

## GUST's e-foundry font projects

### *Closing report for 2019–2020*

Jerzy B. Ludwichowski  
Jerzy.Ludwichowski@gust.org.pl

DANTE e.V., Frühjahrstagung 2021



## About this presentation

The report proper<sup>1</sup> (mere facts) will be published in one of the forthcoming issues of the DANTE T<sub>E</sub>Xnische Komödie and, most probably, also in other LUG's journals.

This presentation, although reporting on what was achieved, is totally different. It aims to bring forth those elements of the e-foundry's projects which were difficult to place in a mere facts report.

I'll try to show the achievements and hope that it transpires how much effort went into the project(s) ...

---

<sup>1</sup>For the previous report see B. Jackowski et al. in DTK, Heft 3, 2018



## Project goals

The goals of the “new” e-foundry's projects (those emphasized are not yet done):

- ▶ defining math symbols subset(s) for various uses,
- ▶ a sans-serif Math Open Type Format font,
- ▶ a heavy Math Open Type Format font,
- ▶ a monospace font with math symbols for text editors,
- ▶ enhancing the T<sub>E</sub>X Gyre text fonts (all except for Chorus),
- ▶ maintenance,
- ▶ enhancements to existing math fonts.



## A tiny bit of history

Conceived and presented to LUG's in 2015, the project officially started in 2017, though work began already in 2016.

Externally visible progress was being made until 2018, when it finally transpired that the long overdue revamping of MetaType 1, the e-foundry's toolset, could not wait.

Work on the successor, Algotype started in 2018. In the midst of that effort Piotr Strzelczyk left the team. Luckily, Marek Ryćko agreed to step in. However, quite some time was lost.

Further loss of time was caused by the pandemic and, on top of it, Bogusław Jackowski's hospitalisation for both COVID-19 and a heart surgery.



## Math symbol subsets for TG text fonts

The following four slides show samples of the extended repertoire. Over 850 mathematical, geometrical and technical symbols were selected and added.

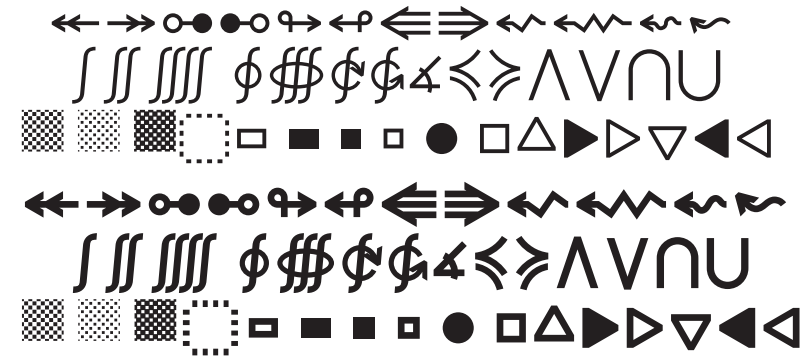
Please note:

- ▶ shown fonts were made with Algotype, the new engine,
- ▶ samples compiled with LuaT<sub>E</sub>X and OTF fonts,
- ▶ shown are examples only for regular (top) and bold (bottom) variants,
- ▶ TG Adventor and TG Pagella were previously (before Alogotype) extended and now “revised”,
- ▶ TG Schola and TG Termes are newly extended.



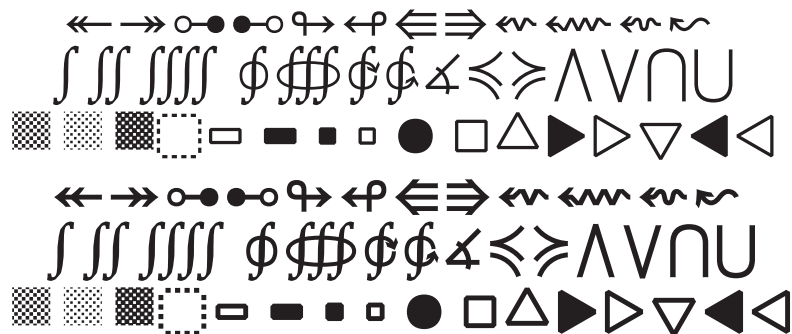
## TG Adventor (revised)

### TG Adventor



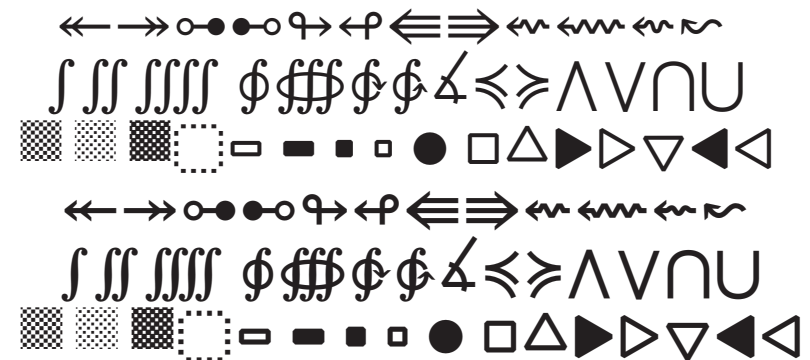
## TG Pagella (revised)

### TG Pagella



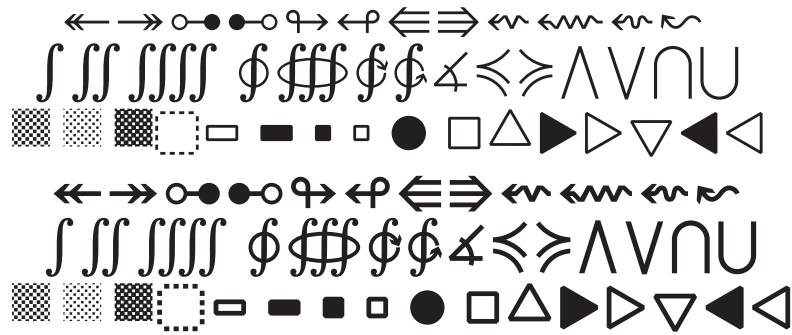
## TG Schola (newly extended)

### TG Schola



## TG Termes (newly extended)

### TG Termes



## Backward compatible math style

The following four slides show samples of the `ss10` feature — “backward compatible math style” — in action.

Please note:

- ▶ shown fonts were made with Algotype, the new engine,
- ▶ samples (formulas) compiled with LuaTeX and OTF fonts,
- ▶ TG Adventor and TG Pagella — revised (remade) from v. 2.501,
- ▶ TG Schola and TG Termes — newly made,
- ▶ lines marked `ss10-`, the default(!) are, hopefully, the better ones, but
- ▶ this is not to say that going `ss10+` will not produce good results in text mode ...



## TG Adventor (revised)

### TG Adventor

<code>ss10-</code>	$f(x)=1/x$	$(x+1)(x-3) > 0$
<code>ss10+</code>	$f(x)=1/x$	$(x+1)(x-3) > 0$
<code>ss10-</code>	<b><math>f(x)=1/x</math></b>	<b><math>(x+1)(x-3) &gt; 0</math></b>
<code>ss10+</code>	<b><math>f(x)=1/x</math></b>	<b><math>(x+1)(x-3) &gt; 0</math></b>
<code>ss10-</code>	$f(x)=1/x$	$(x+1)(x-3) > 0$
<code>ss10+</code>	$f(x)=1/x$	$(x+1)(x-3) > 0$
<code>ss10-</code>	<b><math>f(x)=1/x</math></b>	<b><math>(x+1)(x-3) &gt; 0</math></b>
<code>ss10+</code>	<b><math>f(x)=1/x</math></b>	<b><math>(x+1)(x-3) &gt; 0</math></b>



## TG Pagella (revised)

### TG Pagella

<code>ss10-</code>	$f(x)=1/x$	$(x+1)(x-3) > 0$
<code>ss10+</code>	$f(x)=1/x$	$(x+1)(x-3) > 0$
<code>ss10-</code>	<b><math>f(x)=1/x</math></b>	<b><math>(x+1)(x-3) &gt; 0</math></b>
<code>ss10+</code>	<b><math>f(x)=1/x</math></b>	<b><math>(x+1)(x-3) &gt; 0</math></b>
<code>ss10-</code>	$f(x)=1/x$	$(x+1)(x-3) > 0$
<code>ss10+</code>	$f(x)=1/x$	$(x+1)(x-3) > 0$
<code>ss10-</code>	<b><math>f(x)=1/x</math></b>	<b><math>(x+1)(x-3) &gt; 0</math></b>
<code>ss10+</code>	<b><math>f(x)=1/x</math></b>	<b><math>(x+1)(x-3) &gt; 0</math></b>



### TG Schola

ss10-  $f(\mathbf{x})=1/\mathbf{x} \quad (\mathbf{x}+1)(\mathbf{x}-3) > 0$   
 ss10+  $f(\mathbf{x})=1/\mathbf{x} \quad (\mathbf{x}+1)(\mathbf{x}-3) > 0$   
 ss10-  $\mathbf{f}(\mathbf{x})=\mathbf{1}/\mathbf{x} \quad (\mathbf{x}+1)(\mathbf{x}-3) > \mathbf{0}$   
 ss10+  $\mathbf{f}(\mathbf{x})=\mathbf{1}/\mathbf{x} \quad (\mathbf{x}+1)(\mathbf{x}-3) > \mathbf{0}$   
 ss10-  $f(x)=1/x \quad (x+1)(x-3) > 0$   
 ss10+  $f(x)=1/x \quad (x+1)(x-3) > 0$   
 ss10-  $\mathbf{f}(x)=\mathbf{1}/x \quad (x+1)(x-3) > \mathbf{0}$   
 ss10+  $\mathbf{f}(x)=\mathbf{1}/x \quad (x+1)(x-3) > \mathbf{0}$



### TG Termes

ss10-  $f(x)=1/x \quad (x+1)(x-3) > 0$   
 ss10+  $f(x)=1/x \quad (x+1)(x-3) > 0$   
 ss10-  $\mathbf{f}(x)=\mathbf{1}/x \quad (\mathbf{x}+1)(\mathbf{x}-3) > \mathbf{0}$   
 ss10+  $\mathbf{f}(x)=\mathbf{1}/x \quad (\mathbf{x}+1)(\mathbf{x}-3) > \mathbf{0}$   
 ss10-  $f(x)=1/x \quad (x+1)(x-3) > 0$   
 ss10+  $f(x)=1/x \quad (x+1)(x-3) > 0$   
 ss10-  $\mathbf{f}(x)=\mathbf{1}/x \quad (x+1)(x-3) > \mathbf{0}$   
 ss10+  $\mathbf{f}(x)=\mathbf{1}/x \quad (x+1)(x-3) > \mathbf{0}$



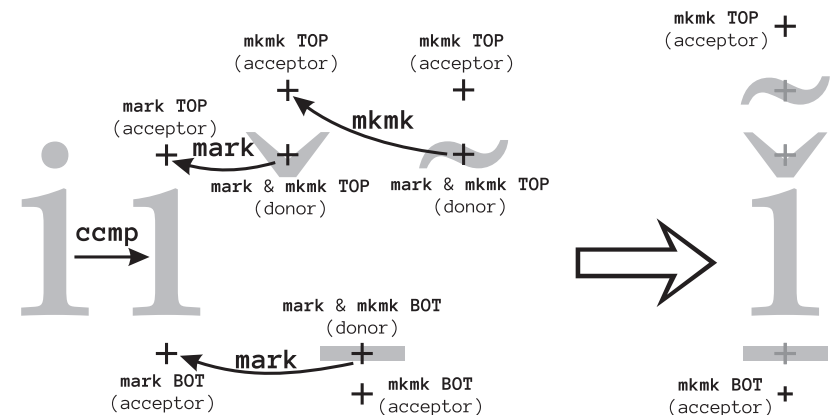
## The anchor mechanism

The following five slides deal with and show samples of placing of accents using the “anchor” mechanism, i.e., the `ccmp+mark+mkmk` features<sup>2</sup>.

Please note:

- ▶ all shown fonts now made with Algotype,
- ▶ samples were compiled with LuaTeX and OTF fonts,
- ▶ feature tables had to be prepared for all fonts and all shapes

## Anchor mechanism scheme - an example



<sup>2</sup>see B. Jackowski et al. in DTK, Heft 3, 2018



## Sample code for placing accents

### Source:

L' or `\char"030C % U+030C (caroncmb, caroncomb)`

g, or `g\char"0326 % U+0326 (uni0326, commaaccentcomb)`

y. or `y\char"0323 % U+0323 (uni0323, dotbelowcomb)`

## TG Adventor (revised)

TG Adventor

L' L' L' L'  
g g g g  
y. y. y. y.



## TG Pagella (revised)

TG Pagella

L' L' L' L'  
g g g g  
y. y. y. y.



## TG Schola (new in the pack)

TG Schola

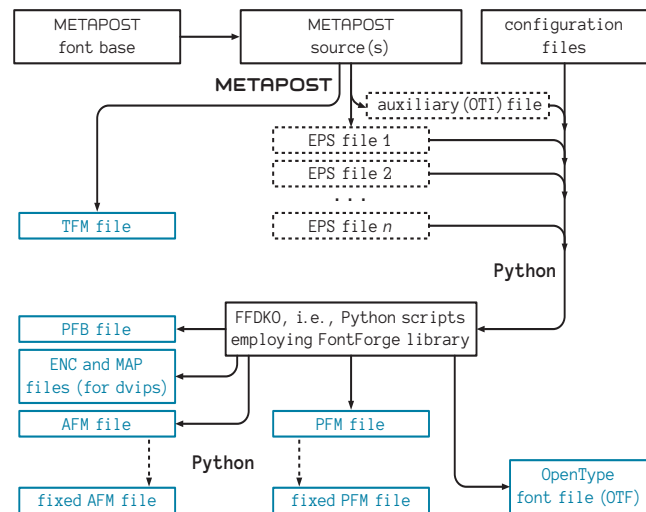
L' L' L' L'  
g g g g  
y. y. y. y.





## Interim MT1(?), no reverse workflow

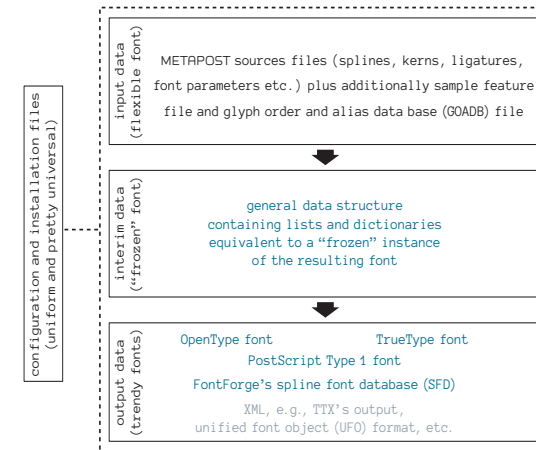
### The scheme of the interim font engine



◀ ▶ ⏪ ⏩ 🔍 ↻

## Algotype: simpler but no reverse workflow<sup>4</sup>

### The scheme of the current font engine



<sup>4</sup>Please note “trendy fonts”: it is now far easier to produce fonts in new/trendy formats.

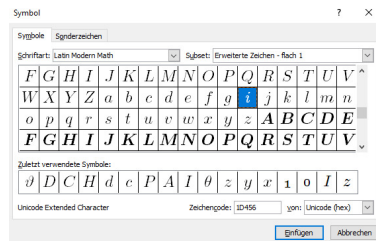
◀ ▶ ⏪ ⏩ 🔍 ↻

## Ten years after

An e-mail dated early March this year by a desperate user:

Missing character in LM Math

I'm having problems with the LM-Math font that I downloaded from your website. It seems that the **small italic h is missing**, see screenshot:



Could you please fix this?

◀ ▶ ⏪ ⏩ 🔍 ↻

## Information out, feedback in

### Information out

The above problem is known since at least 10 years and not fixable by us. How do we<sup>5</sup> disseminate such knowledge? How do we tell users, e.g., of the “backward compatible math style” `ss10`?

### Feedback in

How do we learn about user needs or problems? Do they really need what we are doing? Is, e.g., a monospace font with math symbols for text editors really needed?

**Education(?) is needed!**

**Feedback is needed!**

<sup>5</sup>By “we” the GUST e-foundry team is meant. ◀ ▶ ⏪ ⏩ 🔍 ↻

## Supporters

Support was received from (in no particular order):

- ▶ NTG,
- ▶  $\mathcal{C}_S$ TUG,
- ▶ CG (Context Group),
- ▶ DANTE e.V.,
- ▶ TUG,
- ▶ GUST (non-material),
- ▶ last but not least — individuals.

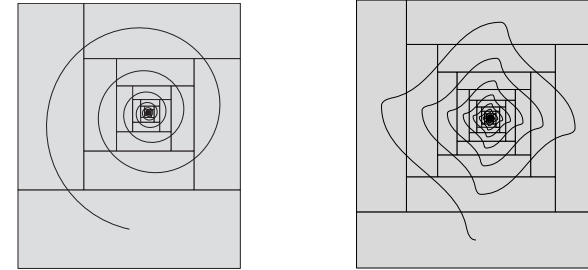
**Thank you very, very much!**



## “Closing”?

Declared financial support came to end, but of course

*closing*  $\neq$  *final*



**To be continued ...**

