϵ, θ, κ, ϕ, ρ, and π, given by U+03F5, U+03D1, U+03F0, U+03D5, U+03F1, and U+03D6.

For some characters in the basic set of Greek characters, two variants of the same character are included. This is because they can appear in the same mathematical document with different meanings, even though they would have the same meaning in Greek text.

Mathematical Accents. The diacritics, or accents, in mathematical text usually have special semantic significance different from that of changing the pronunciation of a letter, as is the case for text accents. Because the use of text accents such as the acute accent would interfere with common mathematical diacritics, only unaccented forms of the letters are used for mathematical notation. Examples of common mathematical diacritics that can be confused with text accents are the circumflex, macron, or the single or double dot above, the latter two of which are commonly used in physics to denote derivatives with respect to the time variable.

Mathematical symbols with diacritics are always represented by combining character sequences, except as required by normalization. See Unicode Standard Annex #15, "Unicode Normalization Forms" [Normalization] for more information. Note that normalization leaves all characters in the Mathematical Alphanumeric Symbols and Letterlike Symbols blocks unaffected. These blocks contain nearly all alphabetic characters used as math symbols.

Additional Characters. In addition to this basic set, mathematical notation also uses the bold upper- and lowercase digamma (U+1D7CA and U+1D7CB), and the four Hebrew-derived characters (U+2135..U+2138), for example in $\aleph_0$ for the first transfinite cardinal. Occasional uses of other alphabetic and numeric characters are known. Examples include U+0428 Ш CYRILLIC CAPITAL LETTER SHA, U+306E の HIRAGANA LETTER NO, the ideograph U+4E2D 中 and Eastern Arabic-Indic digits (U+06F0..U+06F9). However, unlike the characters in the mathematical alphabets, these characters are only used in a single, basic form.

Dotless Characters. In Unicode, the characters "i" and "j", including their variations in the mathematical alphabets have the Soft_Dotted property. Any conformant renderer will remove the dot when the character is followed by a nonspacing combining mark above. Therefore using an individual *mathematical italic i* or *j* with math accents would result in the intended display. However, in mathematical equations an entire sub-expression can be placed underneath a math accent, for example, when a 'wide hat' is placed on top of $i + j$, as in this example shown together with the corresponding [TeX] notation:

$$\widehat{i + j} = \hat{\imath} + \hat{\jmath}$$

$\widehat{\imath + \jmath} = \hat{\imath} + \hat{\jmath}.$

Whenever a mathematical accent applies to an entire subexpression, a renderer can no longer rely simply on the presence of an adjacent combining character to substitute the un-dotted glyph; whether the dots should be removed in such a situation is no longer predictable. In T$_E$X, this decision is left to the author, and some authors would want to use the dotted forms as in $\widehat{i + j}$.

In some documents *mathematical italic dotless i* or *j* are used explicitly without any combining marks, or even in contrast to the dotted versions. Therefore, the Unicode Standard provides the explicitly dotless characters U+1D6A4 MATHEMATICAL ITALIC DOTLESS I and U+1D6A5 MATHEMATICAL ITALIC DOTLESS J. They map to the ISOAMSO entities imath and jmath or the [TeX] macros \imath and \jmath which by default are always italic. Their appearance in the code charts is similar to the shapes documented in the ISO 9573-13 entity sets and used by T$_E$X. They do not form case pairs.

Where a math accent is immediately applied to these entities, as in $\hat{\imath } + \hat{\jmath}$, they could be mapped to *mathematical italic i* or *j* when converting to Unicode, but making general substitutions could result in an unintended appearance or a change to the document.

Semantic Distinctions. Mathematical notation requires a number of Latin and Greek alphabets that initially appear to be mere font variations of one another. For example, the letter H can appear as plain or upright (H), bold (**H**), italic (*H*), and script ($\mathcal{H}$). However, in any given document, these characters have distinct, and usually unrelated mathematical semantics. For example, a normal H represents a different variable from a bold **H**, *etc.* If these attributes are dropped in plain text, the distinctions are lost and the meaning of the text is altered. Without the distinctions, the well-known Hamiltonian formula:

$$\mathcal{H} = \int d\tau(\epsilon E^2 + \mu H^2),$$

turns into the *integral* equation in the variable H:

$$H = \int d\tau (\varepsilon E^2 + \mu H^2).$$

Mathematicians will object that a properly formatted integral equation requires all the letters in this example (except for the "d") to be in italics. However, because the distinction between $\mathcal{H}$ and H has been lost, they would recognize it as a fallback representation of an integral equation, and not as a fallback representation of the Hamiltonian. By encoding a separate set of alphabets, it is possible to preserve such distinctions in plain text.

Mathematical Alphabets. The alphanumeric symbols encountered in mathematics are given in the following table:

Table 2.1 Mathematical Alphabets

Math Style	Characters from Basic Set	Location
plain (upright, serifed)	Latin, Greek and digits	BMP
bold	Latin, Greek and digits	Plane 1
italic	Latin and Greek	Plane 1*
bold italic	Latin and Greek	Plane 1
script (calligraphic)	Latin	Plane 1*
bold script (calligraphic)	Latin	Plane 1
Fraktur	Latin	Plane 1*
bold Fraktur	Latin	Plane 1
double-struck	Latin and digits	Plane 1*
sans-serif	Latin and digits	Plane 1
sans-serif bold	Latin, Greek and digits	Plane 1
sans-serif italic	Latin	Plane 1
sans-serif bold italic	Latin and Greek	Plane 1
monospace	Latin and digits	Plane 1

* Some of these alphabets have characters in the BMP as noted in the following section.

The plain letters have been unified with the existing characters in the Basic Latin and Greek blocks. There are 24 double-struck, italic, Fraktur and script characters that already exist in the Letterlike Symbols block (U+2100—U+214F). These are explicitly unified with the characters in this block and corresponding holes have been left in the mathematical alphabets.

Compatibility Decompositions. All mathematical alphanumeric symbols have compatibility decompositions to the base Latin and Greek letters—folding away such distinctions, however, is usually not desirable as it loses the semantic distinctions for which these characters were encoded. See Unicode Standard Annex #15, *Unicode Normalization Forms* [Normalization] for more information.

Typical Uses. The following list catalogs examples of typical uses for some of these styles without intending to be exhaustive or exclusive.

- *lightface italic* -- variables
- *double-struck* -- sets
- *bold* -- vectors (more physics and applied areas, usually lowercase)

- *bold italic* -- matrices (uppercase)
- *lightface roman* -- operator names (sin, cos, etc.), some constants, units
- *lowercase Greek* -- angles
- *script (caps)* -- various operators, functions and transforms
- *sans-serif* -- dimensions of SI base quantities ([NISTGuide],   p.23; uncertain whether lightface or bold)
- *bold italic sans-serif* -- tensors ([NISTGuide], p.34, also [NISTStyle] style sheet)

2.33 Fonts Used for Mathematical Alphabets

Mathematicians place strict requirements on the *specific* fonts being used to represent mathematical variables. Readers of a mathematical text need to be able to distinguish single letter variables from each other, even when they do not appear in close proximity. They must be able to recognize the letter itself, whether it is part of the text or is a mathematical variable, and lastly which mathematical alphabet it is from.

Fraktur. The black letter style is often referred to as *Fraktur* or *Gothic* in various sources. Technically, Fraktur and Gothic typefaces are distinct designs from black letter, but any of several font styles similar in appearance to the forms shown in the charts can be used.

Math Italics. Mathematical variables are most commonly set in a form of italics, but not all italic fonts can be used successfully. In common text fonts, the *italic letter v* and *Greek letter nu* are not very distinct. A rounded *italic letter v* is therefore preferred in a mathematical font, as long as it is distinct from the *Greek upsilon*. There are other characters, which sometimes have similar shapes and require special attention to avoid ambiguity. Examples are shown in the table below.

italic a	a	α	alpha
italic v (pointed)	v	ν	nu
italic v (rounded)	υ	υ	upsilon
script X	$\mathscr{X}$	χ	chi
plain Y	Y	Υ	Upsilon

Theorems are commonly printed in a text italic font. A font intended for mathematical variables should support clear visual distinctions so that variables can be reliably separated from italic text in a theorem. Some languages have common single letter words (English *a*, Scandinavian *i*, etc.), which can otherwise be easily confused with common variables.

Hard-to-distinguish Letters. Not all sans-serif fonts allow an easy distinction between *lowercase l*, and *uppercase I* and not all monospaced (fixed width) fonts allow a distinction between the *letter* 1 and the *digit* 1. Such fonts are not usable for mathematics. In Fraktur, the letters I and J in particular must be made distinguishable. Overburdened Black Letter forms like I and J are inappropriate. Similarly, the *digit zero* must be distinct from the *uppercase letter O*, and the empty set ∅ must be distinct from the *letter o with stroke ('Ø')* for all mathematical alphanumeric sets. Some characters are so similar that even mathematical fonts do not attempt to provide distinguished glyphs for them. Their use is normally avoided in mathematical notation unless no confusion is possible in a given context, for example *uppercase A* and *uppercase Alpha* (A).

Font Support for Combining Diacritics. Mathematical equations require that characters be combined with diacritics (dots, tilde, circumflex, or arrows above are common), as well as followed or preceded by super- or subscripted letters or numbers. This requirement leads to designs for *italic* styles that are less inclined, and *script* styles that have smaller overhangs and less slant than equivalent styles commonly used for text such as wedding invitations.

Typestyle for Script Characters. In some instances, a deliberate unification with a non-mathematical symbol has been undertaken; for example, U+2133 $\mathcal{M}$ SCRIPT CAPITAL M is unified with the pre-1949 symbol for the German currency unit *Mark.* This unification restricts the range of glyphs that can be used for this character in the charts. Therefore the font used for the reference glyphs in the code charts uses a simplified 'English Script' style, as recommended by the American Mathematical Society. For consistency, other script characters in the Letterlike Symbols block are now shown in the same typestyle.

The two characters U+2113 ℓ SCRIPT SMALL L, and U+2118 $\wp$ SCRIPT CAPITAL P, are not regular script characters, despite their character names. The latter is the symbol for the *Weierstrass elliptic function*, a calligraphic letter shape based on the small p, and the former is derived from a special italic letter shape called an 'ell', and is unified with the common non-SI symbol for the liter [SI]. The characters U+1D4C1 MATHEMATICAL SCRIPTS SMALL L and U+1D4AB MATHEMATICAL SCRIPT CAPITAL P are the preferred characters for the script style.

Double-struck Characters. The double-struck glyphs shown in earlier editions of the standard attempted to match the design used for all the other Latin characters in the standard, which is based on Times. The current set of fonts for use in the character code charts was prepared after consultation with the American Mathematical Society and leading publishers of mathematics, and shows much simpler forms that are derived from the forms written on a blackboard. However, this font represents just one possible representation of double-struck characters; both serifed and non-serifed forms can be used in mathematical texts, and inline fonts are found in works published by certain publishers. Some fonts differ in which strokes of a glyph to double, for example the left or right leg of the *uppercase A*. There is no intention to support any of these stylistic preferences via character encoding, therefore only one set of double-struck mathematical alphanumeric symbols are encoded.

2.3.1 Representative Glyphs for Greek Phi

With Unicode 3.0 and the concurrent second edition of ISO/IEC 10646-1, the representative glyphs for U+03C6 GREEK LETTER SMALL PHI and U+03D5 GREEK PHI SYMBOL were exchanged. In ordinary Greek text, the character U+03C6 is used exclusively, although this character has considerable glyphic variation, sometimes represented with a glyph more like the representative glyph shown for U+03C6 (the "loopy" form) and less often with a glyph more like the representative glyph shown for U+03D5 (the "straight" form). See the Greek table in the character code charts [Charts].

For mathematical and technical use, the straight form of the *small phi* is an important symbol and needs to be consistently distinguishable from the loopy form. The straight form phi glyph is used as the representative glyph for the *phi symbol* at U+03D5 to satisfy this distinction.

The assignment of representative glyphs was reversed in versions of the Unicode Standard prior to Unicode 3.0. As a result, the character explicitly identified as the mathematical symbol did not have the straight form of the character that is the preferred glyph for that use. Furthermore, it made it unnecessarily difficult for general purpose fonts supporting ordinary Greek text to also add support for Greek letters used as mathematical symbols, because many of those fonts already used the loopy form

glyph for U+03C6, as preferred for Greek body text. To support the phi symbol as well, they would have had to disrupt glyph choices already optimized for Greek text.

When mapping symbol sets or SGML entities to the Unicode Standard, it is important to make sure that codes or entities, such as `phi1`, that require the straight form of the *phi symbol* be mapped to U+03D5 and not to U+03C6. Mapping to the latter should be reserved for codes or entities that represent the *small phi* as used in ordinary Greek text.

Fonts used primarily for Greek text may use either glyph form for U+03C6, but fonts that also intend to support technical use of the Greek letters should use the loopy form to ensure appropriate contrast with the straight form used for U+03D5.

2.3.2 Representative Glyphs for U+2278 and U+2279

In Unicode 3.2 the representative glyphs for U+2278 NEITHER LESS-THAN NOR GREATER-THAN and U+2279 NEITHER GREATER-THAN NOR LESS-THAN were changed from using a vertical cancellation to using a slanted cancellation to match the long standing canonical decompositions for these characters, which use U+0338 COMBINING LONG SOLIDUS OVERLAY. Irrespective of this change to the representative glyphs, the symmetric forms using the vertical stroke remain acceptable glyph variants. Using U+2276 ≶ or U+2277 ≷ followed by U+20D2 COMBINING LONG VERTICAL LINE OVERLAY represents these upright variants explicitly.

Except for those fonts created with the intention to add support for *both* forms (via combination of U+2276 ≶ or U+2277 ≷ with U+20D2 for the upright forms) there is no need to revise the glyphs for U+2278 and U+2279: the glyphic range implied by using these character codes encompasses both shapes.

2.4 Locating Mathematical Characters

Mathematical characters can be located by looking in the code charts [Charts] at the blocks listed below or by checking the Unicode MATH property, which is assigned to characters that naturally appear in mathematical contexts (see *Section 3, Mathematical Character Properties*). In the text of this report, all block names are linked to their corresponding online code chart. Mathematical characters can be found in the following blocks:

Table 2.2 Locations of Mathematical Characters

Block Name	Range	Character Types
Basic Latin	U+0021–U+007E	Variables, operators, digits*
Greek	U+0370–U+03FF	Variables*
General Punctuation	U+2000–U+206F	Spaces, Invisible operators*
Letterlike Symbols	U+2100–U+214F	Variables*
Arrows	U+2190–U+21FF	Arrows, arrow-like operators
Mathematical Operators	U+2200–U+22FF	Operators
Miscellaneous Technical Symbols	U+2300–U+23FF	Braces, operators*
Geometrical Shapes	U+25A0–U+25FF	Symbols
Misc. Mathematical Symbols-A	U+27C0–U+27EF	Symbols and operators
Supplemental Arrows-A	U+27F0–U+27FF	Arrows, arrow-like operators
Supplemental Arrows-B	U+2900–U+297F	Arrows, arrow-like

		operators
Misc. Mathematical Symbols-B	U+2980–U+29FF	Braces, symbols
Suppl. Mathematical Operators	U+2A00–U+2AFF	Operators
Misc. Symbols and Arrows	U+2B00–U+2BFF	Arrows, operators or symbols
Mathematical Alphanumeric Symbols	U+1D400–U+1D7FF	Variables and digits
Other blocks	…	Characters for occasional use

This block contains non-mathematical characters as well.

2.5 Duplicated Characters

Some Greek letters are encoded elsewhere as technical symbols. These include U+00B5 µ MICRO SIGN, U+2126 Ω OHM SIGN, and several characters among the APL functional symbols in the Miscellaneous Technical block. U+03A9 Ω GREEK LETTER CAPITAL OMEGA is the canonical equivalent of U+2126 Ω and its use is preferred. *Micro sign* is included in several parts of ISO/IEC 8859, and therefore supported in many legacy environments where U+03BC µ GREEK LETTER SMALL MU is not available. Implementations therefore need to be able to recognize the micro sign, even though *mu* is the preferred character in a Unicode context.

Latin letters duplicated include U+212A K KELVIN SIGN and U+212B Å ANGSTROM SIGN. As in the case of the *ohm sign*, the corresponding regular Latin letters are canonical equivalents, therefore their use is preferred.

The *left* and *right angle brackets* at U+2329 and U+232A have long been canonically equivalent with the CJK punctuation characters at U+3008 〈 and U+3009 〉. Canonical equivalence implies that the use of the latter code points is preferred and that not only 3008 and 3009 but also the characters 2329 and 232A are 'wide' characters. See Unicode Standard Annex #11, *East Asian Width* [EAW]. Unicode 3.2 added two new *mathematical angle bracket* characters (U+27E8 and U+27E9) that are unequivocally intended for mathematical use.

2.6 Accented Characters

Mathematical characters are often enhanced via use of combining marks in the ranges U+0300..U+036F and the combining marks for symbols in the range U+20D0..U+20FF. These characters follow the base characters as in non-mathematical Unicode text. This section discusses these characters and preferred ways of representing accented characters in mathematical expressions. If a span of characters is enhanced by a combining mark, for example, a tilde over AB, typically some kind of higher-level markup is needed as is done in [MathML]. Unicode does include some combining marks that are designed to be used for pairs of characters, for example, U+0360..U+0362. However, their use for mathematical text is not encouraged.

For some mathematical characters, such as many negated relations, there are multiple ways of expressing the character: as precomposed or as a sequence of base character and combining mark (see also *Section 2.17, Negations*). Having only a single way to represent any given character would simplify recognizing the character in searches and other manipulations. Selecting a unique representation among multiple equivalent representations is called *normalization*. Unicode Standard Annex #15 *Unicode Normalization Forms* [Normalization] discusses the subject in detail; however, due to requirements of non-mathematical software, not all the normalization forms presented there are ideal from the perspective of mathematics.

Ideally, one always uses the shortest form of a math operator symbol wherever possible. So U+2260 ≠ should be used for the *not equal sign* instead of the combining sequence <003D, 0338>. If a negated operator lacking a precomposed form is needed, U+0338 COMBINING LONG SOLIDUS OVERLAY or U+20D2 COMBINING VERTICAL LONG OVERLAY can be used to indicate negation. This approach concurs with Normalization Form C (NFC), which is also the preferred normalization form for use on the web.

On the other hand, for accented *alphabetic* characters used as variables, ideally only decomposed sequences are used, because mathematics uses a multitude of combining marks that greatly exceeds the predefined composed characters in Unicode. Accordingly, it is better to have the math display facility handle all of these cases uniformly to give a consistent look between characters that happen to have a fully composed Unicode character and those that do not. The combining character sequences also typically have semantics as a group, so it is useful to be able to manipulate and search for them individually without the need for special tables to decompose characters for this purpose. Since there are no precomposed math alphanumeric symbols, this approach concurs with Normalization Form C, *except* for the upright alphabetic characters (ASCII letters).

To facilitate interchange on the web, accented characters should conform to NFC when interchanged. However, to achieve consistent results, a mathematical display system should transiently decompose any precomposed upright letters when used in mathematical expressions, and should use a single algorithm to place embellishments.

Normalization Form D (NFD) uses the opposite approach from NFC. It works naturally for mathematical use of alphabetic characters, but does not use the shortest encoding of math operator symbols, making it less attractive. The other two normalization forms NFKC and NFKD remove the distinction between math alphanumeric alphabets, mapping all of them to plain ASCII or Greek characters. As a result they would destroy the semantics of many mathematical expressions, should never be used with mathematical texts.

2.7 Operators

The Mathematical Operators (U+2200—U+22FF) and Supplemental Mathematical Operators (U+2A00—U+2AFF) blocks contain many mathematical operators, relations, geometric symbols and other symbols with special usages confined largely to mathematical contexts. In addition to the characters in these blocks, mathematical operators are also found in the Basic Latin (ASCII) and Latin-1 Supplement Blocks. A few of the symbols from the Miscellaneous Technical block and characters from General Punctuation are also used in mathematical notation. The allocation of any operator to a particular block is rarely significant.

Semantics. Mathematical operators often have more than one meaning in different subdisciplines or different contexts. For example, the "+" symbol normally denotes addition in a mathematical context, but might refer to concatenation in a computer science context dealing with strings, or incrementation, or have any number of other functions in given contexts. Therefore the Unicode Standard only encodes a single character for a single symbolic form. There are numerous other instances in which several semantic values can be attributed to the same Unicode value. For example, U+2218 ∘ RING OPERATOR may be the equivalent of *white small circle* or *composite function* or *apl jot*. The Unicode Standard does not attempt to distinguish all possible semantic values that may be applied to mathematical operators or relational symbols. It is up to the application or user to distinguish such meanings according to the appropriate context. Where information is available about the usage (or usages) of particular symbols, it has been indicated in the character annotations in the code charts printed in [Unicode] and in the online code charts [Charts].

Similar Glyphs. The Standard includes many characters that appear to be quite similar to one another, but that may convey different meaning in a given context. On the other hand, mathematical operators, and especially relation symbols, may appear in various standards, handbooks, and fonts with a large number of purely graphical variants. Where variants were recognizable as such from the sources, they were not encoded separately.

For relation symbols, the choice of a vertical or forward-slanting stroke typically seems to be an aesthetic one, but both slants might appear in a given context. However, a back-slanted stroke almost always has a distinct meaning compared to the forward-slanted stroke. See *Section 2.18, Variation Selector* for more information on some particular variants.

Unifications. Mathematical operators such as *implies* and *if and only if* have been unified with the corresponding arrows (U+21D2 ⇒ RIGHTWARDS DOUBLE ARROW and U+2194 ↔ LEFT RIGHT ARROW, respectively) in the Arrows block.

The operator U+2208 ∈ ELEMENT OF is occasionally rendered with a taller shape than shown in the code charts. Mathematical handbooks and standards treat these characters as variants of the same glyph. U+220A ϵ SMALL ELEMENT OF is a distinctively small version of the *element of* that originates in mathematical pi fonts.

The operators U+226B ≫ MUCH GREATER-THAN and U+226A ≪ MUCH LESS-THAN are sometimes rendered in a nested shape, but the Unicode Standard provides a single encoding for each operator.

A large class of unifications applies to variants of relation symbols involving equality, similarity, and/or negation. Variants involving one- or two-barred *equal signs*, one- or two-*tilde similarity signs*, and vertical or slanted *negation slashes* and *negation slashes* of different lengths are not separately encoded. Thus, for example, U+2288 ⊈ NEITHER A SUBSET OF NOR EQUAL TO, is the archetype for at least six different glyph variants noted in various collections.

In a few exceptional instances, essentially stylistic variants are separately encoded because the need for roundtrip character mapping to other standards that distinguish the two forms. Examples include U+2265 ≥ GREATER-THAN OR EQUAL TO, which is distinguished from U+2267 ≧ GREATER-THAN OVER EQUAL TO; the same distinction applies to U+2264 ≤ LESS-THAN OR EQUAL TO and U+2266 ≦ LESS-THAN OVER EQUAL TO.

Greek-Derived Operators. Several mathematical operators derived from Greek characters have been given separate encodings because they are used differently than the corresponding letters. These operators may occasionally occur in context with Greek-letter variables. They include U+2206 ∆ INCREMENT, U+220F ∏ N-ARY PRODUCT, and U+2211 ∑ N-ARY SUMMATION. The latter two are large operators that take limits. Some typographical aspects of operators are discussed in *Section 3.2, Classification by Typographical Behavior*. For example, the n-ary operators are distinguished from letter variables by their larger size and the fact that they take limit expressions.

Minus sign. U+2212 − MINUS SIGN is the preferred representation of the unary and binary minus sign rather than the ASCII-derived U+002D - HYPHEN-MINUS, because *minus sign* is unambiguous and because it is rendered with a more desirable length, usually longer than a *hyphen*.

Miscellaneous Symbols. The symbol U+2205 ∅ EMPTY SET is distinct from the letters U+00D8 Ø and U+00F8 ø, even though historically derived from the letter forms. A widespread alternate symbol for the empty set is a slashed *digit zero*. This can be encoded as U+0030 DIGIT ZERO followed by U+0338 COMBINING LONG SOLIDUS OVERLAY.

The range from U+22EE ⋮ to U+22F1 ⋱. contains a set of ellipses used in matrix notation.

U+2023 ‣ TRIANGULAR BULLET and U+25B8 ▸ BLACK RIGHT-POINTING SMALL TRIANGLE are not intended to be distinct in appearance. For historical reasons these two are encoded separately and not made canonical equivalents of each other. U+25B8 ▸ is the preferred character.

2.8 Superscripts and Subscripts

The Superscripts and Subscripts block U+2070.. U+209F together with U+00B2 ^{2}, U+00B3 ^{3}, and U+00B9 ^{1} contain a collection superscript and subscript digits and punctuation that can be useful in mathematics. If they are used, it is recommended that they be displayed with the same font size as other subscripts and superscripts at the corresponding nested script level. For example, a^2 and `a<super>2</super>` should be displayed the same. However, these subscript/superscript characters are not used in MathML or T$_E$X and their use with XML documents for mathematical use is discouraged, see Unicode Technical Report #20, *Unicode in XML and other Markup Languages* [UXML]. Editors for these formats may offer facilities to convert these characters to regular characters plus markup.

Parsing of Superscript and Subscript Digits. Unlike regular digits the superscript and subscript digits have not been given the General Category property of Decimal_Digit (Nd). This prevents expressions like 2^3 from being interpreted as 23 by simplistic numeric parsers. More sophisticated numeric parsers, such as general mathematical expression parsers, can nevertheless choose to identify these compatibility superscript and subscript characters as digits and interpret them appropriately within their own scope.

2.9 Arrows

Arrows are used for a variety of purposes in mathematics and elsewhere, such as to imply directional relation, to show logical derivation or implication, and to represent the cursor control keys. Accordingly Unicode includes a fairly extensive set of arrows. (U+2190..U+21FF, U+27F0..U+27FF, U+2900..U+297F), many of which appear in mathematics. It does not attempt to encode every possible stylistic variant of arrows separately, especially where their use is mainly decorative. For most arrow variants, the Unicode Standard provides encodings in the two horizontal directions, often in the four cardinal directions. For the single and double arrows, the Unicode Standard provides encodings in eight directions.

Unifications. Arrows expressing mathematical relations have been encoded in the Arrows block as well as in Supplemental Arrows-A and Supplemental Arrows-B. An example is U+21D2 ⇒ RIGHTWARDS DOUBLE ARROW, which may be used to denote *implies*. Where available, such usage information is indicated in the annotations to individual characters in the Unicode Standard 5.0 [U5.0], Chapter 17, *Code Charts*, and in the online code charts [Charts].

Long Arrows. The long arrows encoded in the range U+27F5..U+27FF map to standard SGML entity sets supported by MathML. Long arrows represent distinct semantics from their short counterparts, rather than mere stylistic glyph differences. For example, the shorter forms of arrows are often used in connection with limits, whereas the longer ones are associated with mappings. The use of the long arrows is so common that they were assigned entity names in the ISOAMSA entity set, one of the suite of mathematical symbol entity sets covered by the Unicode Standard.

2.10 Delimiters

The mathematical white square brackets, angle brackets, and double angle brackets encoded at U+27E6..U+27EB are intended for ordinary use of these particular bracket types. They are unambiguously narrow, for use in mathematical and scientific notation, and should be distinguished from the corresponding wide forms of white square brackets, angle brackets, and double angle brackets used in CJK typography. (See the CJK Symbols and Punctuation block.)

However, the set of lenticular and tortoise-shell brackets in the CJK Punctuation block have not been duplicated because mathematical use has not yet been demonstrated. Fonts containing 'wide glyphs' for these characters that include white space padding, are unsuitable for mathematical or other non-CJK use.

Deprecated Delimiters. The angle brackets formerly aliased as "bra" and "ket", U+2329 〈 LEFT-POINTING ANGLE BRACKET and U+232A 〉 RIGHT-POINTING ANGLE BRACKET, are now deprecated for use with mathematics because their canonical equivalence to CJK angle brackets is likely to result in unintended spacing problems when used in mathematical formulae.

Horizontal Delimiters. Delimiters are often used horizontally, where they expand to the width of the expression they encompass, as in this example from [TeX].

```
$$\overbrace{x+\cdots+x}^{k\rm\;times}$$

$$\underbrace{x+y+z}_{>\,0}.$$
```

$$\overbrace{x + \cdots + x}^{k\text{ times}}$$
$$\underbrace{x + y + z}_{> 0}.$$

By providing character codes for these delimiters, mathematical layout systems can be designed so that both regular and horizontal delimiters are encoded as characters, with markup designating the scope where necessary. When the horizontal mathematical brackets are used, all other letters, symbols and digits remain upright as illustrated in the example above. Table 2.3 lists the Unicode characters for horizontal delimiters.

Table 2.3 : Horizontal Delimiters

Code	Description
23B4	TOP SQUARE BRACKET
23B5	BOTTOM SQUARE BRACKET
23DC	TOP PARENTHESIS
23DD	BOTTOM PARENTHESIS
23DE	TOP CURLY BRACKET
23DF	BOTTOM CURLY BRACKET
23E0	TOP TORTOISE SHELL BRACKET

Use of horizontal delimiters is different from horizontal display of delimiters in vertical layout of East Asian text, where ideographic characters remain upright, but non-ideographic characters (letters, digits) are rotated 90°. For example, the parentheses in the vertical text in the figure to the right have very different rendering from the under/overbrace examples above.

The CJK Compatibility Forms U+FE35 ︵ through U+FE39 ︹ have shapes that are

独立
護士

superficially similar to the horizontal delimiters, but these characters are not mathematical and have quite different rendering requirements. They are encoded for compatibility with character sets that use explicit character codes for the vertical glyph variants of punctuation characters. Like other CJK punctuation, CJK Compatibility Forms have the [EAW] property of W (wide) and are typically implemented in one half of an EM square, with the other half empty. Layout algorithms using these characters predict the empty half cell based on the character code, and reduce intercharacter spacing accordingly in some circumstances.

2.11 Geometrical Shapes

The basic geometric shapes (circle, square, triangle, diamond, and lozenge) are used for a variety of purposes in mathematical texts. Because their shapes are distinct and they are easily available in multiple sizes from a variety of widely available fonts, they are also often used in an ad-hoc manner. In Unicode they are encoded in the Geometrical Shapes, Miscellaneous Technical, Block Elements, Miscellaneous Symbols and Miscellaneous Symbols and Arrows blocks as shown in Table 2.4.

Ideal Sizes. Mathematical usage requires at least four distinct sizes of simple shapes, and sometimes more. The size gradation must allow each size to be recognized, even when it occurs in isolation. In other words, shapes of the same size should ideally have roughly the same visual "impact" as opposed to same nominal height or width. The shapes shown here for a given size all have the same area.

For mathematical usage simple shapes ideally share a common center. The following diagram shows the ideal size relationship across shapes of the same nominal size.
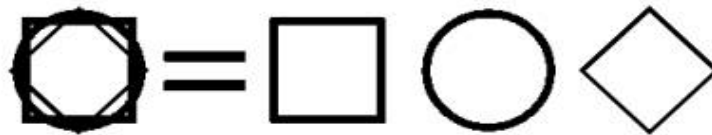
The precise sizes and shapes chosen, however, are a matter for the font designer. Note that neither the current set of representative glyphs in the standard nor the glyphs from many commonly available non-mathematical fonts achieve the ideals set forth here.

> *Note to reviewers:* In a previous review cycle, a reanalysis of this material has been proposed in document L2/06-034 (accessible to Unicode members). This has been reviewed by the authors in consultations with other mathematical experts. The conclusion is to request additions to the repertoire. Table 2-4 has been updated accordingly, with the proposed additions to the repertoire indicated as shapes, but with dummy code points.

Suggested Sizes [proposed]. The sizes of existing characters and their names as shown in the code charts are not always consistent. The suggested sizes here correspond to a geometric progression where for each size all characters have the same visual impact. Shapes for which only one of the columns with a "default" size exists can be implemented either as regular or medium size. The former is shown here, the latter may be more suitable for mathematical work. Table 2.4 summarizes the available sizes for a given symbol.

Table 2.4 Sizes of Simple Shapes [proposed]

Shape	tiny	very small	small (Bullet)	medium small	medium (default1)	regular (default2)	large

triangle left				◄ 25C2	◁ 25C3				◄ 25C0	◁ 25C1	
triangle right				► 25B8 2023	▷ 25B9				► 25B6	▷ 25B7	
triangle up				▲ 25B4	△ 25B5				▲ 25B2	△ 25B3	
triangle down				▼ 25BE	▽ 25BF				▼ 25BC	▽ 25BD	
square	▪ 2Bxx	▫ 2Bxx	■ 25AA	▫ 25AB	☐ 25FD	■ 25FE	■ 25FC	☐ 25FB	■ 25A0	☐ 25A1	■ 2Bxx / ☐ 2Bxx
diamond	◆ 2Bxx	◇ 22C4	◆ 2Bxx	◇ 2Bxx					◆ 25C6	◇ 25C7	
lozenge	◆ 2Bxx	◇ 2Bxx	◆ 2Bxx	◇ 2Bxx					◆ 29EB	◇ 25CA	
pentagon									⬟ 2Bxx	⬠ 2B20	
pentagon right									⬟ 2Bxx	⬠ 2Bxx	
hexagon horizontal									⬣ 2B23	⬡ 2394	
hexagon vertical									⬢ 2B22	⬡ 2B21	
arabic star				★ 066D	☆ 2Bxx	★ 22C6	☆ 2Bxx	★ 2605	☆ 2606		
ellipse horizontal									⬬ 2Bxx	⬭ 2Bxx	
ellipse vertical									⬮ 2Bxx	⬯ 2Bxx	
circle	· 22C5	• 2219 00B7	∘ 2218	• 2022	◦ 25E6	● 2981	○ 26AC	● 26AB	○ 26AA	● 25CF / ○ 25CB	● 2Bxx / ○ 25EF
circled circles	⊙ 2299	☉ 2609		◎ 233E							
circled circles	⨀ 2A00	⦿ 29BF	⊚ 229A	⦾ 29BE	⦿ 25C9	◎ 25CE					

Most simple geometrical shapes exist in both black and outline (white) form in a single default size. The default size as shown in the code charts would be in the column marked "regular", while for many font implementations, a size corresponding to the column marked "medium" is chosen. As it is difficult to distinguish higher-order polygons at smaller sizes, size distinctions for these shapes are less useful for notational purposes. Triangles exist in two sizes, a default size and a small, bullet size. Lozenges and diamonds exist in a default size, and interim size and a bullet size. Squares and circles exist in black and white in all sizes from very small to large. There is also a tiny circle, essentially a centered dot. At the tiny size, distinction between different shapes, or black and outline forms, becomes

impossible.

Arrangement in Code Space. For circles in particular, but also for lozenges, diamonds and stars, the white and black forms are not encoded under matching names or close together. The series of circled circles is also distributed across the Unicode code space.

Sizes of Derived Shapes. Circled and squared operators and similar derived shapes are more constrained in their usage than "plain" geometric shapes. They tend to occur in two generic sizes based on function: a smaller size for binary operators and large size for n–ary operators. Other than circled circles, they are not shown here. Circled circles come in two series, based on the size of the enclosing circle.

Orientation. Some geometric shapes can exist in more than one orientation. For triangles, the Unicode Standard encodes the four principal directions. Ovals, pentagons and hexagons exist in two orientations; U+2394 ⎔ SOFTFWARE FUNCTION SYMBOL can be used as a horizontal white hexagon. The choice of right–pointing pentagon is based on its use as an avatar of the unit pentagon on the complex plane. Generic use in geometry would use the upright orientation.

Positioning. For a mathematical font, the centerline should go through the middle of a parenthesis, which should go from bottom of descender to top of ascender. This is the same level as the minus or the middle of the plus and equal signs. For correct positioning, the glyph will descend below the baseline for the larger sizes of the basic shapes as in the following schematic diagram:
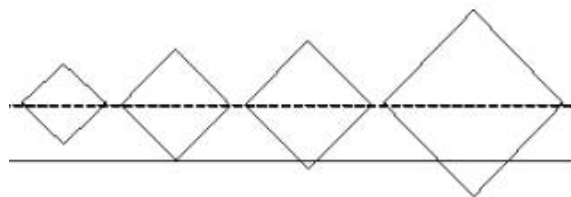
The standard triangles used for mathematics are also center aligned. This differs from the positioning for the representative glyphs shown in the charts, which are often based on existing non–mathematical fonts. Therefore, mathematical fonts may need to deviate in positioning of these triangles.

2.12 Other Symbols

Other symbols used in mathematics are contained in the Miscellaneous Technical block (U+2300—U+23FF), the Geometric Shapes block (U+25A0—U+25FF), the Miscellaneous Symbols block (U+2600—U+267F), and the General Punctuation block (U+2000—U+206F).

Generally any easily recognized and distinct symbol is fair game for mathematicians faced with the need of creating notations for new fields of mathematics. For example, the card suits, U+2665 ♥ BLACK HEART SUIT, U+2660 ♠ BLACK SPADE SUIT, *etc.* can be found as operators and as subscripts.

2.13 Symbol Pieces

The characters from the Miscellaneous Technical block in the range U+239B—U+23B3, plus U+23B7, comprise a set of bracket and other symbol fragments for use in mathematical typesetting. These pieces originated in older font standards, but have been used in past mathematical processing as characters in their own right to assemble extra–tall glyphs for enclosing multi–line mathematical

formulae. Mathematical fences are ordinarily sized to the content that they enclose. However, in creating a large fence, the glyph is not scaled proportionally; in particular the displayed stem weights must remain compatible with the accompanying smaller characters. Thus, simple scaling of font outlines cannot be used to create tall brackets. Instead, a common technique is to build up the symbol from pieces. In particular, the characters U+239B LEFT PARENTHESIS UPPER HOOK through U+23B3 SUMMATION BOTTOM represent a set of glyph pieces for building up large versions of the fences (,), [,], {, and }, and of the large operators Σ and ∫. These brace and operator pieces are compatibility characters. They should not be used in stored mathematical text, but are often used in the data stream created by display and print drivers.

Table 2.5 shows which pieces are intended to be used together to create specific symbols.

Table 2.5 Use of Symbol Pieces

	2-row	3-row	5-row
Summation	23B2, 23B3		
Integral	2320, 2321	2320, 23AE, 2321	2320, 3×23AE, 2321
Left Parenthesis	239B, 239D	239B, 239C, 239D	239B, 3×239C, 239D
Right Parenthesis	239E, 23A0	239E, 239F, 23A0	239E, 3×239F, 23A0
Left Bracket	23A1, 23A3	23A1, 23A2, 23A4	23A1, 3×23A2, 23A3
Right Bracket	23A4, 23A6	23A4, 23A5, 23A6	23A4, 3×23A5, 23A6
Left Brace	23B0, 23B1	23A7, 23A8, 23A9	23A7, 23AA, 23A8, 23AA, 23A9
Right Brace	23B1, 23B0	23AB, 23AC, 23AD	23AB, 23AA, 23AC, 23AA, 23AD

For example, an instance of U+239B can be positioned relative to instances of U+239C and U+239D to form an extra-tall (three or more line) *flattened left parenthesis*. The center sections are meant to be used only with the top and bottom pieces encoded adjacent to them, since the segments are usually graphically constructed within the fonts so that they match perfectly when positioned at the same *x* coordinates.

2.14 Invisible Operators

In mathematics some operators or punctuation are often implied, but not displayed. This poses few problems to the human reader, as the meaning is usually clear from context. However, machine interpretation of mathematical expressions may need the intent be made more explicit. To support this without altering the appearance of the equation when displayed, the Unicode Standard provides several invisible operators that can be used to unambiguously denote the intent whenever an operator is implied, or more importantly when more than one operator could be implied. Use of invisible operators is optional and is not required for intended for interchange with math-aware programs.

Invisible Separator. U+2063 INVISIBLE SEPARATOR or *invisible comma* is intended for use in index expressions and other mathematical notation where two adjacent variables form a list and are not implicitly multiplied. In mathematical notation, commas are not always explicitly present, but need to be indicated for symbolic calculation software to help it disambiguate a sequence from a multiplication. For example, the double $_{ij}$ subscript in the variable a_{ij} means $a_{i,j}$ — that is, the *i* and *j* are separate indices and not a single variable with the name *ij* or even the product of *i* and *j*. Accordingly to represent the implied list separation in the subscript $_{ij}$ one can insert a non-displaying *invisible separator* between the *i* and the *j*. In addition, use of the invisible comma would hint to a math layout program to set a small space between the variables.

Invisible Multiplication. Similarly, an expression like mc^2 implies that the mass m multiplies the square of the speed c. To unambiguously represent the implied multiplication in mc^2, one inserts a non-displaying U+2062 INVISIBLE TIMES between the m and the c. Another example is the expression $f^{ij}(\cos(ab))$, which means the same as $f^{ij}(\cos(a \times b))$, where $\times$ is used here to represents *multiplication*, not the *cross product.* Note that the spacing between characters may also depend on whether the adjacent variables are part of a list or are to be concatenated, that is, multiplied.

Invisible Function Application. U+2061 FUNCTION APPLICATION is used for an implied function dependence as in $f(x+y)$. To indicate that this is the function f of the quantity $x + y$ and not the expression $fx + fy$, one can insert the non-displaying *function application symbol* between the f and the left parenthesis.

Invisible Plus [proposed]. The final member of this set of invisible operators denoting the implied intent of juxtaposition in uses where it is not possible to rely on a human reader to disambiguate is a [proposed] INVISIBLE PLUS OPERATOR character to be able to unambiguously represent expressions like $3\frac{1}{2}$, which occur frequently in school or engineering texts. Not having an operator at all would imply multiplication as in the example

$$3\ \frac{abc}{d}$$

where the 3 represents a factor multiplying the following fraction.

2.15 Fraction Slash

U+2044 ⁄ FRACTION SLASH is used to build up simple fractions in running text. It applies to the immediately adjacent sequences of decimal digits, that is characters with the General Category=Nd. In general mathematical use a more general method for layout of fractions is needed, however parsers of mathematical texts should be prepared to handle *fraction slash* when it is received from other sources.

2.16 Other Characters

All remaining Unicode characters may appear in mathematical expressions, typically in spelled-out names for variables in fractions or simple formulae, but they most commonly appear in ordinary text. An English example is the equation

distance = rate $\times$ time,

which uses ordinary ASCII letters to aid in recognizing sequences of letters as words instead of products of individual symbols. Such usage corresponds to identifiers as discussed elsewhere in this report.

2.17 Negations

Many negated forms, particularly of relations, can be encoded by using the base symbol, together with a combining overlay. Occasionally, both a vertical and a slanted negation are used; which one is often a matter of style. Sometimes the negation is only indicated for part of a symbol. In these cases, the negated relations are encoded directly, and variants can be accessed via the *variation selector* method described in the next section.

Table 2.6 lists the currently encoded negated mathematical relations for which a variant can be realized via composition, by using U+20D2 COMBINING LONG VERTICAL LINE OVERLAY together with a base character. In the table, the part of the description in SMALL CAPS is the character name of the corresponding standard character; the part in lowercase indicates the variation in appearance.

Table 2.6 Negated Relations Using Vertical Line Overlay

Std Symbol		Alternate Symbol		Description of alternate symbol
∉	2209	∈	2208,20D2	NOT AN ELEMENT OF with vertical stroke
∌	220C	∋	220B,20D2	DOES NOT CONTAIN AS MEMBER with vertical stroke
≁	2241	∼	223C,20D2	NOT TILDE with vertical stroke
≄	2244	≃	2243,20D2	NOT ASYMPTOTICALLY EQUAL TO with vertical stroke
≇	2247	≅	2245,20D2	NEITHER APPROXIMATELY NOR ACTUALLY EQUAL TO with vertical stroke
≉	2249	≈	2248,20D2	NOT ALMOST EQUAL TO with vertical stroke
≠	2260	=	003D,20D2	NOT EQUAL TO with vertical stroke
≢	2262	≡	2261,20D2	NOT IDENTICAL TO with vertical stroke
≭	226D	≍	224D,20D2	NOT EQUIVALENT TO with vertical stroke
≮	226E	<	003C,20D2	NOT LESS-THAN with vertical stroke
≯	226F	>	003E,20D2	NOT GREATER-THAN with vertical stroke
≰	2270	≤	2264,20D2	NEITHER LESS-THAN NOR EQUAL TO with vertical stroke
≱	2271	≥	2265,20D2	NEITHER GREATER-THAN NOR EQUAL TO with vertical stroke
≸	2278	≶	2276, 20D2	NEITHER LESS-THAN NOR GREATER-THAN with vertical stroke (*)
≹	2279	≷	2277, 20D2	NEITHER GREATER-THAN NOR LESS-THAN with vertical stroke (*)
⊀	2280	≺	227A,20D2	DOES NOT PRECEDE with vertical stroke
⊁	2281	≻	227B,20D2	DOES NOT SUCCEED with vertical stroke
⊄	2284	⊂	2282,20D2	NOT A SUBSET OF with vertical stroke
⊅	2285	⊃	2283,20D2	NOT A SUPERSET OF with vertical stroke
⊈	2288	⊆	2286,20D2	NEITHER A SUBSET OF NOR EQUAL TO with vertical stroke
⊉	2289	⊇	2287,20D2	NEITHER A SUPERSET OF NOR EQUAL TO with vertical stroke
⋠	22E0	≼	227C,20D2	DOES NOT PRECEDE OR EQUAL with vertical stroke
⋡	22E1	≽	227D,20D2	DOES NOT SUCCEED OR EQUAL with vertical stroke

* The representative glyphs shown in the code charts [Charts] were revised in Unicode 4.0 [U4.0]to show the slanted forms – this matches their existing decomposition using U+0338 COMBINING LONG SOLIDUS OVERLAY (see *Section 2.32, Representative Glyphs for U+2278 ≸ and U+2279* for more information).

Note that the use of a base character together with the *slanted* U+0338 COMBINING LONG SOLIDUS OVERLAY is equivalent to the use of the precomposed negation (see also the discussion in *Section 2.6, Accented Characters*). For those symbols for which only a partial vertical stroke is used, use of U+20D2 would not give the intended result; U+FE00 VARIATION SELECTOR-1 is used instead, as described in *Section 2.18, Variation Selector*.

Table 2.7 lists some of the negated forms of mathematical relations that can *only* be encoded by using either U+0338 COMBINING LONG SOLIDUS OVERLAY or U+20D2 COMBINING LONG VERTICAL LINE OVERLAY. (For issues with using 0338 in MathML, see *Section 3.2.7, Combining Marks*. Depending on the overlay used, the negation has a diagonal or vertical stroke. The part of the description that is in SMALL CAPS reflects the Unicode character name of the non-negated symbol. Because these are not glyph variants of existing characters, the word "negated" is used instead of "NOT" as in the list above, to indicate that the negation is expressed by the combining character sequence, and not inherent in the character.

Table 2.7 Using Vertical Line or Solidus Overlay

Glyph / Sequence		Glyph / Sequence		Description
∉	220A,0338	∉	220A,20D2	negated SMALL ELEMENT OF
∌	220D,0338	∌	220D,20D2	negated SMALL CONTAINS AS MEMBER
≠	2242,0338	≠	2242,20D2	negated MINUS TILDE
≢	2263,0338	≢	2263,20D2	negated STRICTLY EQUIVALENT TO
≰	2266,0338	≰	2266,20D2	negated LESS−THAN OVER EQUAL TO
≱	2267,0338	≱	2267,20D2	negated GREATER−THAN OVER EQUAL TO
∉	22F7,0338	∉	22F7,20D2	negated ELEMENT OF WITH OVERBAR
∌	22FE,0338	∌	22FE,20D2	negated SMALL CONTAINS WITH OVERBAR
≉	2A6C,0338	≉	2A6C,20D2	negated SIMILAR MINUS SIMILAR
≇	2A70,0338	≇	2A70,20D2	negated APPROXIMATELY EQUAL OR EQUAL TO
≰	2A7D,0338	≰	2A7D,20D2	negated LESS−THAN OR SLANTED EQUAL TO
≱	2A7E,0338	≱	2A7E,20D2	negated GREATER−THAN OR SLANTED EQUAL TO
⪡̸	2A95,0338	⪡	2A95,20D2	negated SLANTED EQUAL TO OR LESS−THAN
⪢̸	2A96,0338	⪢	2A96,20D2	negated SLANTED EQUAL TO OR GREATER−THAN
⪙̸	2A99,0338	⪙	2A99,20D2	negated DOUBLE−LINE EQUAL TO OR LESS−THAN
⪚̸	2A9A,0338	⪚	2A9A,20D2	negated DOUBLE−LINE EQUAL TO OR GREATER−THAN
⫅̸	2AC5,0338	⫅	2AC5,20D2	negated SUBSET OF ABOVE EQUALS SIGN
⫆̸	2AC6,0338	⫆	2AC6,20D2	negated SUPERSET OF ABOVE EQUALS SIGN

In some cases, as seen in the two preceding tables, simply using the generic glyph for the *vertical overlay* will not give the correct appearance. U+2266 ≦ LESS−THAN OVER EQUAL TO and U+2A99 DOUBLE−LINE EQUAL TO OR LESS−THAN are examples of characters that may require a taller stroke. Similarly, the generic position of the *solidus overlay* as shown for U+2AC6 SUPERSET OF ABOVE EQUALS SIGN above is not ideal.

2.18 Variation Selector

The variation selector VS1 is used to represent well−defined variants of particular math symbols. The variations include: different slope of the cancellation element in some negated symbols, changed orientation of an equating or tilde operator element, and some well−defined different shapes. These mathematical variants are all produced with the addition of U+FE00 VARIATION SELECTOR 1 (VS1) to mathematical operator base characters. To select one of the predefined variations, follow the base character with the variation selector.

Table 2.8 lists only the currently defined combinations that are of interest for mathematics. In the table, the part of the description in SMALL CAPS is the character name of the corresponding standard character; the part in lowercase indicates the variation in appearance. The table of standardized variants [StdVar] in the Unicode Character Database list the full set of all valid and recognized combinations together with their representative glyphs. All combinations not listed there are unspecified and are reserved for future standardization; no conformant process may interpret them as standardized variants. For more information, see *Section 16.4, Variation Selectors*, in Unicode 5.0 [U5.0].

Table 2.8 Variants of Mathematical Symbols using VS1

Sequence	Description
2229 + VS1	INTERSECTION with serifs
222A + VS1	UNION with serifs
2268 + VS1	LESS−THAN BUT NOT EQUAL TO − with vertical stroke

2269 + VS1	GREATER-THAN BUT NOT EQUAL TO – with vertical stroke
2272 + VS1	LESS-THAN OR EQUIVALENT TO – following the slant of the lower leg
2273 + VS1	GREATER-THAN OR EQUIVALENT TO – following the slant of the lower leg
228A + VS1	SUBSET OF WITH NOT EQUAL TO – variant with stroke through bottom members
228B + VS1	SUPERSET OF WITH NOT EQUAL TO – variant with stroke through bottom members
2293 + VS1	SQUARE CAP with serifs
2294 + VS1	SQUARE CUP with serifs
2295 + VS1	CIRCLED PLUS with white rim
2297 + VS1	CIRCLED TIMES with white rim
229C + VS1	CIRCLED EQUALS – equal sign inside and touching the circle
22DA + VS1	LESS-THAN slanted EQUAL TO OR GREATER-THAN
22DB + VS1	GREATER-THAN slanted EQUAL TO OR LESS-THAN
2A3C + VS1	INTERIOR PRODUCT – tall variant with narrow foot
2A3D + VS1	RIGHTHAND INTERIOR PRODUCT – tall variant with narrow foot
2A9D + VS1	SIMILAR OR LESS-THAN – following the slant of the upper leg – or less-than
2A9E + VS1	SIMILAR OR GREATER-THAN – following the slant of the upper leg – or greater-than
2AAC + VS1	SMALLER THAN OR slanted EQUAL
2AAD + VS1	LARGER THAN OR slanted EQUAL
2ACB + VS1	SUBSET OF ABOVE NOT EQUAL TO – variant with stroke through bottom members
2ACC + VS1	SUPERSET OF ABOVE NOT EQUAL TO – variant with stroke through bottom members

Using a variation selector allows users and font designers to make a distinction between two alternate glyph shapes *both* of which are ordinarily acceptable glyphs for generic, non-distinguishing usage of the standalone character code. This situation is somewhat analogous to the variants of Greek letterforms, which are not distinguished when used in text, but must be distinguished when used as symbols. See *Section 2.3.1, Representative Glyphs for Greek phi*. However, unlike the Greek symbols that have distinct character codes, the Unicode Standard considers the distinctions expressed via the variation selector as optional. Processes or fonts that cannot support a variation selector should yield acceptable results by ignoring it.

A variation selector only selects a different *appearance* of an already encoded character. It is not intended as a general code extension mechanism. If the two shapes can be shown to have consistently different usage and semantics in some context because of a change over time or because of better evidence about how each shape is actually used in mathematical notation, this constitutes support for adding another character so that the distinction in meaning can be expressed by a difference in character code.

2.19 Novel Symbols not yet in Unicode

Mathematicians are inventive people who continue to invent new symbols to express their concepts. Novel symbol must become established before they can be standardized. Therefore, one needs a way to handle these novel symbols in the interim.

The Private Use Areas (U+E000—U+F8FF, U+F0000—U+FFFFD, and U+100000—U+10FFFD) can be used for such nonstandard symbols. However, that can be a tricky business, because the Private Use Area (PUA) is used for many purposes. Hence when using the PUA, it is a good idea to have higher-level backup to define what kind of characters are involved. If they are used as math symbols, it would be helpful to assign them a math attribute that is maintained in a rich-text layer parallel to the plain text.

Markup languages also may have other ways of using arbitrary glyphs as 'pseudo-characters'; for

instance, MathML [MathML] has an `mglyph` element.

3 Mathematical Character Properties

Unicode assigns a number of mathematical character properties to aid in the default interpretation and rendering of mathematical characters. Such properties include the classification of characters into operator, digit, delimiter, and variable. These properties may be overridden, or explicitly specified in some environments, such as MathML [MathML], which uses specific tags to indicate how Unicode characters are used, such as <mo> for operator, <mn> for one or more digits comprising a number, and <mi> for identifier. T$_E$X, [TeX] is a higher-level composition system that uses implicit character semantics. In the following, these properties are described in greater detail.

Many Unicode characters occur nearly always as part of mathematical expressions and are given the generic mathematics property [Math]. These include the math operators in the ranges U+2200..U+22FF and U+29B0..U+2AFF, the math combining marks U+20D0..U+20FF, and the math alphanumeric characters (some of the Letterlike Symbols block at U+2100-214F, together with the Mathematical Alphanumeric Symbol block U+1D400..U+1D7FF). Other characters may occur in mathematical usage depending on context. The `math` property useful in heuristics that seek to identify mathematical expressions in plain text.

For more information about character properties, see the Unicode Character Property Model [PropMod].

3.1 Classification by Degree of Mathematical Usage

Each character in the Unicode Standard is given a General Category. This is one of a set of values that represent a primary feature or function of a character. Characters that are primarily used as mathematical symbols and operators are given the General Category (gc) value of Symbol_Math (Sm).

However, many characters commonly or exclusively used in mathematics are classified by their function as delimiting punctuation, rather than as math symbols. This particularly affects many of the math delimiters. The Math property, which is designed to be applied to all characters used primarily or exclusively with mathematical notation, is therefore a superset of the characters with gc = Sm. The difference between the sets of characters that have the math property and those for which gc = Sm, is given by the set of characters that have the Other_Math property.

3.1.11 Strongly Mathematical Characters

Strongly mathematical characters are characters that are used primarily or exclusively in mathematical notation. This includes all characters with the `math` property in Unicode.

Despite their classification as strongly mathematical characters, many characters also occur in non-mathematical texts as well, and the concept of mathematical use is deliberately kept broad. However, the delimiters in the ASCII range, such as parentheses, and brackets are so common in non-mathematical use, that they are considered weakly mathematical characters. For details on the assignment of the math property see the Unicode Character Database [UCD].

> **Note:** The `math` property in Unicode 4.0 and earlier did include these ASCII characters, and did not include many characters more specifically used for mathematics. The `math` property was revised in Unicode 4.0.1 [U4.0.1] to match the definition of strongly mathematical character presented here.

3.1.2 Weakly Mathematical Characters

Weakly mathematical characters commonly appear in mathematical expressions, but also appear in ordinary text. They include the ASCII letters and punctuation, as well as the arrows, and many of the geometric and technical shapes. The ASCII hyphen minus (U+002D -) is a weakly mathematical character that may be used for the subtraction operator, but U+2212 − MINUS SIGN is preferred for this purpose and looks better. Geometric shapes are frequently used as mathematical operators, but have other uses as well.

Weakly mathematical characters include the characters listed in Table 3.1. However this list is not comprehensive. It does not list the Miscellaneous Technical, or the Miscellaneous Symbols blocks, even though they contain characters such as the die faces or card suits that are occasionally used for a specific purpose in mathematical context. On the other hand, Table 3.1 includes characters that some authorities would not consider proper for mathematical notation.

Table 3.1 : Weakly Mathematical Characters

Code	Description
0021	Exclamation mark (factorial)
0028..0029	ASCII Parentheses
002A	ASTERISK
002C	COMMA
002F	SOLIDUS
002D	HYPHEN-MINUS
002E	FULL STOP
0030..0039	Digits
0041..005A	Uppercase Latin letters
0061..007A	Lowercase Latin letters
006E	CIRCUMFLEX ACCENT
005B,005D	Square brackets
005C	Backslash
007B,007D	Curly brackets
007E	TILDE
3010..3011	CJK brackets unified with math use
3014..3019	CJK brackets unified with math use

Additionally:

- All arrows in the Arrows block, not given the math property, except 21EA..21F3 which are specifically keyboard symbols.
- All arrows and geometric shapes in the Miscellaneous Symbols and Arrows block.
- All geometric shapes in the Geometric Shapes block, not given the math property.

The characters in Table 3.2 are compatibility variants of weakly mathematical characters. Since the list of characters that have the math property in Unicode includes compatibility variants, the characters in this table should also be considered weakly mathematical characters.

Table 3.2 : Weakly Mathematical Compatibility Characters

Code	Description
FE35..FE38	Vertical parentheses and brackets
FE47..FE48	Vertical parentheses and brackets
FE59..FE5C	CJK small forms of parentheses and brackets
FF0D	FULLWIDTH HYPHEN-MINUS
FF0F	FULLWIDTH SOLIDUS
FF08..FF09	FULLWIDTH Parentheses
FF4E	FULLWIDTH CIRCUMFLEX ACCENT
FF3B,FF3D	FULLWIDTH Square brackets
FF3C	FULLWIDTH Backslash
FF5B,FF5D	FULLWIDTH Curly brackets
FF5C	FULLWIDTH Vertical bar
FF5E	FULLWIDTH TILDE
FFE9..FFEC	Halfwidth arrows

3.1.3 Other

Any of the other Unicode characters may occur in mathematical texts, though, when they do, it is more common to find them as part of the descriptive text than as part of the mathematical expressions.

3.2 Classification by Typographical Behavior

Math characters fall into a number of subcategories, such as operators, digits, delimiters, and identifiers (constants and variables). This section discusses some of the typographical characteristics of these subcategories. These characteristics and classifications are useful in the absence of overriding information. For example, there is at least one document that uses the letter P, in upright roman typestyle, as a relational operator.

3.2.1 Alphabetic

In general italic Latin characters are used to represent single-character Latin variables. In contrast, mathematical function names like sin, cos, tan, tanh, *etc.*, are represented by upright and usually serifed text to distinguish them from products of variables. Such names should then not use the math alphanumeric characters. The upright uppercase Greek letters are favored over the italic ones. In Europe, upright d, D, e, and i can be used today for the two differential, exponential, and imaginary unit functionalities, respectively. In common American mathematical practice, these quantities are represented by italic letters. Products of italicized variables have slightly wider spacing than the letters in italicized words in ordinary text.

3.2.2 Operators

Operators fall into one or more categories. Table 3.3 shows two sets of mutually independent categories:

Table 3.3 Some Operator Categories

Category	Notes

binary	some spacing around binary operators
unary	closer to modified character than binary operators
n–ary	often called "large" operators, take limits
arithmetic	arithmetic includes binary and unary operators
logical	unary not and binary and, or, exclusive or in a host of guises
set–theoretic	inclusion, exclusion, in a variety of guises
relational	binary operators like less/greater than in many forms

As in arithmetic, operators have precedence, which streamlines the interpretation of operands and reduces the notational complexity of expressions. Operator precedence is commonly used for this purpose in computer programming languages, calculus, and algebra. Assigning consistent default precedence to the operators allows software to automate the transition from data input (or plain text) to fully marked up forms of mathematical data such as TeX or MATHML.

For example, in arithmetic, $3+1/2 = 3.5$, not 2. Similarly the plain–text expression $\alpha + \beta/\gamma$ means

$$\alpha + \frac{\beta}{\gamma} \quad \text{not} \quad \frac{\alpha + \beta}{\gamma} \ .$$

As in arithmetic, precedence can be overruled by explicit delimitation, so $(\alpha + \beta)/\gamma$ gives the latter.

3.2.3 Large Operators

Large Operators include n–ary operators like summation and integration. They may expand in size to fit their associated expressions. They generally also take limits. The placement of the limits on an operator is different when it is used in–line compared to its use in displayed formulae. For example when the expression $\sum_{n=1}^{\infty} a_n$ is laid out in–line, the limits are placed at the top and bottom right hand side. However, when displayed out–of–line, as in

$$\sum_{n=1}^{\infty} a_n$$
.

the limits are normally placed above and below. The Unicode Standard does not specify any particular layout for limit expressions, instead, it assumes that implementations follow the accepted typographical practices for mathematical layout.

European tradition prefers a more upright shape for the integrals. To implement this style preference an appropriate font must be used, as there is only a single character code for each integral.

3.2.4 Digits

Digits include 0–9 in various styles. All digits of a particular style have the same width.

3.2.5 Delimiters

Delimiters include punctuation, opening/closing delimiters such as parentheses and brackets, braces, and fences. Opening and closing delimiters and fences may expand in size to fit their associated expressions. Some bracket expressions do not appear to be "logical" to readers unfamiliar with the

notation, for example, $]x,y[$. In right–to–left layout, delimiters are mirrored. See Section 4.2, Bidirectional Layout of Mathematical Text.

3.2.6 Fences

Fences are similar to opening and closing delimiters, but are not paired.

3.2.7 Combining Marks

Combining marks are used with mathematical alphabetic characters (see *Section 2.6,* *Accented Characters*), instead of precomposed characters. Use <U+0061, U+0308> for the second derivative of acceleration with respect to time, not the precomposed letter ä. On the other hand, precomposed characters are used for operators whenever they exist. Combining slash (solidus) or vertical overlays can be used to indicate negation for operators that do not have precomposed negated forms.

Where both long and short combining marks exist, use the long, for example, use U+0338, not U+0337 COMBINING SHORT OVERLAY and use U+20D2, not U+20D3 COMBINING SHORT VERTICAL LINE OVERLAY. The actual shape or position of a combining mark is a typesetting problem and not specified in plain text. When using combining marks, the composite characters have the same typesetting class as the base character.

In MathML combining marks are used to select math accents, which may be applied to single variables or entire expressions. If possible, do not use combining marks to denote math accents, but use the spacing equivalent. For example, instead of U+0303 COMBINING TILDE use U+02DC SMALL TILDE, which is a spacing character. The reason for that recommendation is that such combining marks would start an element, and, in the source code, would therefore combine with the preceding ">". While this ordinarily does not present problems for parsers, a particularly challenging case is U+0338 COMBINING LONG SOLIDUS OVERLAY because it is part of a canonical decomposition of U+226F ≯ NOT GREATER–THAN.

4 Implementation Guidelines

4.1 Use of Normalization with Mathematical Text

If Normalization Form C is applied to mathematical text, some accents or overlays used with BMP alphabetic characters may be composed with their base character, even though for mathematical text the decomposed forms would have been preferred. Parsers should allow for this. Normalization forms KC or KD remove the distinction between different mathematical alphabets. These forms *cannot* be used with mathematical texts. For more details on Normalization see Unicode Standard Annex #15, "Unicode Normalization Forms" [Normalization] and the discussion in *Section 2.6,* *Accented Characters*.

If combining accents follow syntax characters in a markup language, there may be several issues. A source editor might display the combining mark as if the syntax character was the intended base character. This is an issue where the syntax character precedes data, such as for the terminating > characters. This is usually not an issue in processing the data, as the parser can correctly separate the data from the syntax characters.

However, U+0338 ̸ COMBINING LONG SOLIDUS OVERLAY is a combining diacritical mark that combines with U+003E > GREATER–THAN SIGN under NFC (producing U+226F ≯ NOT GREATER–THAN). That means that NFC changes the encoding of the syntax character in this case. On the other hand, the parser should probably not try to decompose any instances of the *not greater than* operator. Therefore, use of

U+0338̸ following a markup tag does not work. In [MathML] 2.0 mathematical accents are tagged with <mo> (operator) tag so the accents do not appear directly in mathematical text. But that causes U+0338̸ to follow >. Because normalization changes U+003E > to U+226F, if followed by U+0338̸ an alternative representation is needed. In this case it would be useful to allow the use the ASCII / as an alias for 0338, for example,

```
<mover accent="true")
  <mo>=</mo>
  <mo>/</mo>
</mover>
```

Because MathML already uses spacing diacritics as aliases for the actual combining marks where they exist, this extension would not be too disruptive. In plain HTML or XML the use of precomposed U+226F ≯ does not give any problems, as long as data is not normalized with NFD.

4.2   Bidirectional Layout of Mathematical Text

In a bidirectional context, the glyphs for mathematical operators and delimiters, other than arrows, are adjusted as described in Unicode Standard Annex #9, "The Bidirectional Algorithm" [Bidi]. During display, the software must ensure that the rendered glyph is the correct one in the context of bidirectional texts.

Characters such as parentheses, whose images are mirrored horizontally in text that is laid out from right to left have the Bidi_Mirrored property. For example, in a left-to-right context, U+0028 (LEFT PARENTHESIS will appear as "(", while in a right-to-left context it will appear with the mirrored glyph ")". In some mathematical usage, brackets may not be paired, or may be deliberately used in the reversed sense, such as]a,b[. Mirroring assures that in a right-to-left environment, such specialized mathematical text continues to read]b,a[and not [b, a].

If any of these expression is displayed from right to left, then the mirrored glyphs are used. Because of the difficulty in interpreting such expressions, authors of bidirectional text need to make sure that readers can determine the desired directionality of the text from context. Mirroring is not limited to paired characters: any character with the mirrored property will need two mirrored glyphs–for example, U+222B ∫ INTEGRAL

For some mathematical symbols, the "mirrored" form is not an exact mirror image. For example, the direction of the circular arrow in U+2232 ∲ CLOCKWISE CONTOUR INTEGRAL reflects the direction of the integration along the contour, not the text direction. In a right-to-left context, the integral sign would be mirrored, but the circular arrow would retain its clockwise direction. Another example is the bidi-mirrored form of U+221B ∛ CUBE ROOT, which consists of a mirrored radix symbol with a non-mirrored digit '3'.

The list of mirrored characters appears in the Unicode Character Database [UCD]. This normative property is not to be confused with the related Bidi Mirroring Glyph property, an informative property, which can assist in rendering a subset of mirrored characters in a right-to-left context by mapping to a paired character which happens to have the mirrored glyph. For more information, see BidiMirroring.txt in the Unicode Character Database.

Arrows. In bidirectional layout, arrows are *not* automatically mirrored, because the direction of the arrow could be relative to the text direction or relative to an absolute direction on the page or in a diagram. Therefore, if text is copied from a left-to-right to a right-to-left context or vice versa, the character code for the desired arrow direction in the new context must be used. For example, it might be necessary to change 21D2 ⇒ RIGHTWARDS DOUBLE ARROW to U+21D0 ⇐ LEFTWARDS DOUBLE ARROW to maintain the semantics of "implies" in a right-to-left context.

See also *Section 4.7, Bidi Mirrored (normative)* in [Unicode] and "Semantics of Paired Punctuation" subsection in *Section 6.2, General Punctuation*, in [Unicode].

4.3 Input of Mathematical and Other Unicode Characters

In view of the large number of characters used in mathematics, a brief and informal discussion of possible approaches for input methods may be appropriate. Most keyboard layouts support the ASCII letters, digits and some of the more common math symbols and delimiters, for example, + − / * [ ] ( ) { }. In addition to the limits on the number of symbols supported for direct keyboard entry, sometimes the ASCII character only approximates the proper mathematical character.

Post-entry Correction. From a syntactical point of view, U+2212 − MINUS SIGN is certainly preferable to the U+002D - HYPHEN-MINUS in the ASCII range and U+2032 ′ PRIME is preferable to U+0027 ' APOSTROPHE, but users may locate the ASCII characters more easily. Similarly, it is easier to type ASCII letters than italic letters, but when used as mathematical variables, such letters are traditionally italicized in print. Accordingly a user might want to make italic the default alphabet in a math context, reserving the right to overrule this default when necessary. Other post-entry enhancements include automatic-ligature and left-right quote substitutions, which can be done automatically by some word processors. Intelligent input algorithms can dramatically simplify the entry of mathematical symbols.

Input Method Editors. Many systems support interfaces for a user-selectable Input Method Editor (IME). While the technology of IMEs and the interfaces that support them were developed based on the needs of East Asian language input, the task of selecting one of over thousand mathematical symbols at input time could be solved with a similar approach making use of the existing interfaces.

Math Keyboards. A special math shift facility for keyboard entry could bring up proper math symbols. The values chosen can be displayed on an *on-screen keyboard*. For example, the left Alt key could access the most common mathematical characters and Greek letters, the right Alt key could access italic characters plus a variety of arrows, and the right Ctrl key could access script characters and other mathematical symbols. On systems that support it, the numeric keypad offers locations for a variety of symbols, such as sub/superscript digits using the left Alt key. Left Alt CapsLock could lock into the left-Alt symbol set, etc. This approach yields what one might call a "sticky" shift. Other possibilities involve the NumLock and ScrollLock keys in combinations with the left/right Ctrl/Alt keys. This approach rapidly approaches literally billions of combinations, that is, several orders of magnitude more than Unicode can handle!

Macros. The auto-correct and keyboard macro features of some word processing systems provide other ways of entering mathematical characters for people familiar with TeX. For example, typing `\alpha` inserts α if the appropriate auto-correct entry is present. This approach is noticeably faster than using menus.

Hexadecimal input. A handy hex-to-Unicode entry method works with recent Microsoft text software (similar approaches are available on other systems) to insert Unicode characters, including math

characters. Basically one types the hexadecimal code (in ASCII), making corrections as need be, and then types Alt+x. The hexadecimal code is replaced by the corresponding Unicode character. The Alt+x can be a toggle, that is, type it once to convert a hex code to a character and type it again to convert the character back to a hex code. If the hex code is preceded by one or more hexadecimal digits, one needs to "select" the code so that the preceding hexadecimal characters are not included in the code. The code can range up to the value 0x10FFFF, which is the highest character in the 17 planes of Unicode.

Pull-down Menus. Pull-down menus are a popular, but slow method for handling large character sets. A better approach is the *symbol box*, which is an array of symbols either chosen by the user or displaying the characters in a font. Symbols in symbol boxes can be dragged and dropped onto key combinations on an on-screen keyboard, or directly into applications. On-screen keyboards and symbol boxes are valuable for entry of mathematical expressions and of Unicode text in general.

4.4 Use of Math Characters in Computer Programs

It can be very useful to have typical mathematical symbols available in computer programs. To realize the full potential of supporting mathematical symbols as part of identifiers, a development environment should display the desired characters in both edit and debug windows. While a preprocessor could be used translate MathML, for example, into C++, it would not be able to make the debug windows use the math-oriented characters because the language cannot handle the underlying Unicode characters. Java has made an important step in this direction by allowing Unicode characters to be used in identifiers. The mathematical alphanumeric symbols make this approach quite powerful for the user with relatively little effort for compilers.

There are three key advantages of using Unicode characters directly in computer program identifiers:

1. Many formulas in document files can be programmed simply by copying them into a program file and inserting appropriate multiplication dots. This dramatically reduces coding time and errors.
2. The use of the same notation in programs and the associated journal articles and books leads to an unprecedented level of self-documentation.
3. In addition to providing useful tools for the present, these proposed initial steps will ease the way toward the ultimate goal of teaching computers to understand and use arbitrary mathematical expressions.

For more information on identifiers and syntax characters, see Unicode Standard Annex #31, *Identifier and Pattern Syntax* [Identifier].

4.5 Recognizing Mathematical Expressions

Mathematical expressions must be formatted using different rules than those applied to the surrounding text. When markup is used, the limits of the mathematical text are defined explicitly. In plain text it is possible to use a number of heuristics for identifying mathematical expressions. Once recognized, they can be treated appropriately, for example expressions input as plain text could be tagged with a rich-text math style. Such math style would connect in a straightforward way to appropriate MathML tags. Heuristics are not foolproof, but if they are applied as part of post-entry correction, the user could override cases that were tagged incorrectly.

Heuristics are based on the fact that a large set of characters in Unicode are primarily or exclusively used for mathematics, see *Section 3.1.1, Strongly Mathematical Character*s. Their presence potentially

identifies their surrounding characters as math characters as well. For example, the fraction (U+2044) and ASCII slashes, would tend to identify the characters immediately surrounding them as parts of mathematical expressions. The same applies to other mathematical characters and operators. On the other hand, many Unicode characters are not mathematical in nature and suggest that the characters immediately preceding or following them are not parts of mathematical expressions.

If Latin or Greek letter mathematical variables are already given in one of the math alphabets, they are considered parts of math expressions. If they are not, one can still use some recognition heuristics as well as the opportunity to italicize appropriate variables. Specifically, ASCII letter pairs surrounded by whitespace are often mathematical expressions, and should be converted to math italics. If a letter pair fails to appear in a list of common English and European two-letter words, it is treated as a mathematical expression and converted to italics.

Strings of characters containing no whitespace but containing one or more unambiguous mathematical characters are generally treated as mathematical expressions. Certain two-, three-, and four-letter words inside such expressions should *not* use italics. These include trigonometric function names like sin and cos, as well as ln, cosh, etc. Words or abbreviations that are often used as subscripts *should not be italicized*, even when they clearly appear inside mathematical expressions.

4.6 Some Examples of Mathematical Notation

This section gives some additional, but still relatively straightforward examples of mathematical notation for the benefit of readers not familiar with it. There are two styles for presenting mathematical formula in text. Simple expressions are often presented in the so called inline format to conserve space and not break up the text. More complex formulae or those to which the author wants to call attention or that need to be numbered, are built-up and presented in the so called display style. This use of the word display is not to be confused with the action of making text visible on display devices. The examples shown here are enlarged for clarity.

The simple built-up fraction

$$\frac{abc}{d}$$

appears in inline text as $(abc)/d$; similarly the inline text $(a+c)/d$ could appear as

$$\frac{a+c}{d}$$

when built-up. For the ratio

$$\frac{\alpha_2{}^3}{\beta_2{}^3 + \gamma_2{}^3} \, ,$$

an inline format is $\alpha_2{}^3 / (\beta_2{}^3 + \delta_2{}^3)$.

The size of mathematical delimiters or operators may change on the size of the enclosed text. In an equation such as

$$W\,{}^{3\beta}_{\delta_1\rho_1\sigma_2} = U\,{}^{3\beta}_{\delta_1\rho_1} + \frac{1}{8\pi^2} \int_{\alpha_1}^{\alpha_2} d\alpha_2{}' \left[\frac{U\,{}^{2\beta}_{\delta_1\rho_1} - \alpha_2{}' U\,{}^{1\beta}_{\rho_1\sigma_2}}{U\,{}^{0\beta}_{\rho_1\sigma_2}} \right]$$

the size of the bracket scales with the size of the enclosed expression, in this case a fraction, and the size of the integral scales with the size of the integrand. This example also shows the positioning of multiple sub– and superscripts as well as the positioning of limit expressions on the integral..

Punctuation following math in display is commonly on the local baseline or centerline.

5 Data Files

5.1 Mathematical Classification

The data file [Data] provides a classification of characters by primary usage in mathematical notation. The classes used in this file are defined as follows:

Table 5.1 Classes of Mathematical Characters

Class	Name	Comments
N	Normal	This includes all the digits and symbols requiring only one form
A	Alphabetic	
B	Binary	
C	Close	Paired with opening delimiter
D	Diacritic	
F	Fence	Unpaired delimiter
O	Open	Paired with closing delimiter
L	Large	N–Ary or Large operator, often takes limits
P	Punctuation	
R	Relation	Includes arrows
U	Unary	Operators that can be unary or binary

Character classification information will be updated when new characters are added to the standard, or to amend the classification of existing characters as necessary. The data file specifies the version of the [Unicode] to which it has been updated.

5.2 Mapping to other Standards

The mapping data file [Mapping] contains mappings to standard entity sets commonly used for SGML and MathML documents. Mapping data will be updated when new mapping information becomes available.

6 Security Considerations

The use of the repertoire of mathematical characters in a mathematical context is not known to present special security considerations. However, many mathematical symbols can be confused with characters used in regular text. In particular, the mathematical alphanumeric symbols described in *Section 2.2, Mathematical Alphabets* can be confused with styled text. These characters are therefore excluded from use in security sensitive environments, such as domain names. For more information, see Unicode Technical Standard #36, "Unicode Security Considerations" [Security].

References

[Bidi]   Unicode Standard Annex #9: *Unicode Bidirectional Algorithm*
                http://www.unicode.org/reports/tr9/

[Charts]   The online code charts can be found at http://www.unicode.org/charts/
                An index to characters names with links to the corresponding chart is
                found at http://www.unicode.org/charts/charindex.html

[Data]   Classification of math characters by usage: MathClass-7d2.txt
                For earlier versions of the data file see prior versions of this report.

[EAW]   Unicode Standard Annex #11, *East Asian Width*.
                http://www.unicode.org/reports/tr11
                For a definition of East Asian Width

[FAQ]   Unicode Frequently Asked Questions
                http://www.unicode.org/faq/
                For answers to common questions on technical issues.

[Feedback]   *To report errors or submit suggestions please use*
                http://www.unicode.org/reporting.html

[Glossary]   Unicode Glossary
                http://www.unicode.org/glossary/
                For explanations of terminology used in this and other documents.

[Identifier]   Unicode Standard Annex #31: *Identifier and Pattern Syntax*
                http://www.unicode.org/reports/tr31/

[ISO9573]   ISO TR9573-13: *Information technology – SGML support facilities
                – Techniques for using SGML*
                Part 13: Public entity sets for mathematics and sciences

[LaTeX]   *LaTeX: A Document Preparation System, User's Guide & Reference Manual*,
                2nd edition, by Leslie Lamport (Addison-Wesley, 1994; ISBN 1-201-
                52983-1)

[Mapping]   Mapping Data (Location TBD)

[Math]   Math Property
                Defined in the Unicode Character Database, see
                http://www.unicode.org/Public/UNIDATA/UCD.html#Math

[MathML]   *Mathematical Markup Language (MathML™) Version 2.0 . (W3C*

Recommendation, second edition 10 October 2003) Editors: David Carlisle, Patrick Ion, Robert Miner and Nico Poppelier.
For the latest MathML specification see
http://www.w3.org/TR/MathML/

[Meystre]   P. Meystre and M. Sargent III (1991), *Elements of Quantum Optics*, Springer-Verlag

[NISTGuide]   NIST publication 811, *Guide for the use of the international system of units.*
See: http://physics.nist.gov/Pubs/pdf.html

[NISTStyle]   *Typefaces for Symbols in Scientific Manuscripts*
See: http://physics.nist.gov/Document/typefaces.pdf

[Normalization] Unicode Standard Annex #15: *Unicode Normalization Forms*
http://www.unicode.org/reports/tr15/

[OpenMath]   *The OpenMath Standard, 1.0, see:*
http://www.openmath.org/cocoon/openmath/standard/index.html

[PropMod]   Unicode Technical Report #23: *The Unicode Character Property Model*
http://www.unicode.org/reports/tr23/

[Reports]   Unicode Technical Reports
http://www.unicode.org/reports/
For information on the status and development process for technical reports, and for a list of technical reports.

[Security]   Unicode Technical Standard #36, *Unicode Security Considerations*
http://www.unicode.org/reports/tr36/

[SI]   International System of Units (SI) – *Le Système International d'Unités*. The metric system of weights and measures based on the meter, kilogram, second and ampere, Kelvin and candela.
For background information see
http://physics.nist.gov/cuu/Units/index.html.

[StdVar]   For the formal list of Standardized Variants in the Unicode Character Database, see:
http://www.unicode.org/Public/UNIDATA/StandardizedVariants.html
(with glyphs) or
http://www.unicode.org/Public/UNIDATA/StandardizedVariants.txt

[STIX]   STIX Project Home Page: http://www.ams.org/STIX/

[TeX]   Donald E. Knuth, *The T_EXbook*, (Reading, Massachusetts: Addison-Wesley 1984)
The T_EXbook is the manual for Donald Knuth's T_EX composition system. Appendix G describes the somewhat idiosyncratic mechanism used by T_EX to accomplish the composition of mathematical notation; it is based on

the principles laid out in [Chaundy, Wick, Swanson], as well as on examination of a large number of published samples that demonstrated Knuth's style preferences.

Donald E. Knuth, *T_EX, the Program*, Volume B of *Computers & Typesetting*, (Reading, Massachusetts: Addison–Wesley 1986)

See also http://www.ams.org/tex/publications.html

[U3.0]   *The Unicode Standard, Version 3.0,* (Reading, MA, Addison–Wesley, 2000. ISBN 0-201-61633-5) or online as http://www.unicode.org/uni2book/u2.html

[U3.1]   Unicode Standard Annex #27: *Unicode 3.1* http://www.unicode.org/reports/tr27/

[U3.2]   Unicode Standard Annex #28: *Unicode 3.2* http://www.unicode.org/reports/tr28/

[U4.0]   *The Unicode Standard, Version 4.0,* (Boston, MA, Addison–Wesley, 2003. ISBN 0-321-18578-1) or online as http://www.unicode.org/versions/Unicode4.0.0/

[U4.0.1]   *Unicode 4.0.1,* http://www.unicode.org/versions/Unicode4.0.1/

[U4.0.1]   *Unicode 4.1.0,* http://www.unicode.org/versions/Unicode4.1.0/

[U5.0]   The Unicode Consortium. *The Unicode Standard, Version 5.0* (Boston, MA, Addison–Wesley, 2007. ISBN 0-321-48091-0) or online as http://www.unicode.org/versions/Unicode5.0.0/

[UCD]   Unicode Character Database. http://www.unicode.org/Public/UNIDATA/UCD.html *For and overview of the Unicode Character Database and a list of its associated files*

[Unicode]   The latest version of the Unicode Standard can be found at http://www.unicode.org/versions/latest/

[UXML]   Unicode Technical Report #20: *Unicode in XML and other Markup Languages* http://www.unicode.org/reports/tr20/

[Versions]   Versions of the Unicode Standard http://www.unicode.org/standard/versions/ *For details on the precise contents of each version of the Unicode Standard, and how to cite them.*

[XML]   Tim Bray, Jean Paoli, C. M. Sperberg-McQueen, Eve Maler, Eds., *Extensible Markup Language (XML) 1.0 (Second Edition)*, W3C Recommendation 6-

October-2000, <http://www.w3.org/TR/REC-xml/>

Additional References

The following four books are entirely about the composition of mathematics:

[Chaundy]   T.W. Chaundy, P.R. Barrett and Charles Batey, *The Printing of Mathematics*, (London: Oxford University Press 1954, third impression, 1965) [out of print]

[Wick]   Karel Wick, *Rules for Type-setting Mathematics*, (Prague: Publishing House of the Czechoslovak Academy of Sciences 1965) [out of print]

[Swanson]   Ellen Swanson, *Mathematics into Type*, (Providence, RI: American Mathematical Society, 1971, revised 1979, updated 1999 by Arlene O'Sean and Antoinette Schleyer)
The original edition is based on "traditional" composition (Monotype and "cold type", that is Varityper and Selectric Composer); the 1979 edition adds material for computer composition, and the 1999 edition mostly assumes T_EX or a comparably advanced system.

[Byrd]   *Mathematics in Type*, (Richmond, VA: The William Byrd Press 1954) [out of print]

The following books contain material on mathematical composition, but it is not the principal topic covered:

[Maple]   *The Maple Press Company Style Book*, (York, PA: 1931) (reprinted 1942)
Contains sections on fractions; mathematical signs; simple equations; alignment of equations; braces, brackets and parentheses; integrals, sigmas and infinities; hyphens, dashes and minus signs; superiors and inferiors; ...
[out of print]

[Manual]   *A Manual of Style, Twelfth Edition, Revised* (Chicago: The University of Chicago Press 1969)
A chapter "Mathematics in Type" was produced using the Penta (computer) system.

Acknowledgements

Modifications

Changes from Revision 7

Split the data file into separate classification and mapping data. Added a section discussing bidirectional layout. Updated the discussion of geometrical shapes and combining marks.

Changes from Revision 6

Added information on characters added in Unicode 4.1 and Unicode 5.0. This includes discussion of dotless characters and horizontal delimiters. Split the listing of weakly mathematical characters into two numbered tables 3.1 and 3.2. Added a section on security considerations. Integrated the results of extensive copy editing. Added section 4.2 on mirroring. (AF)

Changes from Revision 5

Rewrote the Overview. Brought table 2.7 into alignment with the standardized variant listing in the Unicode Character Database: 2278 and 2279 have been moved to table 2.5. 2225 was removed from table 2.7 since there is now a new character 2AFD and the variation is no longer needed. Added Table 2.3. Added Section 2.15. Removed section 3.3. Renumbered the appendix to become Section 5. Moved the actual classification of characters into a separate data file. Updated references to the Unicode Standard to Unicode 4.0 where appropriate. Improved the layout of tables 2.5, 2.6 and 2.7. Many minor spelling, wording and formatting fixes throughout. Updated status and conformance section. Completed the classification in sections 3.1.1 and 3.1.2. Changed header and improved visual layout of the data file. (AF)

Changes from Revision 4

Added section 2.16. Added section 3.3. Removed section 5 on plain text math. Added Appendix A. Added a few typographical samples. (AF)

Changes from Revision 3

Fixed some CSS issues.

Changes from Revision 2

Changed many special symbols to NCRs. Fixed an HTML glitch affecting table formatting and fixed contents of Table 2.4. A number of additional typographical mistakes and inconsistencies in the original proposed draft have been corrected. Merged duplicated text in section 2.7 and made additional revisions to further align the text with Unicode 3.2. Minor wording changes for clarity or consistency throughout. (bnb/AF).

Changes from Revision 1

A large number of minor, but annoying typographical and HTML mistakes in the original proposed draft have been corrected. This includes the occasional mistaken character name or code point. Additional entries were made to the references section and new bookmarks and internal links have been added to refer to them from the text. Other minor improvements to the text and formatting have been carried out. Added Section 2.10 and revised the first paragraph of Section 2 to bring the text inline with Unicode 3.2 (bnb/AF)