OTF math fonts

GUST e-foundry's workbench

Breskens, The Netherlands, 8–12 X 2012

Bogusław Jackowski

WARNING

Describing the whole process of an OTF math font creation in details would be as dull as ditchwater.

WARNING

Describing the whole process of an OTF math font creation in details would be as dull as ditchwater. Therefore, I'll give just a few less or more representative examples in hope that it will sufficiently illustrate the TEXnique we apply in the GUST e-foundry.

OTF Math font structure

According to “Unicode Support for Mathematics” (Draft Unicode Technical Report #25, by Barbara Beeton, Asmus Freytag, and Murray Sargent III) and a confidential Microsoft® document “The MATH table and OpenType Features for Math Processing”, an OTF math font should contain:

OTF Math font structure

According to “Unicode Support for Mathematics” (Draft Unicode Technical Report #25, by Barbara Beeton, Asmus Freytag, and Murray Sargent III) and a confidential Microsoft® document “The MATH table and OpenType Features for Math Processing”, an OTF math font should contain:

- glyphs falling into several groups of different kind: alphabetic glyphs, subdivided further into various type of alphabets – sans serif, calligraphic, double struck (aka blackboard bold), fraktur, and… typewriter (monospace), some of them including Greek letters (symbols), some – not:

ABC $\mathcal{ABC}$ $\mathbb{ABC}$ $\mathfrak{ABC}$ ABC $\boldsymbol{\alpha\beta\gamma}$

OTF Math font structure

According to “Unicode Support for Mathematics” (Draft Unicode Technical Report #25, by Barbara Beeton, Asmus Freytag, and Murray Sargent III) and a confidential Microsoft® document “The MATH table and OpenType Features for Math Processing”, an OTF math font should contain:

- glyphs falling into several groups of different kind: alphabetic glyphs, subdivided further into various type of alphabets – sans serif, calligraphic, double struck (aka blackboard bold), fraktur, and… typewriter (monospace), some of them including Greek letters (symbols), some – not:

  ABC $\mathcal{ABC}$ $\mathbb{ABC}$ $\mathfrak{ABC}$ ABC $\boldsymbol{\alpha\beta\gamma}$

- symbols (letter-like, math operators, elements of music notation, playing card suit symbols, etc.):

  אבגד ∫ ∬ ∭ ∮ ∯ ∰ ⨍ ⨏ ⨑ ⨒ ♯♮♭ ♤♠♡♥♢♦♧♣

OTF Math font structure

According to “Unicode Support for Mathematics” (Draft Unicode Technical Report #25, by Barbara Beeton, Asmus Freytag, and Murray Sargent III) and a confidential Microsoft® document “The MATH table and OpenType Features for Math Processing”, an OTF math font should contain:

- glyphs falling into several groups of different kind: alphabetic glyphs, subdivided further into various type of alphabets – sans serif, calligraphic, double struck (aka blackboard bold), fraktur, and… typewriter (monospace), some of them including Greek letters (symbols), some – not:

  ABC 𝒜ℬ𝒞 𝔸𝔹ℂ 𝔄𝔅ℭ 𝙰𝙱𝙲 **α β γ**

- symbols (letter-like, math operators, elements of music notation, playing card suit symbols, etc.):

  ℵ ∫ ∬ ∭ ⨌ ⨍ ⨎ ⨏ ♯ ♮ ♭ ♤ ♠ ♡ ♥ ♢ ♦ ♧ ♣

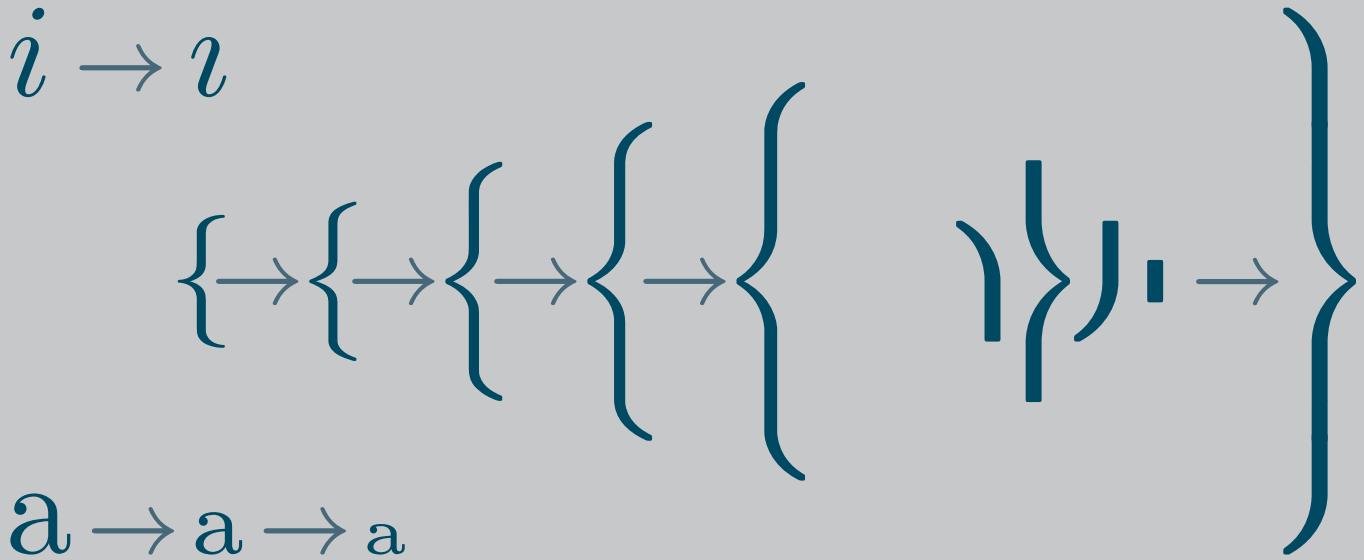
- other symbols (arrows, geometrical shapes, etc.):

  ↑ ↓ ↖ ↘ ↕ ↧ ↪ ↩ ↬ ↫ ⇑ ⇓ ⇖ ⇘ ⇐ ⇒ ▭ ▬ ▁ ▇ ■ □ ■ ▫ ▪ ⬚

OTF Math font structure

According to “Unicode Support for Mathematics” (Draft Unicode Technical Report #25, by Barbara Beeton, Asmus Freytag, and Murray Sargent III) and a confidential Microsoft® document “The MATH table and OpenType Features for Math Processing”, an OTF math font should contain:

- structural information, governed by so called “math table”, linking certain glyphs together, e.g., *dotless i* and *j* and their dotted counterparts, chains of glyphs of increasing size, groups of glyphs to be assembled into one symbol, normal–subscript–subsubscript, etc.



OTF Math font structure

According to “Unicode Support for Mathematics” (Draft Unicode Technical Report #25, by Barbara Beeton, Asmus Freytag, and Murray Sargent III) and a confidential Microsoft® document “The MATH table and OpenType Features for Math Processing”, an OTF math font should contain:

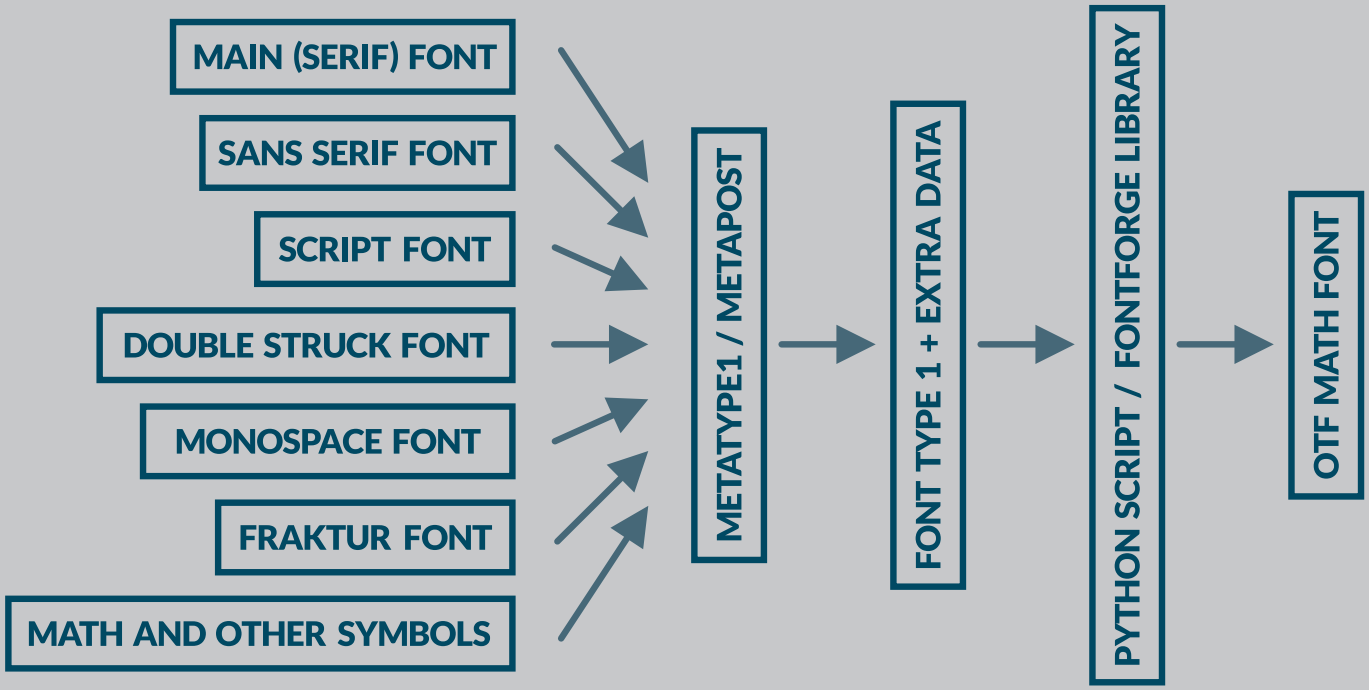
- structural information, governed by so called “math table”, linking certain glyphs together, e.g., *dotless i* and *j* and their dotted counterparts, chains of glyphs of increasing size, groups of glyphs to be assembled into one symbol, normal–subscript–subsubscript, etc.
- metric data: (1) basic metric data (glyph dimens, kerning, italic correction, etc. – very much like in TEX TFMs) and (2) extended (math) metric data (there are a few dozen of relevant parameters, e.g., parameters that control the position of sub- and superscripts, the position of numerator and denominator with respect to the rule in fractions, etc. – again, very much like in TEX math fonts).

TEXnology

General technological line is as follows: prepare a uniform METATYPE1 source of the whole font, add structural data (at the level of METAPOST code), generate PostScript Type 1 font and an auxiliary file containing the structural information and then, using a command line Python tool with FontForge library, generate the final OTF math font. Up to now, three fonts have been generated using this technology: Latin Modern Math, TG Pagella Math, and TG Termes Math.

TEXnology

General technological line is as follows: prepare a uniform METATYPE1 source of the whole font, add structural data (at the level of METAPOST code), generate POSTSCRIPT Type 1 font and an auxiliary file containing the structural information and then, using a command line Python tool with FontForge library, generate the final OTF math font. Up to now, three fonts have been generated using this technology: Latin Modern Math, TG Pagella Math, and TG Termes Math.



Preliminary phase – collecting alphabets

Most troublesome and time-consuming is the preliminary stage. The variety of different sources of different origins necessitates manual checking and enhancing of most of the source stuff.

Preliminary phase – collecting alphabets

Most troublesome and time-consuming is the preliminary stage. The variety of different sources of different origins necessitates manual checking and enhancing of most of the source stuff.

The main source was, of course, the basic serif font. In our case, Latin Modern, TG Pagella and TG Termes were used (all three prepared by the GUST e-foundry). Additional alphabets were taken from various sources.

Preliminary phase – collecting alphabets

Most troublesome and time-consuming is the preliminary stage. The variety of different sources of different origins necessitates manual checking and enhancing of most of the source stuff.

The main source was, of course, the basic serif font. In our case, Latin Modern, TG Pagella and TG Termes were used (all three prepared by the GUST e-foundry). Additional alphabets were taken from various sources.

CALLIGRAPHY

Latin Modern: not Knuthian (sorry to say, ugly); instead, slanted Euler was used (*eusm* and *eusb*), no lowercase.

TG Pagella: Odstemplik (by courtesy gluksza@wp.pl).

TG Termes: our own drawing.

Preliminary phase – collecting alphabets

Most troublesome and time-consuming is the preliminary stage. The variety of different sources of different origins necessitates manual checking and enhancing of most of the source stuff.

The main source was, of course, the basic serif font. In our case, Latin Modern, TG Pagella and TG Termes were used (all three prepared by the GUST e-foundry). Additional alphabets were taken from various sources.

DOUBLE STRUCK

Latin Modern: Alan Jeffrey's blackboard bold.

TG Pagella: our own drawing.

TG Termes: our own drawing.

Preliminary phase – collecting alphabets

Most troublesome and time-consuming is the preliminary stage. The variety of different sources of different origins necessitates manual checking and enhancing of most of the source stuff.

The main source was, of course, the basic serif font. In our case, Latin Modern, TG Pagella and TG Termes were used (all three prepared by the GUST e-foundry). Additional alphabets were taken from various sources.

GREEK SERIF LOWERCASE

Latin Modern: Knuthian (upright – unslanted, the relevant sidebearings were calculated from CM sources for *slant* = 0).

$\alpha\ \boldsymbol{\alpha}\ \alpha\ \boldsymbol{\alpha}\ \beta\ \boldsymbol{\beta}\ \beta\ \boldsymbol{\beta}\ \gamma\ \boldsymbol{\gamma}\ \gamma\ \boldsymbol{\gamma}$

TG Pagella: Math Pazo (courtesy of Diego Puga; upright, as above – just unslanted and, in a few cases, manually tuned).

$\alpha\ \boldsymbol{\alpha}\ \alpha\ \boldsymbol{\alpha}\ \beta\ \boldsymbol{\beta}\ \beta\ \boldsymbol{\beta}\ \gamma\ \boldsymbol{\gamma}\ \gamma\ \boldsymbol{\gamma}$

TG Termes: our own drawing (based on TG Termes text fonts).

$\alpha\ \boldsymbol{\alpha}\ \alpha\ \boldsymbol{\alpha}\ \beta\ \boldsymbol{\beta}\ \beta\ \boldsymbol{\beta}\ \gamma\ \boldsymbol{\gamma}\ \gamma\ \boldsymbol{\gamma}$

Preliminary phase – collecting alphabets

Most troublesome and time-consuming is the preliminary stage. The variety of different sources of different origins necessitates manual checking and enhancing of most of the source stuff.

The main source was, of course, the basic serif font. In our case, Latin Modern, TG Pagella and TG Termes were used (all three prepared by the GUST e-foundry). Additional alphabets were taken from various sources.

GREEK SANS SERIF BOLD LOWERCASE

Latin Modern: based on Knuthian sources (manually tuned).

$\boldsymbol{\alpha}\ \boldsymbol{\alpha}\ \boldsymbol{\beta}\ \boldsymbol{\beta}\ \boldsymbol{\gamma}\ \boldsymbol{\gamma}$

TG Pagella: taken from sans serif DejaVu.

$\boldsymbol{\alpha}\ \boldsymbol{\alpha}\ \boldsymbol{\beta}\ \boldsymbol{\beta}\ \boldsymbol{\gamma}\ \boldsymbol{\gamma}$

TG Termes: our own drawing (corrected TG Hermes bold).

$\boldsymbol{\alpha}\ \boldsymbol{\alpha}\ \boldsymbol{\beta}\ \boldsymbol{\beta}\ \boldsymbol{\gamma}\ \boldsymbol{\gamma}$

Preliminary phase – collecting alphabets

Most troublesome and time-consuming is the preliminary stage. The variety of different sources of different origins necessitates manual checking and enhancing of most of the source stuff.

The main source was, of course, the basic serif font. In our case, Latin Modern, TG Pagella and TG Termes were used (all three prepared by the GUST e-foundry). Additional alphabets were taken from various sources.

HEBREW (FOUR LETTERS)

Latin Modern: Knuthian *alef* plus our own drawing.
א ב ג ד

TG Pagella: our own drawing.
א ב ג ד

TG Termes: our own drawing.
א ב ג ד

Preliminary phase – collecting alphabets

Most troublesome and time-consuming is the preliminary stage. The variety of different sources of different origins necessitates manual checking and enhancing of most of the source stuff.

The main source was, of course, the basic serif font. In our case, Latin Modern, TG Pagella and TG Termes were used (all three prepared by the GUST e-foundry). Additional alphabets were taken from various sources.

FRAKTUR

Latin Modern: Euler (Hermann Zapf's design).

𝔄 𝔅 ℭ 𝔄 𝔅 ℭ 𝔞 𝔟 𝔠 𝔞 𝔟 𝔠

TG Pagella: again Euler (tentative?).

𝔄 𝔅 ℭ 𝔄 𝔅 ℭ 𝔞 𝔟 𝔠 𝔞 𝔟 𝔠

TG Termes: Leipziger Fraktur (courtesy of Peter Wiegel).

𝔄 𝔅 ℭ 𝔄 𝔅 ℭ 𝔞 𝔟 𝔠 𝔞 𝔟 𝔠

Example 1 – Latin Modern sans serif bold Greek

We wanted to use Knuth's Metafont code for generating the outlines. We hoped that using the *cmssbx10* parameters would yield proper shapes.

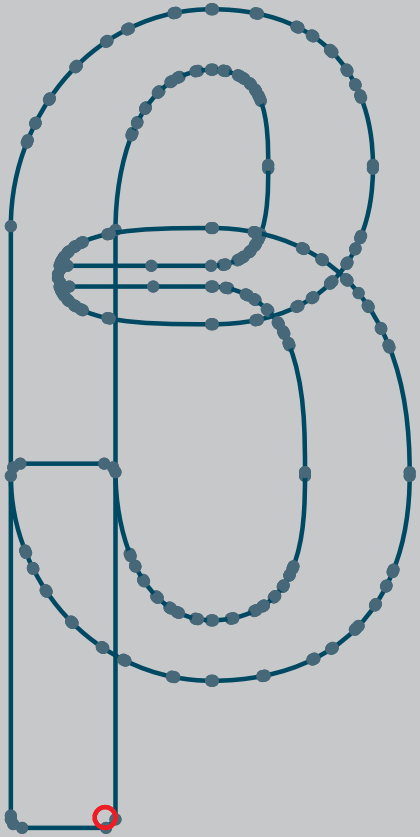
Example 1 – Latin Modern sans serif bold Greek

We wanted to use Knuth's Metafont code for generating the outlines. We hoped that using the *cmssbx10* parameters would yield proper shapes.

Alas, Greek glyphs apparently were not "tuned" for sans serif, especially bold, and in many cases "optical" tuning turned out to be necessary.

Example 1 – Latin Modern sans serif bold Greek

We wanted to use Knuth's Metafont code for generating the outlines. We hoped that using the *cmssbx10* parameters would yield proper shapes.



We wanted to employ Luigi Scarso's MFLua for finding the outlines. Alas, MFLua finds "raw" outlines, i.e., the envelope created by polygonal pen's vertices; as a result, outlines have too many many nodes; their number could be perhaps automatically reduced, but we finally abandoned that approach.

Example 1 – Latin Modern sans serif bold Greek

We wanted to use Knuth's Metafont code for generating the outlines. We hoped that using the *cmssbx10* parameters would yield proper shapes.

All in all, we retrieved the basic paths (i.e., without pen stroking) from Computer Modern sources, we modified them interactively, then we "expanded strokes" (pen diameters were known from sources), and, finally, we again slightly tuned the resulting outlines manually.

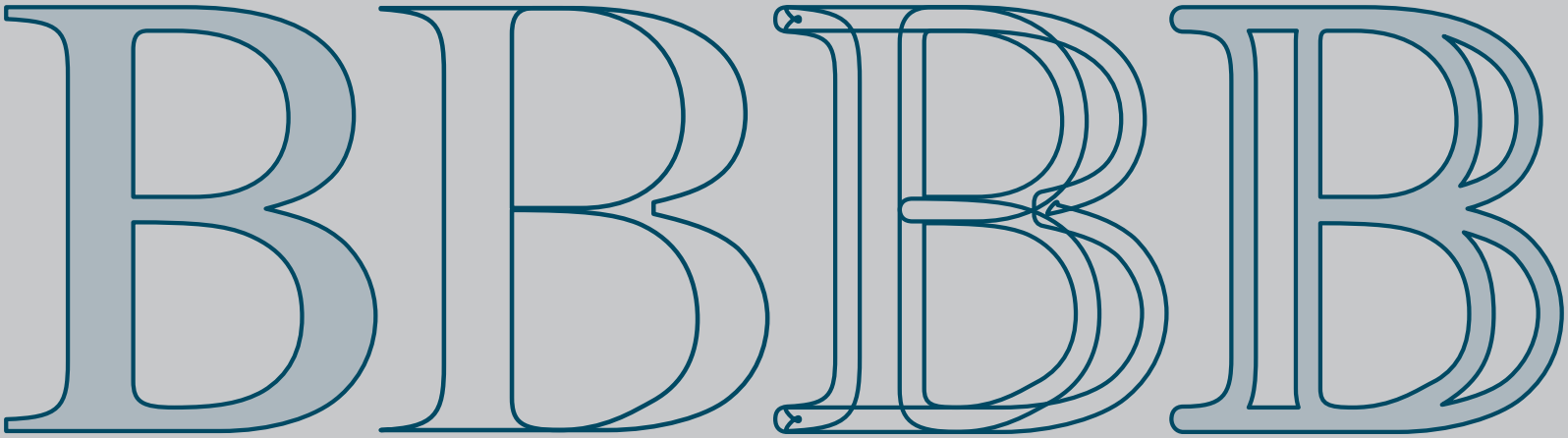
Example 2 – TG Termes double struck

We didn’t want to have identical double struck variants in all our fonts. We took Alan Jeffrey’s time-honoured blackboard bold for Latin Modern, but for Termes and Pagella we decided to prepare our own optically matching double struck alphabets.

Example 2 – TG Termes double struck

We didn't want to have identical double struck variants in all our fonts. We took Alan Jeffrey's time-honoured blackboard bold for Latin Modern, but for Termes and Pagella we decided to prepare our own optically matching double struck alphabets.

This turned out to be a fairly simple although somewhat time-consuming task; again, of course, an interactive tool was used.



The example is excerpted from the TG Termes Math font.

The main phase: preparing symbols

A lion share (circa one quarter of the repertoire) was programmed from scratch, in that number mathematical operator and relation symbols, arrows, geometrical shapes, etc. Needless to say, this stage is for us most interesting.

The main phase: preparing symbols

A lion share (circa one quarter of the repertoire) was programmed from scratch, in that number mathematical operator and relation symbols, arrows, geometrical shapes, etc. Needless to say, this stage is for us most interesting.

We didn't insist on having exactly the same sources in all three cases: to manipulate parameters is not flexible enough to control glyph shapes (even as simple as math symbols) in all details. Thus, we accepted slight modification of sources: each font has its own "base" macros (built on the top of a "base" common to all fonts). Having finished the TG math project, we plan to enhance the universal base by macros that we've developed in the meantime.

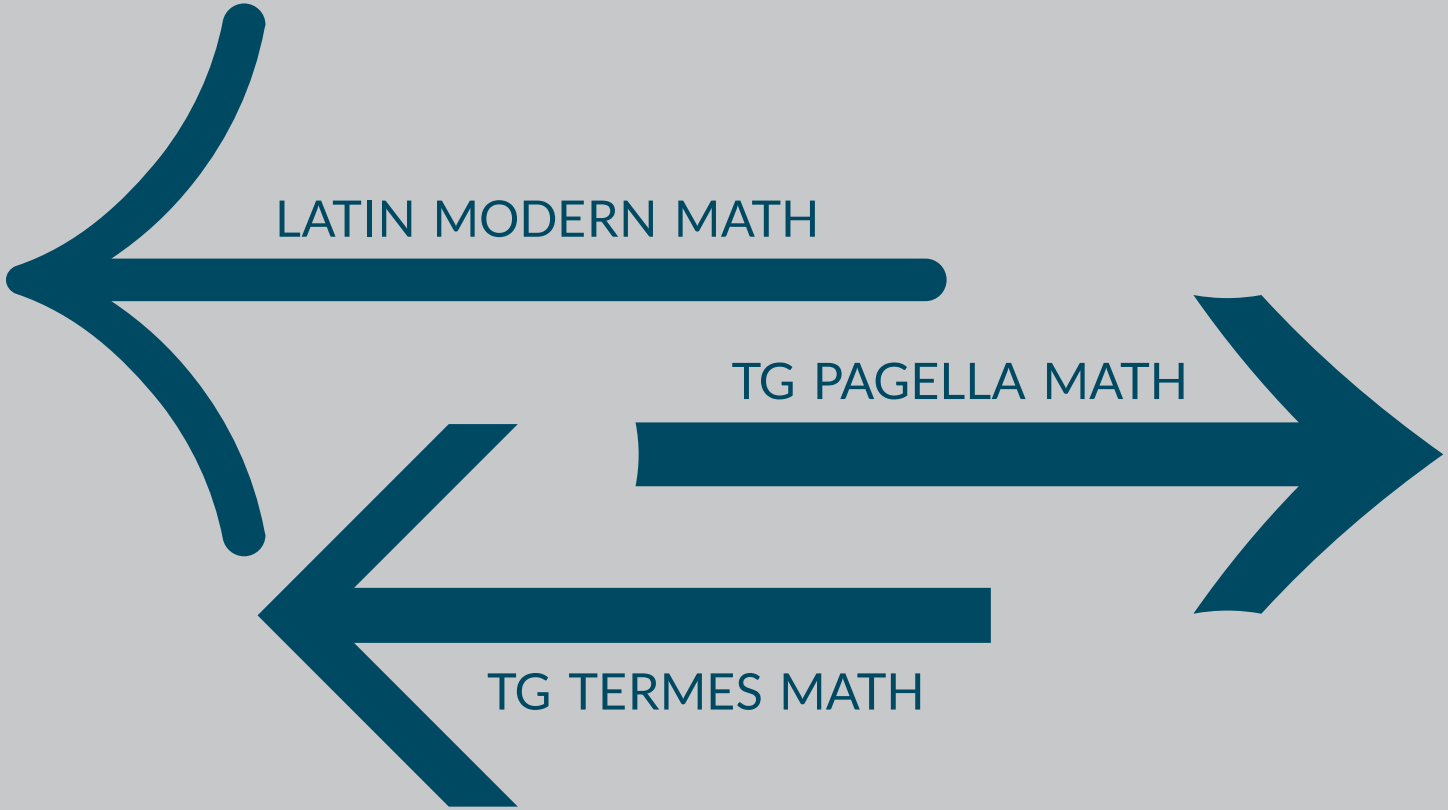
The main phase: preparing symbols

A lion share (circa one quarter of the repertoire) was programmed from scratch, in that number mathematical operator and relation symbols, arrows, geometrical shapes, etc. Needless to say, this stage is for us most interesting.

LATIN MODERN MATH TG PAGELLA MATH TG TERMES MATH

The main phase: preparing symbols

A lion share (circa one quarter of the repertoire) was programmed from scratch, in that number mathematical operator and relation symbols, arrows, geometrical shapes, etc. Needless to say, this stage is for us most interesting.



Dangling stage: documentation

One of our major concern is still missing up-to-date publicly available documentation of the code and of the fonts. Of course, this does not mean that we are not documenting our work – without lots of scraps of documentation we would be hopelessly lost. Although bringing the scraps to a publication form is by no means a trifle.

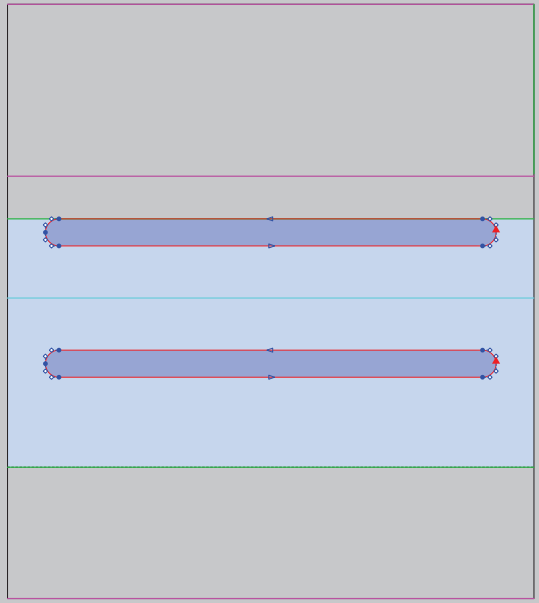
Dangling stage: documentation

One of our major concern is still missing up-to-date publicly available documentation of the code and of the fonts. Of course, this does not mean that we are not documenting our work – without lots of scraps of documentation we would be hopelessly lost. Although bringing the scraps to a publication form is by no means a trifle.

Our basic tool is, of course, METATYPE1 with its preview facility, which (due to the possibility of using stencils in the background) was particularly important in programming math symbols for Latin Modern Math.

Dangling stage: documentation

equal: uni003D

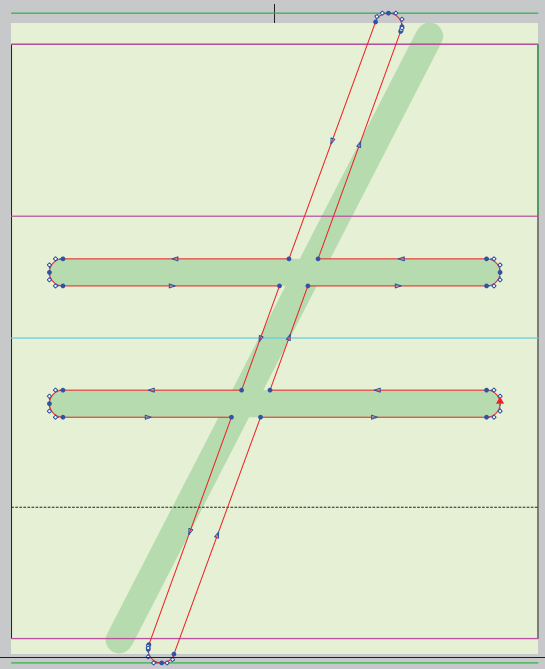


equal:003D

```
unicode_introduce("equal");
beginglyph(equal);
  save p, g; path p[];
  g := 1/2 (math_gap + math_rule);
  p0 := apt_minus(math_axis + g);
  p1 := apt_minus(math_axis − g);
  wd.glyph_name := adjust_width(p0, p1)();
  Fill p0, p1;
  nosb_hsbw;
endglyph;
```

Dangling stage: documentation

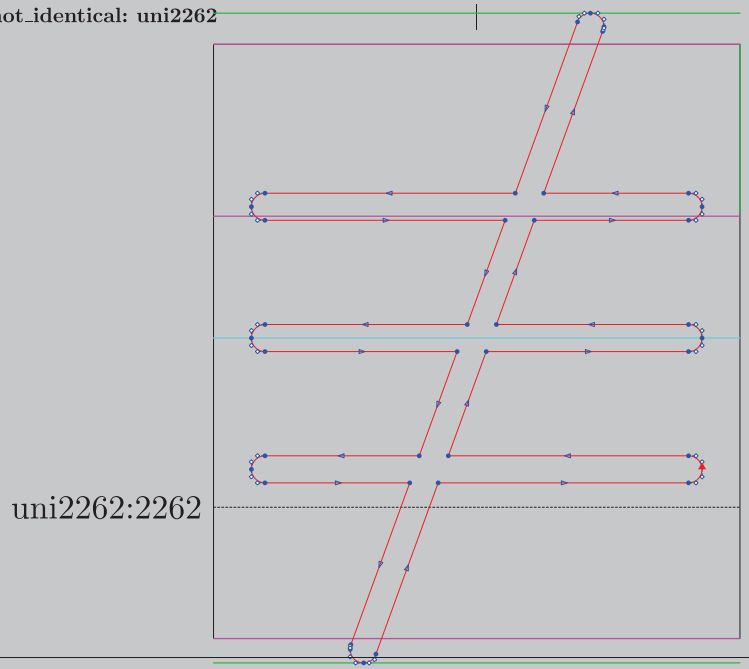
not_equal: uni2260




```
unicode_introduce("not_equal");
beginglyph(not_equal);
  save p; path p[];
  p0 := glyph_stored.equal 1;
  p1 := glyph_stored.equal 2;
  p2 := gen_solidus(slash_angle, slash_height);
  p2 := p2 shifted (CCx(p0) − CCx(p2), 0);
  p3 := blend_paths(p0, blend_paths(p1, p2));
  wd.glyph_name := adjust_width(p3)();
  Fill p3;
  nosb_hsbw;
endglyph;
```

Dangling stage: documentation

not_identical: uni2262

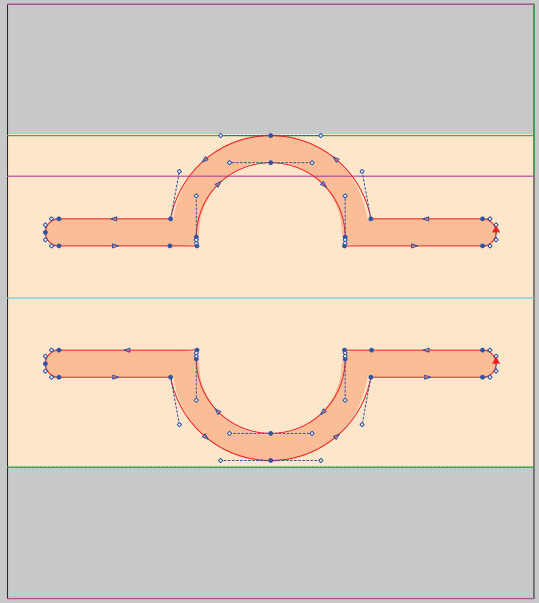



```
unicode_introduce("not_identical");
beginglyph(not_identical);
  save h, p; path p[];
  p1 = glyph_stored.identical 1;
  p2 = glyph_stored.identical 2;
  p3 = glyph_stored.identical 3;
  h = URy(p1, p1, p3) − LLy(p1, p1, p3) + 2math_rule;
  if h < slash_height: h := slash_height; fi
  p4 = gen_solidus(slash_angle, h);
  p4 := p4 shifted (CCx(p1) − CCx(p4), 0);
  p5 = blend_paths(p1, blend_paths(p2, blend_paths(p3, p4)));
  wd.glyph_name := wd.identical;
  Fill p5;
  nosb_hsbw;
endglyph;
```

Dangling stage: documentation

10	LMM--REL	*21:18 6 X 2012*

geometrically_equivalent: uni224E



uni224E:224E

```
unicode_introduce("geometrically_equivalent");
beginglyph(geometrically_equivalent);
  save o, p; path p[];
  p1 := glyph_stored.equal 1;
  p2 := glyph_stored.equal 2;
  local math_stem' = math_rule; apt_ring_o(); endlocal
  z1 = center(p1) + 1/2 math_rule * down;
  z2 = center(p2) + 1/2 math_rule * up;
  z3 = (0, 1/3 math_rule);
  o3 := cut_single_right(o1 shifted z3, left, right) shifted z1; o4 := o2 shifted z3 shifted z1;
  o5 := cut_single_left(o1 shifted -z3, left, right) shifted z2; o6 := o2 shifted -z3 shifted z2;
  p3 := blend_paths(blend_paths(p1, o3), o4);
  p4 := blend_paths(blend_paths(p2, o5), o6);
  wd.glyph_name := adjust_width(p3, p4)();
  Fill p3, p4;
  nosb_hsbw;
endglyph;
```

Dangling stage: documentation

One of our major concern is still missing up-to-date publicly available documentation of the code and of the fonts. Of course, this does not mean that we are not documenting our work – without lots of scraps of documentation we would be hopelessly lost. Although bringing the scraps to a publication form is by no means a trifle.

Our basic tool is, of course, METATYPE1 with its preview facility, which (due to the possibility of using stencils in the background) was particularly important in programming math symbols for Latin Modern Math.

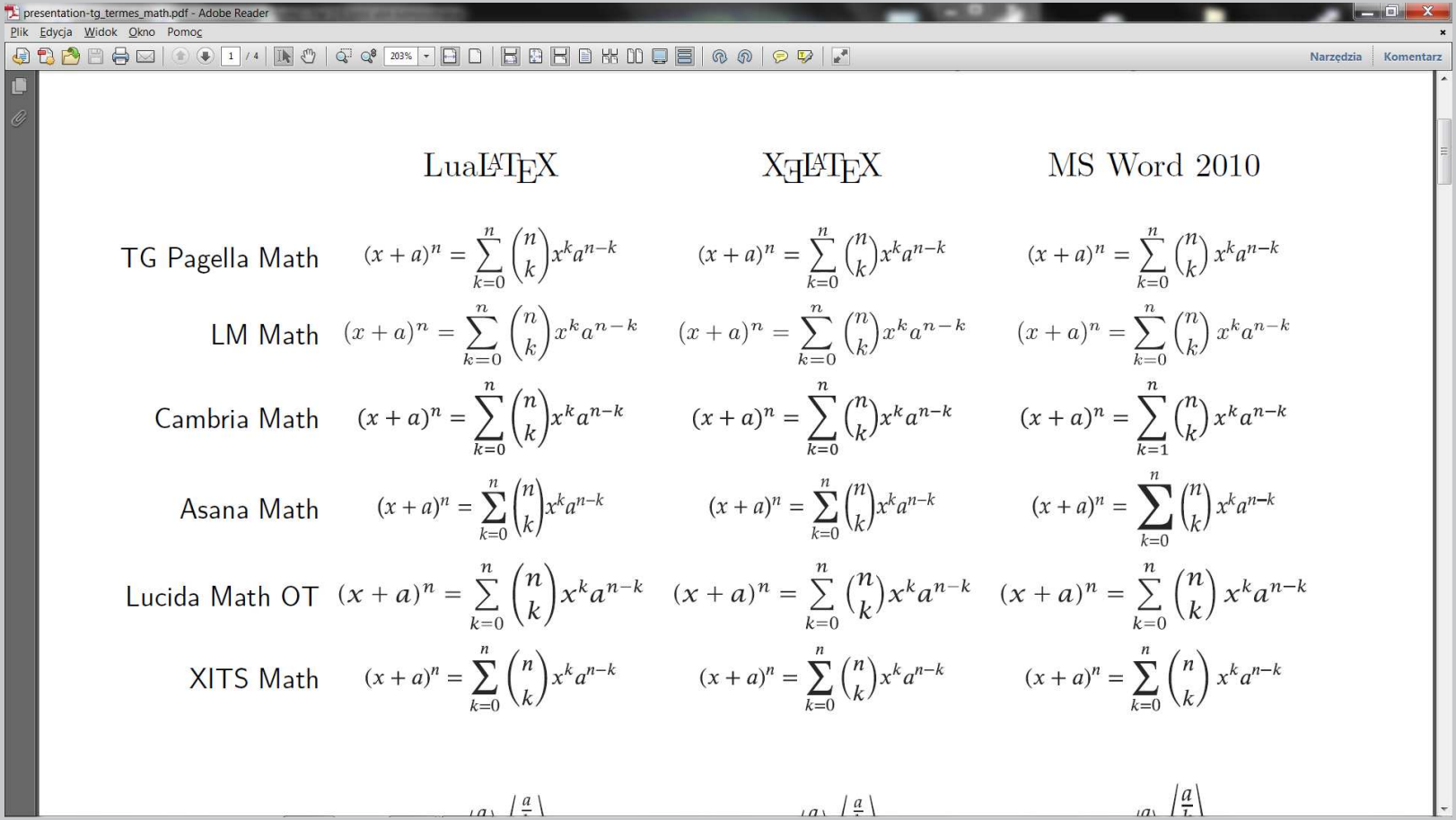
Dangling stage: documentation

One of our major concern is still missing up-to-date publicly available documentation of the code and of the fonts. Of course, this does not mean that we are not documenting our work – without lots of scraps of documentation we would be hopelessly lost. Although bringing the scraps to a publication form is by no means a trifle.

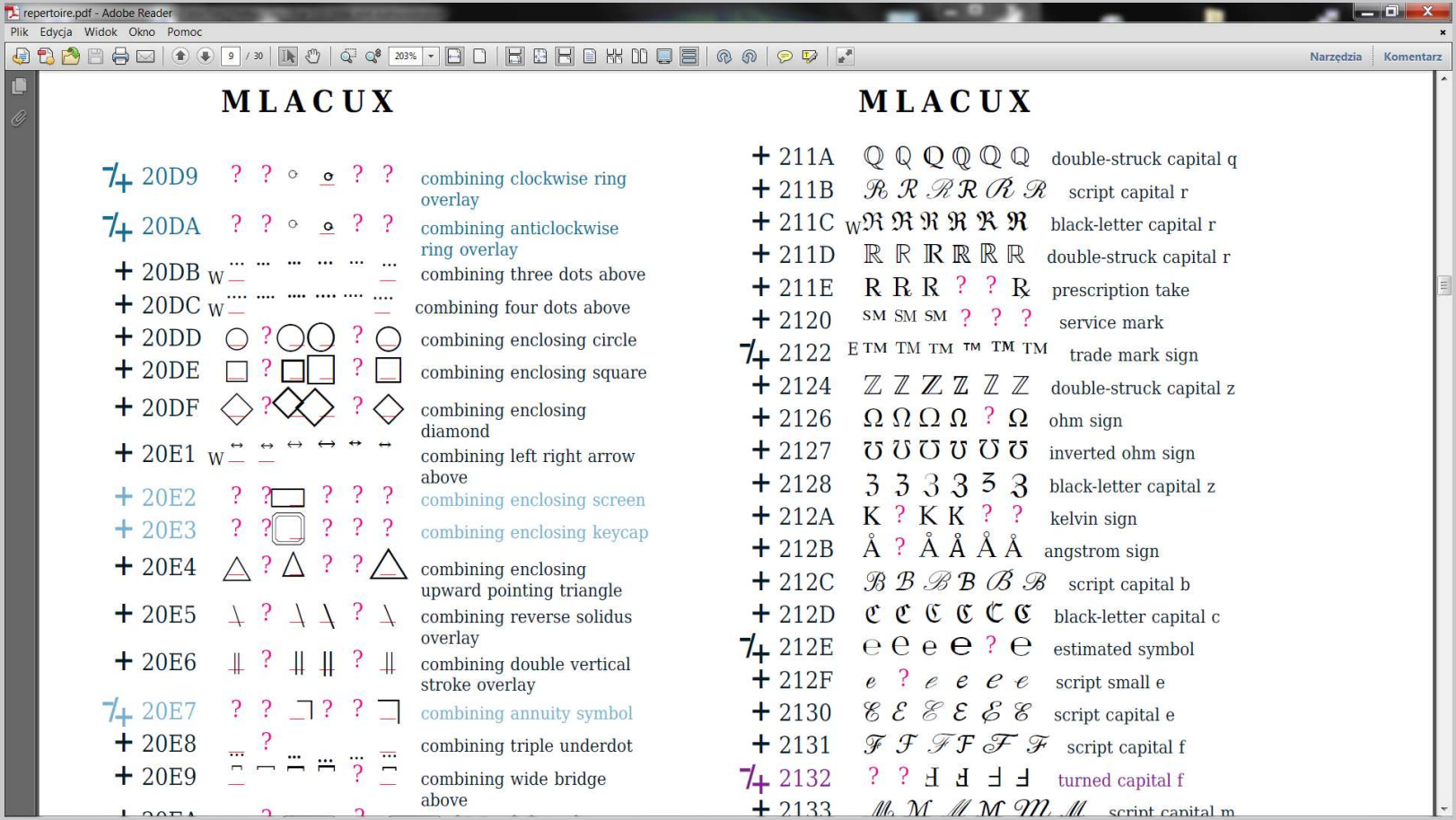
Our basic tool is, of course, METATYPE1 with its preview facility, which (due to the possibility of using stencils in the background) was particularly important in programming math symbols for Latin Modern Math.

Fonts, at every stage, are checked and compared with other fonts using various engines including luaTEX, XETEX, FontForge, MS Word® at al.

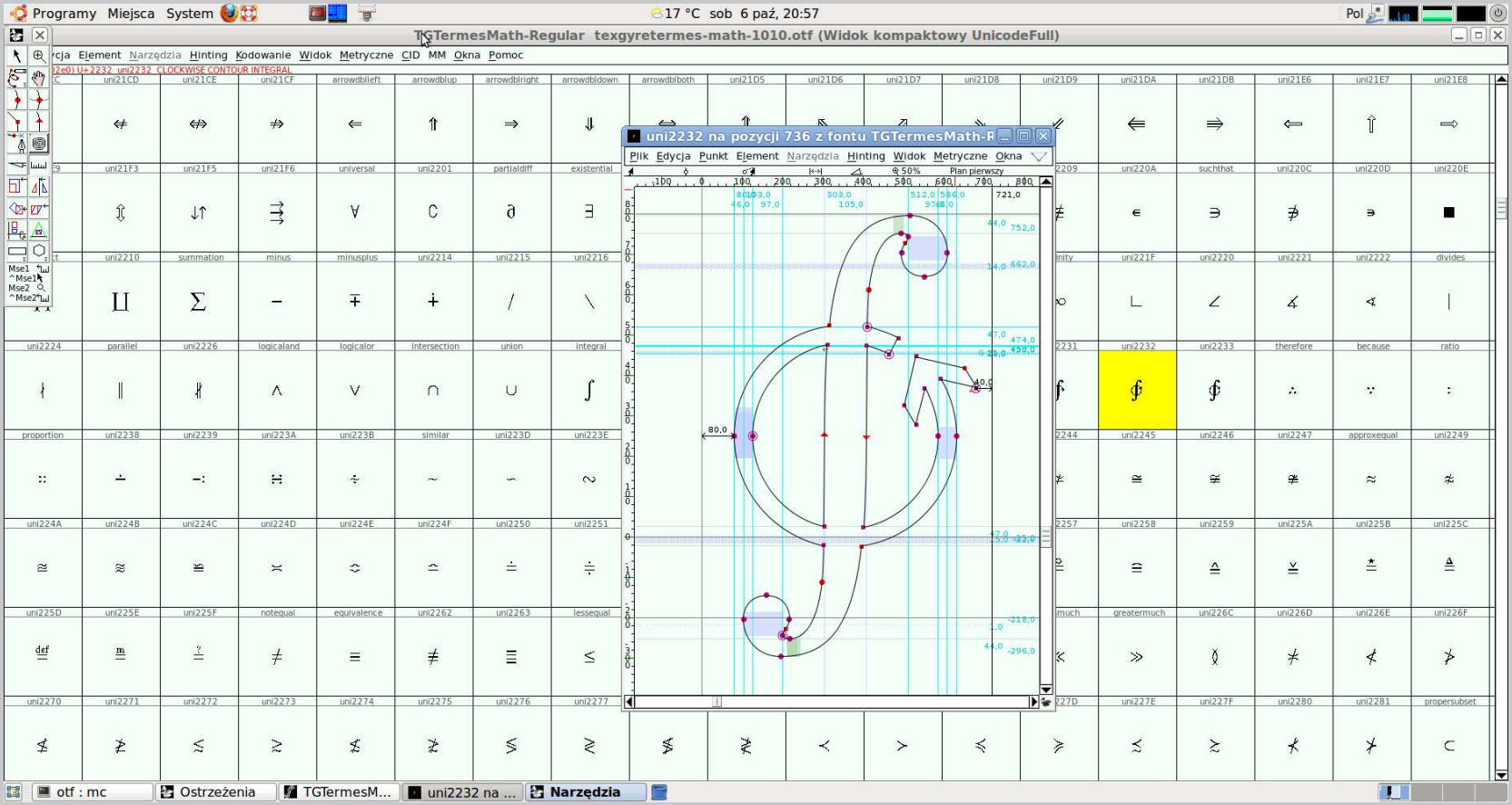
Dangling stage: documentation



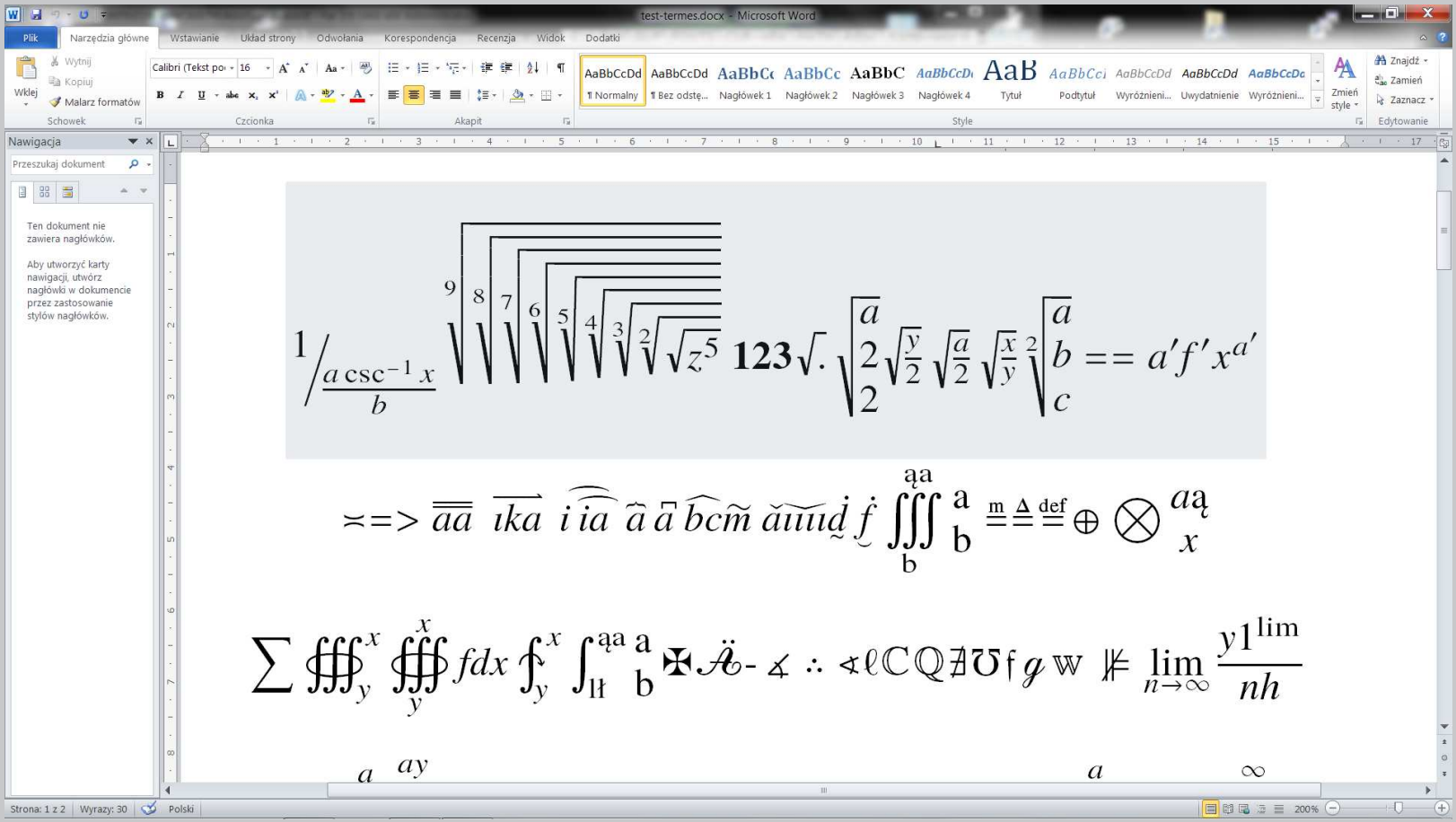
Dangling stage: documentation



Dangling stage: documentation



Dangling stage: documentation



Dangling stage: documentation

What should the comprehensive and legible documentation of the fonts and the engine for generating them look like?

Dangling stage: documentation

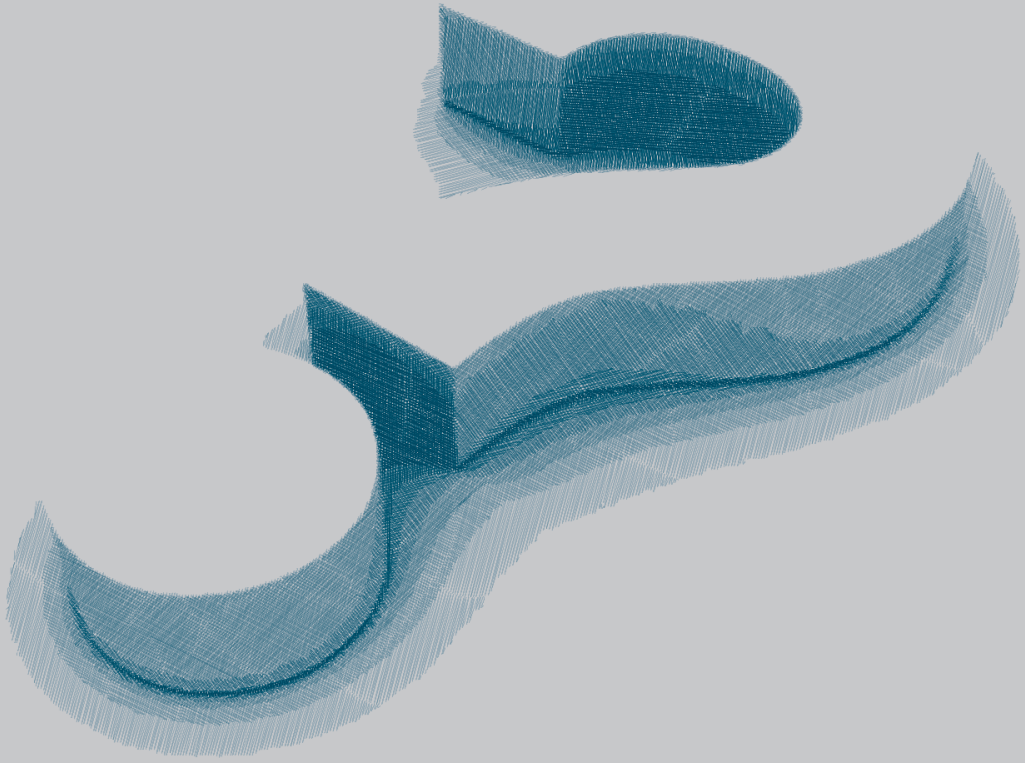
What should the comprehensive and legible documentation of the fonts and the engine for generating them look like?

The answer is simple:

Dangling stage: documentation

What should the comprehensive and legible documentation of the fonts and the engine for generating them look like?

The answer is simple:



The future

The future

- We've handed out the current versions of the three fonts to the organizers of the conference. We consider them "the eleventh hour version". Once we are back home, we will put them onto the GUST web page. After a month or so of a "grace period", we are going to release them oficially.

The future

- We've handed out the current versions of the three fonts to the organizers of the conference. We consider them "the eleventh hour version". Once we are back home, we will put them onto the GUST web page. After a month or so of a "grace period", we are going to release them oficially.
- For the next year, we plan to release two more TG fonts: TG Bonum Math and TG Schola Math.

The future

- We've handed out the current versions of the three fonts to the organizers of the conference. We consider them "the eleventh hour version". Once we are back home, we will put them onto the GUST web page. After a month or so of a "grace period", we are going to release them oficially.
- For the next year, we plan to release two more TG fonts: TG Bonum Math and TG Schola Math.
- The font projects to be carried out in the far future depend to a large extent on the result of the font meeting that is planned today before lunch.

The OpenType math fonts project is supported
by TEX Users Groups, in particular, by the Czechoslovak
TEX Users Group CSTUG, the German-speaking
TEX Users Group DANTE e.V., the Polish TEX Users Group
GUST, the Dutch-speaking TEX Users Group NTG,
TUG India, UK-TUG, and – last but not least – TUG.

THANK YOU FOR YOUR ATTENTION

and

LET'S MEET AT BACHOTEX AGAIN


Homa Gold
MARGARINE 500 g