The STIX Project — From Unicode to fonts

Barbara Beeton
American Mathematical Society
201 Charles Street
Providence, RI 02904-2294
USA
bnb at ams dot org

Abstract

The goal of the STIX project is to provide fonts usable with other existing tools to make it possible to communicate mathematics and similar technical material in a natural way on the World Wide Web. This has involved two major efforts: enlarging Unicode to recognize the symbols of the mathematical language, and creating the fonts necessary to convert encoded texts into readable images.

This ten-year effort is finally resulting in fonts that can actually be used for the intended purpose.

1 Introduction: What is Unicode?

According to the Unicode manual, the original goal of the effort was "to unify the many hundreds of conflicting ways to encode characters, replacing them with a single, universal standard."

Unicode is thus an encoding system capable of representing all the world's languages in a way that will enable any person to interact with a computer in his own language. Nearly all modern computer operating systems are based on Unicode.

The three principal components of Unicode are the character, the block and the plane. A character is the smallest unit, carrying a semantic value. A character may represent a letter, a digit, or some other symbol or function.

A block consists of 256 characters — the number of characters that can be addressed by eight binary digits, addressed as 00–FF. A plane is composed of 256 blocks, for a total of 65,536 characters; there are 17 planes for a capacity of 1,114,112 in all [4, p. 2].

The first plane, Plane 0, is referred to as the Basic Multilingual Plane (BMP); if a piece of software claims to support Unicode, it should be able to access every character in the BMP.

Characters are assigned to blocks with the most heavily used given the lowest addresses; assignments are made in half-block (128-byte) chunks.

The first half of block 00 is the basic character set known as ASCII (the formal name is "C0 Controls and Basic Latin"). This contains the upper- and lowercase Latin alphabet, ten digits, various punctuation marks, and a number of control functions; it is the set of characters found on most computer keyboards. The second half of block 00 is known as "Latin 1", and includes many accented letters found in western European languages, as well as additional punctuation marks and control characters.

The next few blocks contain:

- the Greek and Cyrillic alphabets;
- a collection of diacritics to be used to compose accented letters not accommodated by Latin 1;
- Hebrew;
- Arabic;
- the scripts for many of the languages of India and southeast Asia.

Each script is allotted a half or full block as needed.

Blocks from 10 to 1F accommodate more language scripts, including extensions for Latin and Greek. Except for very basic symbols such as plus (+) or asterisk (*), non-language characters aren't included until blocks beginning at 20.

2 Who is responsible for Unicode, and how do things get added?

Unicode was developed and is maintained by the Unicode Technical Committee (UTC) an arm of the Unicode Consortium. Members of the consortium include most computer hardware manufacturers and software vendors. To align Unicode with ISO 10646, the standard on which hardware and software are actually based, the UTC works closely with the standardization subcommittee for coded character sets of the International Organization for Standardization.

The UTC members are individuals with various areas of expertise. Most have a strong background in

computer software. Many are skilled as well in languages and linguistic-related areas. However, there are very few practicing physical scientists.

If something isn't in Unicode, there is a standard proposal form. This asks for a number of items:

- the repertoire of characters being requested, including character names;
- the context in which the proposed characters are used;
- references to authoritative published sources where the characters have been used;
- relationships the proposed characters bear to characters already encoded;
- contact information for the supplier of a computerized font to be used in printing the standard;
- names and addresses of contacts within national or user organizations.

The on-line description of the proposal review process warns that

- international standardization requires a significant effort on the part of the submitter;
- it frequently takes years to move from an initial proposal to final standardization;
- submitters should be prepared to become involved in the process.

In the case of the STIX proposal, all these warnings were true, in spades.

3 Initial conditions

In 1997, when the STIX project began, Unicode was at version 2.0. It contained several blocks of interest for mathematics:

- combining diacritics (first half of block 03 for text; the last three 16-cell columns of block 20 for diacritics used with symbols)
- Greek (last half of block 03)
- arrows (last half of block 21) (Figure 1)
- mathematical operators (block 22) (Figure 2)
- miscellaneous technical (first half of block 23)
- geometric shapes (last half of block 25)

None of these blocks was entirely full at that time.

4 Character ≠ glyph

Unicode encodes *characters*. Each character has a designated, well-defined meaning. It appears in the Unicode charts as a representative glyph, or image. However, since the purpose of Unicode is to convey meaning, the shape of the glyph may vary. To take a trivial example, in text, an "A" has the same code whether it is upright Roman, italic (A), bold ($\mathbf{A}$),

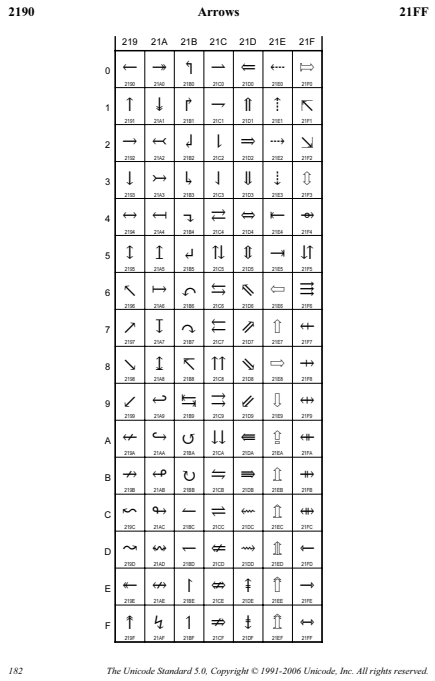

Figure 1: When the STIX project began, positions 21EB–21FF were empty. Copyright Unicode, used by permission.

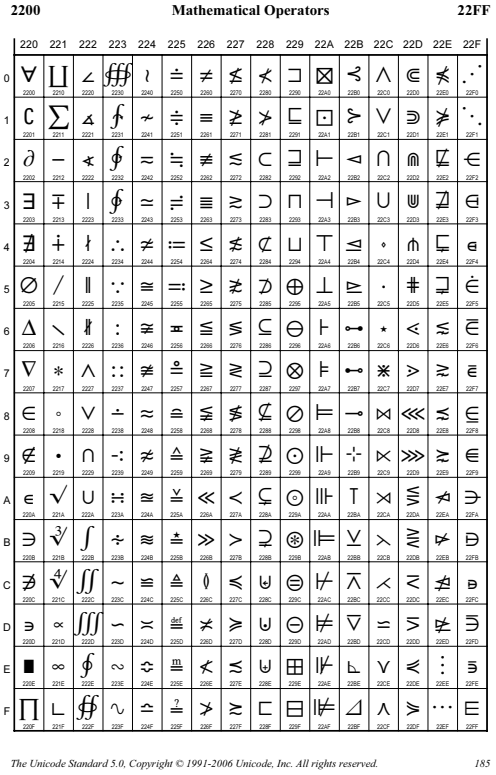

Figure 2: Math operators in Unicode; at the start of the STIX project, the last code assigned was 22F1. Copyright Unicode, used by permission.

or sans serif (A). Similarly, an accented "é" can be represented either by one code or by a combination of the letter "e" and the combining diacritic "´".

This is not true for math notation, however. The same letter in different styles (italic, script, Fraktur, bold, ...) means different things. This is illustrated by the Hamiltonian equation from physics:

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

In 1997, at the beginning of the STIX project, there was no way to unambiguously identify the script $\mathcal{H}$. Based only on the encoding, it was indistinguishable from the H on the right side of the equation:

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

Something more was needed; early proposals by UTC members recommended *markup* (e.g., font changes), such as provided by XML or MathML. However, it was realized that a physicist might wish to search for this entity in a corpus or database, and searching would be much more reliable if it could be done using an unambiguous code.

The UTC solution was to incorporate a substantial set of *mathematical alphanumerics*, about 1,000 characters. These variations on the Latin and Greek alphabets fill four complete blocks (U+1D40–U+1D7F) in Plane 1. Placement outside the BMP was meant to discourage casual users from using these special alphabets for things such as wedding invitations, where stylistic markup is more appropriate.

Another facet of the character/glyph dichotomy is the use in math notation of different-sized operators in text vs. display environments — the size used in text is generally smaller; compare $\sum_{i=0}^{\infty} x_i$ and

$$\sum_{i=0}^{\infty} x_i \,.$$

The sum symbol is just a single character in Unicode. Delimiters (parentheses, brackets, etc.) are also considered to be single characters, but they must be provided in many sizes, including segments suitable for piecing together to span multiple lines.

Unicode takes the position that such substitutions are the responsibility of the application.

5 Requesting additions to Unicode

In addition to the approximately 1,000 mathematical alphanumerics already mentioned, the STIX collection identified roughly 1,000 non-alphanumeric symbols that couldn't be found in Unicode version 2. These were assigned provisional identifiers in the Unicode Private Use Area (PUA) in order to keep track of them. IDs were assigned in order of accession, rather than by shape, usage, or other rational system.

Because of the large number of characters being requested, the UTC invited a representative of STIX to present the proposal in person at a regular UTC meeting, to answer questions directly, rather than carrying on an extensive paper and e-mail interchange. The fact that the proposal was backed by five professional societies and a technical publisher, based on actual experience in their publications, probably lessened the usual requirement for extensive examples. This did not mean that there was no requirement to justify every symbol; it did, however, allow symbols to be considered in groups rather than individually — if one member of a coherent symbol group (e.g., arrows with a triple stem pointing in several directions) was accepted, the rest of the group was accepted as well.

As noted earlier, Unicode assigns characters in blocks, preferably of groups with some inherent relationships. The UTC experts, acting on usage information provided with the proposed characters, classified them into groups that corresponded to the existing symbol blocks: operators, arrows, geometrics, and so forth. Then began the process of shoehorning them into the code space. First, the gaps in existing blocks were filled with appropriate items. Next, the number of characters in each category was tallied, and new blocks of appropriate sizes assigned. The bulk of the math additions first appeared (on line) in Unicode version 3.2, with the first paper publication in version 4.0.

As of Unicode version 5.0, these new blocks have been added:

- miscellaneous mathematical symbols A (U+2700–U+27EF)
- supplemental arrows A (U+27F0–U+27FF)
- supplemental arrows B (U+2900–U+297F)
- miscellaneous mathematical symbols B (U+2980–U+29FF)
- supplemental mathematical operators (U+2A00–U+2AFF)
- miscellaneous symbols and arrows (U+2B00–U+2BFF)

Not all of these blocks are filled yet, but space has been left where experience has shown growth is likely to occur.

One other key feature was adopted: a *variation selector* — a one-character code (U+FE00 for math symbols) identifying the preceding character as having the same meaning, but an alternate shape which cannot be composed from a base character plus a

combining diacritic. An example is the relation ⪈ (U+2269) vs. ⪈ (U+2269,U+FE00). The use of the variation selector is very tightly controlled; all characters using it must be accepted explicitly by the UTC. Other shape variations must be indicated by markup and recognized by the application software.

Some important decisions were made during the course of this exercise that should make future submissions progress more smoothly.

First and foremost, it was accepted that mathematics is a language, and that symbols used in this context are as essential as the letter “e” is to English. Another “given” is that math notation is open-ended — mathematicians and other scientists will continue to invent and adopt new symbols, so the job isn't done, and may never be.

Just within the past few months, a mathematician from Morocco has submitted documentation of mathematical notation in Arabic — it is a mirror image of what we see in European language contexts. This generated a flurry of activity in the UTC to adopt a rational collection of right-to-left symbols to complement the basically left-to-right symbols already present. The new material will appear in Unicode version 5.1.

6 What wasn't accepted, and why not?

In spite of the generally high level of acceptance of characters proposed by STIX, the UTC rejected some symbols. The reason for most rejections was that they weren't “math”. Symbols used by other disciplines (astronomy, meteorology) were not considered to be relevant to the STIX request; it was suggested that an organization involved in those disciplines should make a separate submission, at which time it would be considered on its own merits.

Some symbols were rejected because they were easily constructed as compounds of existing characters and combining diacritics; this includes any negated relations that hadn't already been encoded.

For some symbols, in particular ones that were identified after the initial proposal, the available documentation was deemed insufficient for acceptance. However, when a suitable in-context published example is found, acceptance of these stragglers is very likely.

Finally, some items in the STIX collection aren't considered independent symbols; they are partial glyphs used for constructing larger symbols such as multi-line parentheses or braces, or extenders for arrows. These weren't even submitted to the UTC since they fall into the area that is the responsibility of application software.

7 Okay, Unicodes have been assigned; how can we print them?

Assignment of Unicodes, while necessary, is not sufficient for use of these symbols in electronic or paper communication. It is also necessary to be able to generate images that can be understood by someone trying to read them. Here is where fonts come in.

A popular font for typesetting of math is Times Roman or one of its variants. This font, originally designed for newspaper use, is compact (a lot of material can be squeezed onto a page), and is legible at small sizes. Its adoption for technical material means that a large number of symbols have been designed to be compatible. Times Roman was the overwhelming choice of the STIX organizations as the base font around which the new STIX fonts would be created.

There are some very specific design criteria for a font intended for math:

- Each letter must be unambiguously recognizable in isolation; for Times, this means that a substitute must be provided for the italic v, since the usual Times shape is too easily confused with the Greek letter nu ν.
- Hairlines must be thick enough to keep shapes from breaking up in sub- and superscripts, and to withstand multiple photocopy runs.
- Normal weight must be readily distinguishable from bold.
- An alphabet intended for use as symbols need not be usable for continuous text; in fact, it is often desirable for a math alphabet to look a bit peculiar if used for text.

Implementation of the STIX glyphs was contracted out. The working list was a database in order of provisional ID; assignment of new Unicodes was still in the future. Glyphs were implemented in blocks, which were returned to the STIX Technical Review Committee for comments; any problem glyphs were returned to the contractor for repair.

The random ordering of the glyphs in the working list meant that glyphs intended to be used together, or supposed to be the same shape or weight, often weren't designed in the same batches, and weren't available for review at the same time. This meant that a final design review would be essential.

The random ordering also meant that the fonts couldn't yet be used for anything practical. Among other things, it was necessary to have a well-defined naming scheme. Because the fonts were delivered in Adobe Type 1 form, it was decided to assign glyph names according to the Adobe guidelines.

Except for a relatively small core of glyphs — essentially those representing the ASCII and Latin 1 blocks and some additional punctuation — the recommended form of a name was based on the Unicode, with extensions to indicate compounding or size and shape variations. This name begins with either "`uni`" or "`U`" for glyphs corresponding to characters in Unicode Plane 0 or Plane 1 respectively, or with "`stix`" for (the fewer than 256) glyphs with no corresponding Unicode.

8 Bookkeeping, bookkeeping

In order to keep track of what was happening, master tables or databases were maintained in several places. Tim Ingoldsby (of AIP, the overall project manager) started with the same database as used by the font contractor. To this he added, as phases were delivered, information about what glyphs were delivered in which phase, and the font and position in the font where each was located.

I maintained a list based on the original collection information, sorted by Unicode or provisional ID. This initially included sources, the names by which the sources refer to each glyph, the number of instances required (for weight, posture, size, etc.), and a glyph description. As new information became available, or was defined, it was added to the table:

- newly assigned Unicodes, with cross-references to and from the provisional ID;
- Type 1 glyph names;
- "TEX names", since several of the STIX organizations use that typesetting system;
- MathML entity names.

When delivery of the glyphs was nearing completion, Tim reprocessed my list, merged the differences into his database, and produced a file for checking. In this process we identified items that had been overlooked, and made a final list for completion of the deliveries.

That left only a few tasks:

- design review;
- shape and content corrections;
- packaging and user documentation;
- beta testing;
- (LA)TEX support;
- final coordination of MathML entity names.

9 The design review

One more rearrangement was necessary — organizing the glyphs into groups that reflected shape categories, irrespective of identifier value. Since alphabets are ordered logically within Unicode, they had already been reviewed and corrected, and it was not necessary to look at them again. The other categories included

- diacritics;
- punctuation;
- geometric shapes (circles, squares, diamonds and lozenges, triangles, other polygons);
- arrows;
- relations (equals, greater/less, sub/supersets, others);
- binary operators (cups/caps, and/or, plus/times, other);
- large operators (integrals, other);
- delimiters and fences;
- other shapes.

Within each category, glyphs were arranged by similarity of shape and size (Figure 3). Making sure that everything was accounted for involved one more sweep through the entire STIX master table. This turned up some residual errors, which were corrected so that the permanent documentation would be accurate.

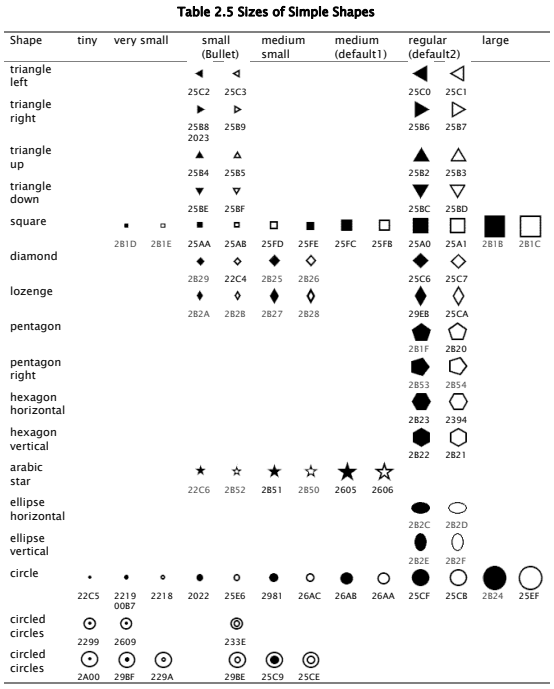

Figure 3: For geometric shapes, Unicode does make a distinction by size. From Unicode Technical Report #25 [1]; copyright Unicode, used by permission.

Figure 4: A proof sheet for the glyphs based on or related to equal signs.

For each group of symbols, proof sheets were generated (Figure 4), reviewed, and comments forwarded to the font specialist making the corrections. No category was accepted the first time around, but most required no more than two cycles for approval.

10 LaTeX support

For non-TeX use, the fonts will be delivered as OpenType. However, some of the STIX organizations are still using TeX implementations that don't even support the use of virtual fonts. For this reason, a set of Type 1 fonts will be provided as well, re-encoded to access the glyphs in 256-glyph chunks.

Most "TeX names" will not change for glyphs that were already available for TeX; Scott Pakin's comprehensive symbols list [2] has been an invaluable resource in the naming effort. For symbols that were not already generally available, new names have been assigned according to established naming principles, being careful to avoid conflicts with existing names. Actual coding of the LaTeX support package will be done by an experienced LaTeX programmer.

LaTeX support will not be included for the initial beta release, which is expected early in the Fall, but it should be available soon afterwards, and we anticipate that it will be ready to go by the time the fonts are posted for general release.

11 The future

We expect that a few more problems will be identified during beta testing, but in general, we believe that our efforts have resulted in a collection of fonts that will make it possible to represent nearly all mathematical expression both on paper and on computer screens. How this is actually done does depend on application developers, but since support of Unicode beyond just Plane 0 is beginning to be viewed as necessary by browser distributors, we are optimistic.

As we mentioned earlier, mathematical notation is open-ended. The mechanism for adding this notation to Unicode is now in place. The only question open is, how will new glyphs become part of these fonts. Presumably the STIX organizations will address that question after they've all had a well deserved rest.

References

[1] Barbara Beeton, Asmus Freytag and Murray Sargent III, Unicode Technical Report #25, Unicode Support for Mathematics, 2007. `http://www.unicode.org/reports/tr25`

[2] Scott Pakin, *The Comprehensive LaTeX Symbol List*, 2005. CTAN: `info/symbols/comprehensive`

[3] The Unicode Consortium, *The Unicode Standard, Version 2.0*, Addison-Wesley Developers Press, Reading, MA, 1996.

[4] The Unicode Consortium, *The Unicode Standard, Version 5.0*, edited by Julie D. Allen et al., Addison-Wesley, Upper Saddle River, NJ / Boston, 2006.