The universal Font

Version v2.1

Christian Holm*

2019/08/26

Abstract

This is my implementation of Herbert Bayer's "universal" font in METAFONT for TEX and LATEX. Extensive support for LATEX is supplied. This font is in no way intended to be a correct, not to mention a comlpete implementation of Herbert Bayer's original design. This document describes how to use the font with LATEX, and also the source code for the characters of the font.

Contents

*E–mail: <cholmcc@gmail.com>

1 Introduction

This package contains the METAFONT source and driver files for the "universal" font, designed by Herbert Bayer, a teacher at the Bauhaus school in Weimar, plus a LaTeX package to utilize this font, along with a number of font definition files, as required by the New Font Selection Scheme (NFSS).

About the Documantation

The full documentation of this font is rather large, more then 80 pages actually. Most of it is the programs for the characters for this font, so if you don't know the METAFONT language, or don't care how the charcters are created, you should insert `\OnlyDescription` into the preamble of `universa.dtx`.

Notice, that METAFONT macros are *not* indexed in this documentation. This is because it would take a *major* rewrite of the doc package to do so, and I really didn't want to do that.

Some of the macros of doc *has* been redefined, and if you want to see which, or how I generally did the documentation, please refer to appendix **??**.

If you in some way are unsatisfied with some of the characters of the font, do please read the documentation of that character. There may be some notes on why the character looks the way it does. Some of the comments, however are a bit silly and should be skipped at high speed (pretend there is a conditional that says `if not silly ... fi`).

This Version

This new version of the universal font, provides a number of new features and improvements, both to the font itself and to the LaTeX support macros. Also, a number of corrections has been made to the font programs.

The reason why I jumped one whole version number from 1.0 to 2.0, is that I have taken a whole new approach to the font programs and shapes of the universal font, plus I decided to put some more effort into the LaTeX support.

Thanks and Other Stuff

To those of you who have had the (mixed) pleasure of using version 1.0 of this font, one of the most notaticable changes are to the symbol  . In the old documenation, I complained that I couldn't find the exact solution to the problem this symbol possed. I also encourged people to send me any solution they may have had — and guess what — somebody did! Therefore I would like to thanck Joseph Collins for providing me with the solution. If you like Joseph Collins and

I like Mathematical puzzles take a look in Appendix A to learn more about this problem and it's solution.

I also direct your attention to Appendix B for the copyright notice on the universal package (it's the *Gnu General Public Lisence* to those of you who know it).

If you in any way have gotten tempted to design your own font, or to implement some font into METAFONT, I feel obligated to bring you a warning, taken from the METAFONTbook by Donald E. Knuth:

> Warning: Type design can be hazardous to your other interests. Once you get hooked, you will develop intense feelings about letterforms; the medium will intrude the message that you read. And you will perpetually be thinking of improvements to the fonts that you see everywhere, especially those of your own design.

2 The Font and it's History

Bauhaus

The Bauhaus school in germany originally located at Dessau, was a school for any kind of design, ranging from potery to furniture, from painting to — what was considered the prime form of design — architecture. Many famous designers came from, or taught at Bauhaus, for example Mies van der Rhoe, Herbert Bayer, Kandinsky, Walter Gropius and Gerrit Rietveld. The style "die stiil" was explored here, and painters like Mondrian made large contributions to what today is known as "the Bauhaus style".

The basic idea of the Bauhaus school, was to design items, which along with it's aesthethic value, also had a high degree of functionallity. Houses were meant to be suited for all kinds of living, while still keeping the beauty that make people glad to see thier house. Chairs should be comfortable for thier use, as well be able to fit–in in a normal house. All this should be done at a price that made it possible for everybody to own designer–furniture, houses, etc.

The political idea of Bauhaus, was that of a socialist one. Houses are for the people — they have to live in them, and that living should be a good as possible. Therefore the Bauhaus school saw it has its task to provide functional, beautiful everyday items that anybody could affort. Paintings and tapestery shouldn't hang on museums or art galleries, but in peoples homes, where they would inrich thier everyday life.

Herbert Bayer and the "universal" Font

At Bauhaus typography was also studied, not just how written text should be typeset, and how printed characters should look like, but also what the essence of writting is, in it's pratical and design–wise sense. This led Herbert Bayer to formulate some principles of writting:

> experiment with simplified way of writing:
>
> 1. this way of writing is recommended by all typographic designers as the future way of writing.
> 2. by writing in minuscules our writing looses nothing, but is easier to read, considerably more economical.
> 3. why must you for one sound have two tokens, e.g. A and a? why two alphabets for one word, why this double set of signs, when the half is enough.
>
> herbert bayer 1925

On this principles, Herbert Bayer designed a font, which should have no majuscles (upper case letters), easy to print, and easy to read since it didn't have any unusefull decoration, but communicated the bare meaning of characters through the simplest forms needed to reconigise a character. This font he called "universal".

This font contained abolutly no majuscles, since Bayer believed them to be superflous, as it is clear form the quote above.

At the time Herbert Bayer formulated these principles and designed the "universal" font, most printers used Gothic letters, which is allmost overly decorated, so his font ofcourse made contraversy.

Later on, in the 1930'ies, the Bauhaus school drew the attention of *Gestapo*[1] of Nazi[2] Germany. The school was finally closed in 1936 by Gestapo, because they believed they were promoters of Jewish and Communist culture and propaganda.

In the aftermatch of the closing of the school, most of it's ideas were shundend by other designers, and the Bauhaus way of thinking died out. This is properly the reason why Herbert Bayers "universal" font is so little known today.

However, the "universal" font still stands as one of the most compelling developments in font designing. It represents an approach to designing where the functionallity is as vital as the expression, and as such I believ it to be one of the most important fonts in the world today.

3 This METAFONT Implementation

This implementation of Herbert Bayers "universal" font, is not supposed to look *exactly* like the original design. Ofcourse I have tried to the best of my ability to mimic his design as far as I could. However, it is not an easy matter to find a complete, not to mention exact, sample or account of Bayers design.

This implementation is bassed on the samples I *could* find, and other implementations of the "universal" font. Many of these other implementations do differ from the original samples, and include characters I couldn't find in any of the original. So whenever I found disparities, I mainly leened on the original samples and my understanding of the original design.

[1] Gehemligche Stats Polizi
[2] Nationalsocializmus

3.1 The METAFONT version versus the Original

As mentioned above, Bayer never did design any majuscles for this font, but nonetheless, I have included them into this implementation. This I did, because I think most people will have a hard time writting in minuscles (lowercase letters) alone. Ofcorse, if you agree with Bayer, you should simple not use them.

There are also some other differences, mostly due to the fact, that I never found a complete sample of the original font. The major differences between the originalk font and this implementation is summerized below:

Majuscles: Majuscles are present, even though they weren't in the original design.

Digigts: These are based on other implementations, and my general conception of the original design.

Punctuations: As above.

Accents: As above.

Symbols: As above.

Bauhaus Symbols: I have added some various symbols I have found in connection with Bauhaus to the font. The reason is I find them beautiful, and I had some space to fill.

Numerous Shapes and Weights: I don't think Bayer ever design slanted characters, a bold face version of the font, and he could never have design a small caps version of the font. However, these are present in this implementation. I included these features, because I believ them to be of general utility, and it makes the font conform more to the Computer Modern Roman font, and NFSS.

3.2 Features of the font

Rather then using `cmbase.mf`[3], and then redefine some macros, I chose to make a new base file myself, i.e., `unibase.mf`. This file contains a number of macros[4] I have used in the character programs.

The macros of `unibase.mf` actually reflects my conception of the font. There are three basic drawing macroes:

`unicir` which draws a circle,

`uniarc` which draws a segment (arc) of a circle, and

`unilne` which draws a straight line.

[3] Computer Modern Roman base file

[4] A better name for 'base' would be 'library', and then the file would be `libuni.mf`, but to conform to CM, I used 'base'.

I believe, that Bayer intended the font to be made of these two basic shapes: the arc, and the line. Also, to keep things simple, and therefore easy to print, all shapes should be of the same thickness, i.e.,as if drawn with a pen of equal thickness. I have made one diviation from this, however. All majuscles are drawn with a thicker 'pen', which makes the output nicer, I think.

Incidently, this made the programs of the characters much simpler, and shorter.

3.2.1 Series, Shapes, Sizes, and Special Characters

Change on 98/08/01, Version 2.0

Below is a sample of each series/shape combination avaliable in this implementation of the font, along with the LaTeX commands that drive them:

Medium upright (`\textuni`): The dazed brown fox quickly gave 1234-567890 jumps!

Medium slanted (`\textunisl`): *The dazed brown fox quickly gave 1234-567890 jumps!*

Medium small caps (`\textunisc`): The dazed brown fox quickly gave 1234-567890 jumps!

Medium strict (`\textunist`): the dazed brown fox quickly gave 1234-567890 jumps!

Bold face upright (`\textunibf`): **The dazed brown fox quickly gave 1234-567890 jumps!**

Bold face slanted (`\textunibsl`): ***The dazed brown fox quickly gave 1234-567890 jumps!***

Bold face small caps (`\textunibsc`): **The dazed brown fox quickly gave 1234-567890 jumps!**

Bold face strict (`\textunibst`): **the dazed brown fox quickly gave 1234-567890 jumps!**

Change on 98/08/01, Version 2.0

Everyone of these shapes are avaliable in size 8, 9, 10, 12, 17 pt, and METAFONT can ofcourse create others.

Also a number of non–standard characters are avaliable in the font. Below is a table of these characters along with the LaTeX commands that drive them.

Notice that `` and '' is present in the table. This is because these characters are not directly defined, but is supplied as *ligatures*. This can be done, because they are simple doubles of ` and '.

A quick look on the table will also reveal some characters that generally isn't present in the standard OT1 encoding[5], but generally present in the T1 encoding[6].

[5] The OT1 encoding is the 7 bit encoding of the Computer Modern fonts by Donald E. Knuth. 7 bit means it contain 128 ($= 2^7$) characters.

[6] The T1 encoding is an encoding especially designed for the (western) European languages. It was founded by the TeX User's Group, on a seminar in Cork, and is the basis of the dc fonts. T1 is an 8 bit encoding, which means it has 256 ($= 2^8$) characters.

I have done this, both to provide an (almost) complete font for the European languages, but also because I anticipate the universal font some time in the future will shift, or a least be avaliable, in the T1 encoding[7]. Please note, that in

■	\bausquare	●	\baucircle
▲	\bautriangle		\bauhead
	\bauforms	ð	\dh
đ	\dj	ŋ	\ng
þ	\th	ɋ	\varQ
Đ	\DH	Đ	\DJ
Ŋ	\NG	Þ	\TH
§	\textsection or \S	{	\textbraceleft or \{
}	\textbracerigth or \}	|	\textbar
‹	\guilsinglleft	›	\guilsinglright
«	\guillemoleft	»	\guillemoright
‚	\quotesinglbase	„	\quotedblbase
“	\textquotedblleft	”	\textquotedblright
¸	\textogonek	_	\textunderscore

Table 1: Non–standard characters in the universal font

the small caps shaped fonts, \dh does not give Đ, but a ɋ, that is a small caps shaped version of \varQ. Also there is no command \varq defined.

In appendix D is some charts showing the font in different series and shapes.

3.2.2 File Names for the METAFONT files

Base File and Source Files The base file and the files containg the code for the characters of the universal font, all starts with uni, to reflect the connection of the files. The next five possible letters reflects what kind of code is contained within the file, e.g., the base file ends in base, the file containing the code for the minuscles (lower case letters) end in lower, and so forth.

Font Driver Files The font driver filenames has been chosen to conform to the fontname scheme, because this scheme is used by most TEX, LATEX, and METAFONT systems (anyway those that use kpathsea, which is the most).

The filename all contain the three characters ful, where the f stands for *public* and ul for *universal*.

Next comes a letter which is one of m (*medium*) or b (*bold*), which represents the series of the font.

Then comes one or two letters, which are r (*upright* or roman), o (*slanted*, or oblique), c (*small caps*), or st (*strict*), which represent the shape of the font.

Finally the filename ends with the designsize in points.

[7] T1 is generally considered *the* encoding of the future, and in the long term, it is most likely the encoding of LATEX3.

Thus the complete syntax for the font driver file names is: The `fontname`

$$
\begin{array}{rcl}
\langle filename \rangle & := & \langle supplier \rangle \langle face \rangle \langle series \rangle \langle shape \rangle \langle size \rangle \texttt{.mf} \\
\langle supplier \rangle & := & \texttt{f} \\
\langle face \rangle & := & \texttt{ul} \\
\langle series \rangle & := & \texttt{m | b} \\
\langle shape \rangle & := & \texttt{r |o |c |st} \\
\langle size \rangle & := & \texttt{| 8 | 9 | 10 | 12 | 17}
\end{array}
$$

scheme actually says to put ⟨*encoding*⟩ information after the ⟨*shape*⟩, but since this is `8r` for TEX Text, it would make filenames longer then 8 characters in the cases of ⟨*size*⟩ of 10, 12, and 17, so this information is left out (which is permisable in `fontname`, but unfortunate).

This way of naming the font driver files will, if you use `kpathsea`, put the `ful*.*pk` files in

⟨*pk–base–dir*⟩`/public/universa/`

and the `ful*.tfm` files in

⟨*tfm–base–dir*⟩`/public/universa/`

which I think is the intuitivly correct place to put them. This also means, that the `*.mf` files provided with this package, should be placed in

⟨*mf–source–base–dir*⟩`/public/universa/`

again very intuitive.

Below is a table of the usual directory names under Unix–like and MSDOS–like (including Windows95) systems.⟨*mode*⟩ is dvips's name for your printer.

Varibale	Unix–like systems	MSDOS–like systems
⟨*tfm–base–dir*⟩	`/var/spool/texmf/pk/`⟨*mode*⟩`/`	`C:\FONTS\PK\`⟨*mode*⟩`\`
⟨*tfm–base–dir*⟩	`/var/spool/texmf/tfm/`	`C:\FONTS\TFM\`
⟨*tfm–base–dir*⟩	`/usr/local/lib/texmf/fonts/source/`	`C:\TEX\MFINPUTS\`

Table 2: Common directory names.

3.3 The LaTeX and NFSS Support

To use the `universal` font with LaTeX 2_ε[8], you should load the package `uni` with the command

`\usepackage[`⟨*options*⟩`]{`⟨*uni*⟩`}`

in your preamble (i.e., after `\documentclass` and before `\begin{document}`). ⟨*options*⟩ can be any of the options described below, but no other.

3.3.1 Options to uni

strict
Change on 98/08/01, *Version* 2.0

The srict option is intended to fassilitate typesetiing of the universal font in a *strict* bauhaus fashion, that is *only* in minuscules.

In this font, only the series may be variated, that is, there is a bold series strict shaped font of any size, and a medium series strict shaped font of any size, in the universal family.

This option can be used in conjuntion with options medium and bold. Please notice, that it doesn't make any sense to ask for a small caps or slanted shaped font, while using this option.

Notice that only \textuni and \uni is defined if option strict was given to uni package.

default
Change on 98/08/01, *Version* 2.0

If you give the default option to the uni package, the default font of the document will be universal.

If you also used the option bold the default font will be the universal font in **bold** series. Otherwise it will be in medium series.

With this option, \textit, \it, and \itshape shifts to *universal slanted* font, i.e., there is no *italic* font avaliable.

\textcmr
\cmr

To make it possible to change back to Donald E. Knuth's Computer Modern Roman font, even when the default option is given, we define macros \textcmr and \cmr, which switches the \fontfamily to cmr locally and gobally respecivly.

You should use this option with some care, since the universal font isn't very suited for longer texts, but rather for short letters, quotes, and other pieces of text where the graphical apperance is important.

medium
Change on 98/08/01, *Version* 2.0

When this option is given, command \textuni switches to medium series universal font, as do \uni. The other font selection commands behave as always (see below). This is the default option to uni, i.e., not normally needed.

bold
Change on 98/08/01, *Version* 2.0

If this option is given, commands \textuni and \uni switches to **bold series universal font**. Other font selection commands behave as always (see below). Notice that it makes no sense to give both option medium and option bold to the uni package.

3.3.2 Font Selection Commands

\textuni
\uni

These two commands only change the current font family to uni *and nothing else*. That means, that if you say for example

```
{\sl Hello \textuni{world}}
```

you get *both* 'Hello' and 'world' in slanted shape, and the output would be

Hello world

To put it in another way: Font encoding, shape, size, and baselineskip is preserved under \textuni and \uni, while font family is not.

[8] I have made *no* attempt to provide support for LaTeX2.09, since this format is obsolete, and those who *do* use it, will proberly never bother to look at CTAN for new fonts anyway.

Execptions: If you gave the `bold` option to the `uni` package, then this command will always give you a bold series font. If you gave the `strict` option, then this command will always give you a strict shaped font.

These commands can be used in conjuntion with LaTeX commands `\textbf`, `\textsl`, and even `\sc`, or `\rm` to give different series and shapes.

`\textunirm` `\unirm`

If you in the previous example intended to shift to medium upright universal font you could instead have used `\textunirm`, since this command *does not* preserve font shape, i.e., the shape is always changed to upright, regardless of the prevouis shape. So if you said

```
{\sl Hello \textunirm{world}}
```

you would get

Hello world

`\textunisl` `\unisl` `\textunisc` `\unisc` `\textunist` `\unist` `\textunibf` `\unibf` `\textunibsl` `\unibsl` `\textunibsc` `\unibsc` `\textunibst` `\unibst`

Actually `\textunirm` and `\unirm` isn't the only commands that aggresively changes most of the font parameters. `\textunisl`, `\textunisc`, and `\textunist`, *always* gives you *medium slanted*, MEDIUM SMALL CAPS, and medium strict respecitivly no matter what the values of `\f@shape` was before.

In the same category is `\textunibf`, `\textunibsl`, `\textunibsc`, and `\textunist` which always changes the font series to **bold**, along with change in shape (upright, slanted, small caps, and strict in that order).

All of the 'aggresive' commands, *do not* however change the *size* and *baselineskip* of the font. This should be done by using LaTeX commands such as `\small`, `\Large`, `\fontsize{`⟨*size*⟩`}{`⟨*lineskip*⟩`}`, etc.

Notice that the 'aggresive' commands always changes to the appropiate font series. That is, even if you gave the `bold` option to `uni`, `\textunirm` will still give you medium upright universal font. This particular instance illustrates the use of the aggresive commands quite well I think.

To summerize: The 'aggresive' commands *doesn't* preserve fammily, series and shapes, but *does* preserve encoding, size, and baslineskip.

Warning: The font shifting commands `\textuni...` and `\uni...` in this section is *not* defined if you gave the `strict` option to the `uni` package.

`\unifamily`
Change on 98/08/01, *Version* 2.0

One can also use the rather primitive command `\unifamily` in conjunction with `\selectfont` as described in *LaTeX 2ε Font Selection*, to change the font family to `universal` if absolute control is preferd.

`\unifamily` is used by all the other font switching commands, so if you redefine it, or `\unifamilydefault`, you could get strange resault.

`\unifamilydefault` This command normally expands to `uni`, which is the 'family' name of the `universal` package. If you redefine this command to be something else, e.g., `cmr`, `\unifamily` will load another font.

`\uniseries`
Change on 98/08/01, *Version* 2.0

If the `bold` option to `uni` is used this command will select the default series of the `universal` font, defined in `\uniseriesdefault`, which ofcourse defaults to `b`, i.e., bold series. if you redefine `\uniseriesdefault` to be `m`, then `\uniseries` will select medium series fonts.

If `bold` option wasn't given, then this expands to nothing, as do `\uniseriesdefault`.

This command is used by `\textuni` and `\uni`.

`\unishape` *Change on 98/08/01, Version 2.0*

If the `strict` option to `uni` is used this command will select the default shape of the `universal` font, defined in `\unishapedefault`, which ofcourse defaults to `st`, i.e., strict shape. If you redefine `\unishapedefault` to be `n`, then `\uniseries` will select upright shaped fonts.

If `strict` option wasn't given, then this expands to nothing, as do `\unishapedefault`.

This command is used by `\textuni` and `\uni`.

`\stshape` *Change on 98/08/01, Version 2.0*

This command switches to *strict* shape, i.e., all majuscules will be typeset as minuscles. This makes it possible in a simple way to typeset text in the way Herbert Bayer thought it should, as evident from the citation above.

This command uses the command `\stdefault`, which defaults to `st`. If you redefine this to something alse, e.g., `sl` you will get a *slanted* font.

This command is used by `\textunist`, `\textunibst`, `\unibst` and `\unist`, and is defined no matter waht options you gave to the `uni` package.

Warning: Since *strict* is a non–standard shape, this command should not be used outside the `universal` font, since this may give you unexpected resaults.

3.3.3 Special Character Commands

For the individual commands that makes various special characters, please consult table 1 above.

3.3.4 Other Commands

`\k` The macro `\k` used in the `universal` font gives the accent ogenek, that is a reversed cidelia accent. It takes one argument, which shuold be a single letter, under which it puts the accent. For example, you could say `\textuni{\k{a}}` and get ą.

`\DeclareUniChar` *Change on 98/08/01, Version 2.0*

Now you can configure the special character commands of the font, via the commands `\DeclareUniChar` and `\DeclareUniCommand`. The commands defined via these commands will only work in accordence with it's definition inside the `universal` font, and if defined elsewhere according to it's definition there, else it will give an error message.

`\DeclareUniChar` is used to define a command sequence representing a single character in the `universal` font, much like `\DeclareTextSymbol`, or in TEX `\chardef`, though the control sequence will produce an error message outside the `universal` font, and properle unexpected resaults outside the `OT1` encoding[9].

`\DeclareUniCommand` is used to define control sequences inside the `universal` font, representing many characters or doing complex manuvers on characters and stuff. The optional argument to `\DeclareUniCommand` can be used to say how many arguments the control sequence should have, just like `\newcommand`. However, it is not possible to give a default first argument.

[9] This should not be a problem.

The below defintion uses the color package to typeset a square, circle, and triangle in different colours[10].

```
\DeclareUniCommand{\mybauforms}{%
  \lower.5ex\hbox{\color{blue}\bautriangle}%
  \kern-.5em\raise.5ex\hbox{\color{read}\baucircle}%
  \kern-.5em\lower.5ex\hbox{\color{yellow}\bausquare}}
```

Yet another example, using arguments could be

```
\DeclareUniCommand{\mybaulogo}[1]{\bauhead\ {\Large #1}}
```

so you could say `\mybaulogo{Christian Holm}` and get:

and I bet you can come up with some even more useful and complex commands.

The syntax of `\DeclareUniChar` and `\DeclareUniCommand` is

`\DeclareUniChar{`⟨*cmd*⟩`}{`⟨*slot*⟩`}`
`\DeclareUniCommand{`⟨*cmd*⟩`}[`⟨*arg*⟩`]{`⟨*definition*⟩`}`

where ⟨*cmd*⟩ is the user command defined, ⟨*slot*⟩ is the number of the character in the font, ⟨*arg*⟩ is the number of arguments and ⟨*definition*⟩ is what ⟨*cmd*⟩ does.

3.3.5 File Names for the LaTeX files

All the LaTeX files contain the three letters `uni`, to reflect the connectedness of the files. The font definition files all start with the letter code appropiate for the encodeing.

To follow the scheme of the METAFONT files, it would be appropiate to place all LaTeX files in

⟨*tex–base–dir*⟩[11]`/tex/latex/universa/`

3.4 The Major differences between version 1.0 and 2.0

First of: *a lot* of bugs and errors has been corrected. In version 1.0, I had made the (stupid) mistake of calling the macro mode_setup before I defined the unsharped units. Ofcourse a quick look in the METAFONTbook showed be just how stupid this is. This made the font very vunerable to mode specifications, which ofcourse isn't the idea.

[10] Since the documentation should be avaliable to all, I can not provide you with the outcome of this example, since it needs the color package wich may not be avaliable on all sites. I suggest you try it out, or something similar, if you can.

[11] On Unix–like systems ⟨*tex–base–dir*⟩ is usually something like `/usr/local/lib/texmf/`, and on MSDOS–like systems something like `C:\TEX\` .

Secondly: I chose a completly new approach to the character programs, which resaultet in `unibase.mf`. The idea is to define a few macros, and then utilize those in the character programs, so that these programs can be kept simple, efficient, and intuitive.

A quick look at `unibase.mf` will also reveal that I chose a new way of adjusting the characters. This means that the macros `bauhaus...` present in version 1.0 no longer is needed, and since they only tended to ubscure things rather then simplify them, I went back to the plain METAFONT macro `beginchar`, which is much stronger.

All in all, `unibase.mf` provides a much stronger and uniform frame work for character design, then did the old `universal.mf`.

Thirdly: The file names have been keept inside MSDOS conventions, that is first name of maximum 8 charcaters, and last name of maximum 3 characters. This does mean, however, that some file names are not intutive, but I have tried to make them as much as I could.

Also, every file associated with this font, execpt the font definition files (`*uni.fd`), and font driver files (`ful*.mf`) begins with the three letters `uni`, to emphasize the connection.

Fouthly: Some new font shapes are avaliable, as explained above. I found out, during the design of the characters, that new the base file `unibase.mf` kept showing new potentiality, and the extension of the font to include more shapes was very easy inside the frame of this base, so I thought "What the heck!"

Fifthly: Some of the bauhaus symbols avaliable in version 1.0, has been taken out, and some new, more general characters have been added. Most of the absent symbols where not really of general use, so I decided to take them out, since I was never really satisfied with those anyway. This also made the font contain exactly 128 characters, just like a normal Computer Modern Roman font.

Sixthly: I improved the LaTeX and NFSS support considerably. The changes are legion, but let me sum up the most important here.

1. Stronger font selection commands.
2. Command names that should be more intuitive.
3. Preperations for `T1` encoding.
4. Conformation to LaTeX 2_ε style, and therefore a better chance to conform with the future LaTeX3 format.
5. More and better options.

A Solution to the `\bauhausforms` problem

A.1 The Problem

This I owe to Joseph Collins `<collins@ARL.MIL>`.

During the preperation of version 1.0 of this font, I was loosing sleep over what I chose to call 'The `\bauhausforms` problem', after the problems the symbol  gave me.

As can be seen from the symbol, the idea is to make a figure out of a circle, a square, and a triangle. From these three figures you can ofcourse make infinitly many figures, even though the sizes are limited. The particular combination of the three figures I was looking, first seemed simple[12], but turned out to be difficult. What I wanted to do was (see also figure 1):

> Take a circle of some radius (e.g., 1). Inside this circle draw a isosceles triangle $\triangle ABC$, with all vertices on the circle, The singular vertice A placed on the horizontal line traveling left from the center of the circle.
>
> Now draw a square $\square PQRS$ inside of the circle, having two vertices on the circle, and two on the triangle.
>
> The wanted triangle and square are such that, the opposing side of the singular vertice `BC` should divede the square into to equal oblongs, i.e., intersect `PQ` midway between P and Q.

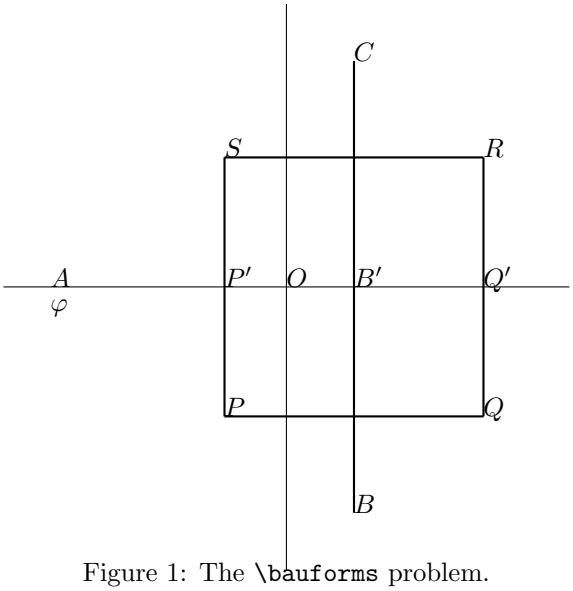

Figure 1: The `\bauforms` problem.

This shouldn't be to difficult, should it. Well I didn't think so, but after many late evenings with pen, paper, ruler, compasses, and heavy use of trigonometric

[12] And after having seen Mr. Collins solution, it did again.

relations, I found out that the half φ of the singular vertice should obey:

$$\begin{aligned} 0 &= \sin\left(\cos^{-1}\left(\frac{\cos 2\varphi}{\sqrt{2}}\right)\right) - \frac{\cos 2\varphi}{\sqrt{2}} \\ &\quad -2\sqrt{2}\frac{\cos\varphi^2 \sin\varphi}{\cos\varphi + \sin\varphi} \end{aligned}$$

Now I dare you to find the exact solution to that.

Using numerical methodes (Newton's method), was ofcourse no problem, and gave satesfactory resualt. If the expression on the right above is labelled f, f' is:

$$\begin{aligned} f' &= \frac{\sin 4\varphi}{2\sqrt{\frac{3-\cos 4\varphi}{4}}} + \sqrt{2}\sin 2\varphi \\ &\quad -\frac{2\sqrt{2}\cos\varphi}{\cos\varphi + \sin\varphi^2}\left((\cos\varphi + \sin\varphi)\left(\cos^2\varphi - 2\sin^2 2\varphi\right)\right. \\ &\quad - \left.(\cos\varphi - \sin\varphi)\cos^2\varphi\sin\varphi\right) \end{aligned}$$

Using these expressions for f and f' in a Fortran program, I reached resaults close to what Mr. Collins found.

A.2 Joseph Collins' Solution

In Mr. Collins notation, the points on figure 1 has the following coordinates:

$$\begin{array}{rrr} A = (0,-1) & B' = (x,0) & C = (x,y) \\ O = (0,0) & P' = (x-h,0) & Q' = (x+h,0) \\ R = (x+h,h) & S = (x-h,h) & \end{array}$$

Below is what Mr. Collins wrote me — thank you very much.

On the unit circle

$$x^2 + y^2 = 1 \tag{1}$$

we have the vertices of a triangle at $(-1,0)$, (x,y), and $(x,-y)$. A square has four vertices $(x\pm h, \pm h)$, where the two points $(x-h,\pm h)$ lie on the triangle (constraint A) and the two points $(x+h,\pm h)$ lie on the circle (constraint B). Thus, the vertical side of the triangle bisects the square. From constraint A, upon consideration of similar triangles, we have

$$\frac{y}{1+x} = \frac{h}{1+x-h}, \quad \text{so that} \quad h = \frac{(1+x)y}{1+x+y}.$$

By (1), this is

$$h = \frac{(1+x)\sqrt{1-x^2}}{1+x+\sqrt{1-x^2}}. \tag{2}$$

From constraint B, we get

$$(x+h)^2 + h^2 = 1. \tag{3}$$

Any simultaneous solution of (2) and (3) is also a solution of

$$8x^3 - 4x^2 - 3x + 1 = 0, \tag{4}$$

the relevant solution being

$$x = \frac{1}{6} + \sqrt{\frac{11}{18}} \sin\left[\frac{\pi}{6} - \frac{1}{3}\arctan\left(\frac{3\sqrt{237}}{23}\right)\right]. \tag{5}$$

Equation (4) and solution (5) courtesy of Mathematica. We get y and h from (1) and (2), respectively. The angle at $(-1, 0)$ is $\varphi = 2\arctan\big(y/(1+x)\big)$.

Approximate values are

$$\begin{aligned} x &\simeq 0.2865914 \\ y &\simeq 0.9580529 \\ h &\simeq 0.5491394 \\ \varphi &\simeq 1.280129 \ (\simeq 73.346^o) \end{aligned}$$

B Copyrights — GNU General Public Lisence

C Wishlist

Below is a list of things I would like to do with the font and package. If anyone has any suggestions, ready–made code, or new ideas, please let me know.

If you would like to take on one or more of the tasks presented below, please do so, but send me a note so that I may coordinate with my own efforts, and perhaps have a constructive dicourse.

I should however instruct you to read the *complete* documentation of the package and font, since this may give some reasons why I have chosen a particular approach.

- Make the font an 8–bit encoded (256 characters) font, conforming somewhat to the `T1` encoding of the Cork fonts.
- Making a package (perhaps mfdoc), to make documentation of METAFONT sources, just like doc is fore LaTeX sources. This is ofcourse a independent project, and I will proberly not work on it.

D Font Charts

Below are some charts of the `universal` font in different series and shapes (medium upright, slanted, small-caps, and strict, as well as bold upright, slanted, small caps, and strict), all in size `8pt`.

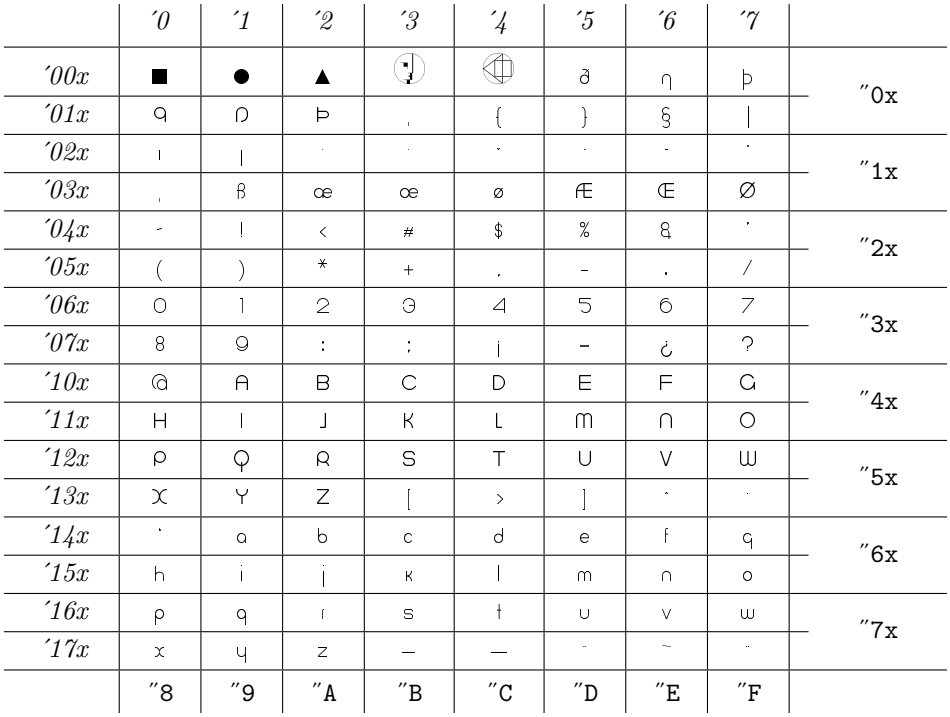

Figure 2: Letters and symbols in `fulmr8.mf` — medium, upright

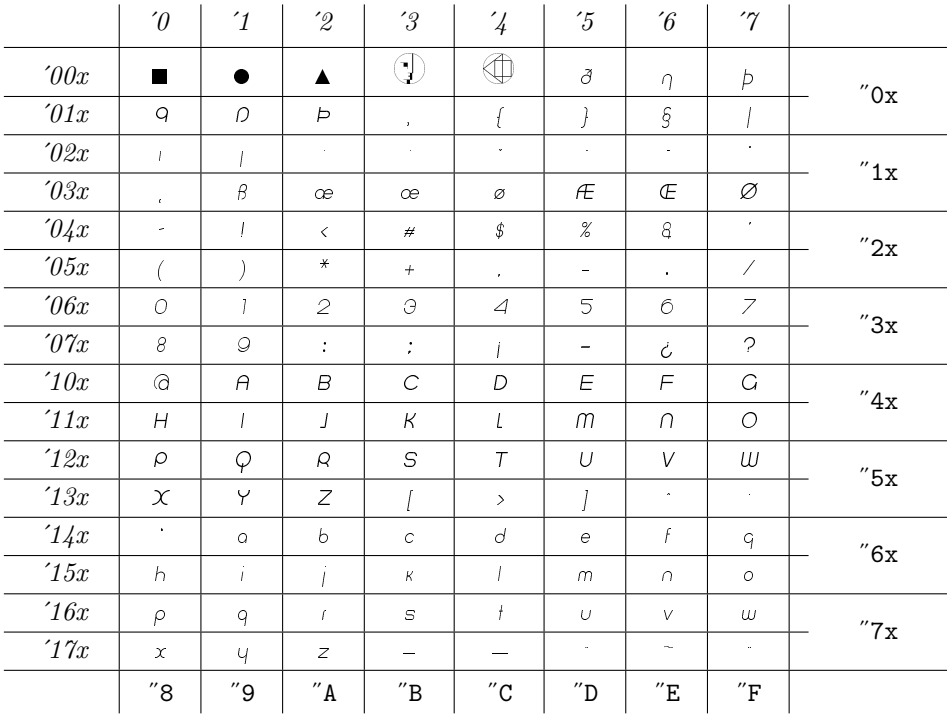

Figure 3: Letters and symbols in `fulmo8.mf` — medium, oblique

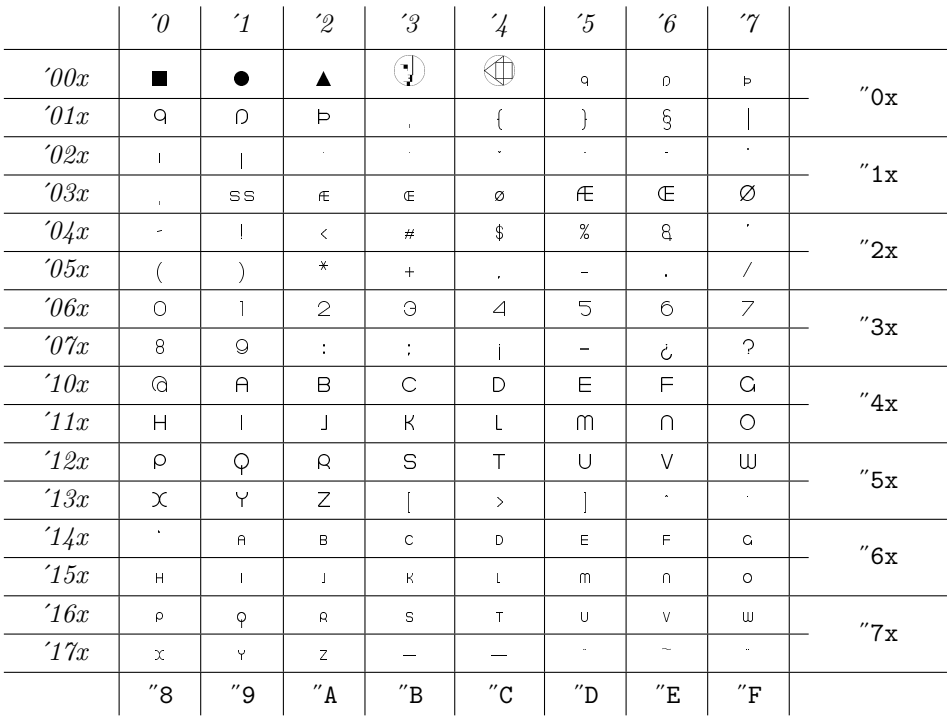

Figure 4: Letters and symbols in `fulmc8.mf` — medium, small-caps

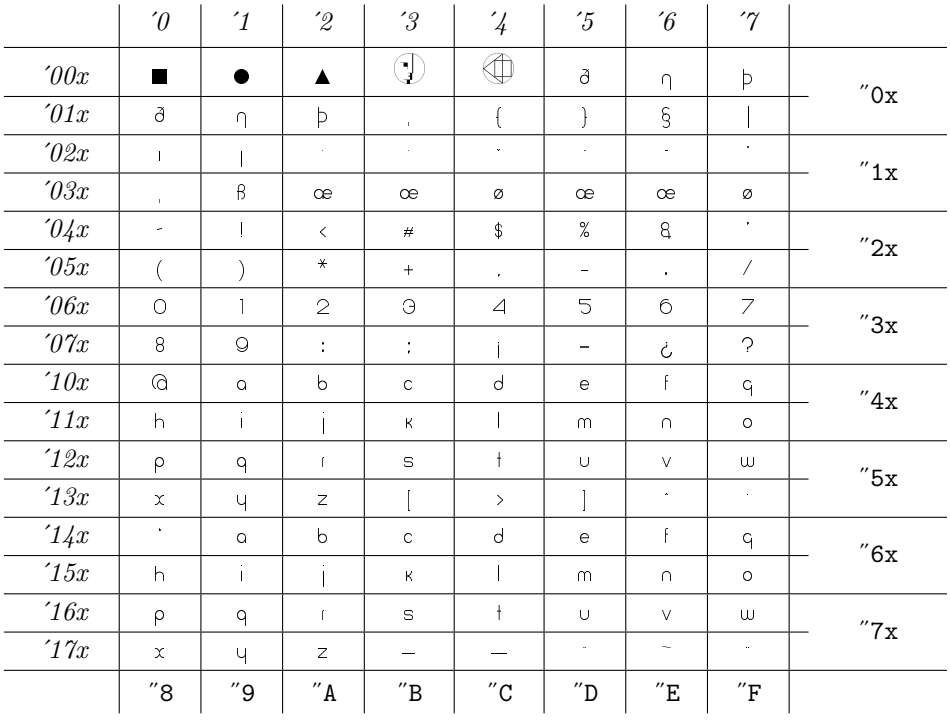

Figure 5: Letters and symbols in `fulmst8.mf` — medium, strict

	´0	´1	´2	´3	´4	´5	´6	´7	
´00x	■	●	▲			ð	ŋ	þ	″0x
´01x	ɋ	ɒ	ᵽ	ˌ	{	}	§	|	
´02x	ˈ	ˌ	ˋ	ˊ	ˇ	˘	˗	˙	″1x
´03x	˛	ß	œ	œ	ø	Æ	Œ	Ø	
´04x	-	¡	‹	#	$	%	&	’	″2x
´05x	(	)	*	+	‚	-	.	/	
´06x	0	1	2	3	4	5	6	7	″3x
´07x	8	9	:	;	¡	=	¿	?	
´10x	@	A	B	C	D	E	F	G	″4x
´11x	H	I	J	K	L	M	N	O	
´12x	P	Q	R	S	T	U	V	W	″5x
´13x	X	Y	Z	[	›	]	^	˙	
´14x	‘	a	b	c	d	e	f	q	″6x
´15x	h	i	j	k	l	m	n	o	
´16x	p	q	r	s	t	u	v	w	″7x
´17x	x	y	z	–	—	˝	~	¨	
	″8	″9	″A	″B	″C	″D	″E	″F	

Figure 6: Letters and symbols in `fulbr8.mf` — bold, upright

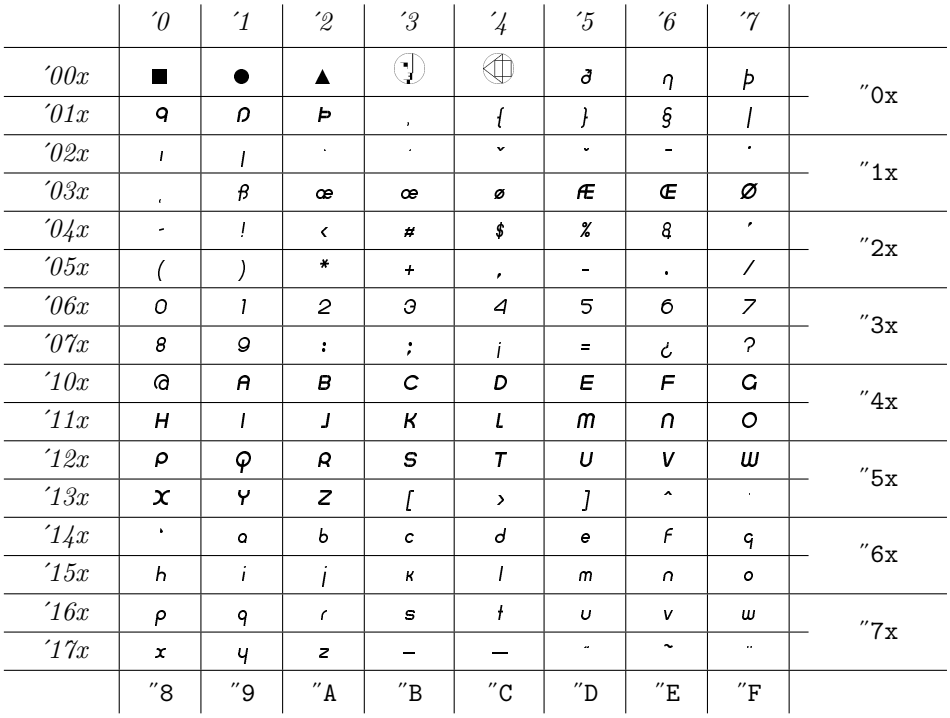

Figure 7: Letters and symbols in `fulbo8.mf` — bold, oblique

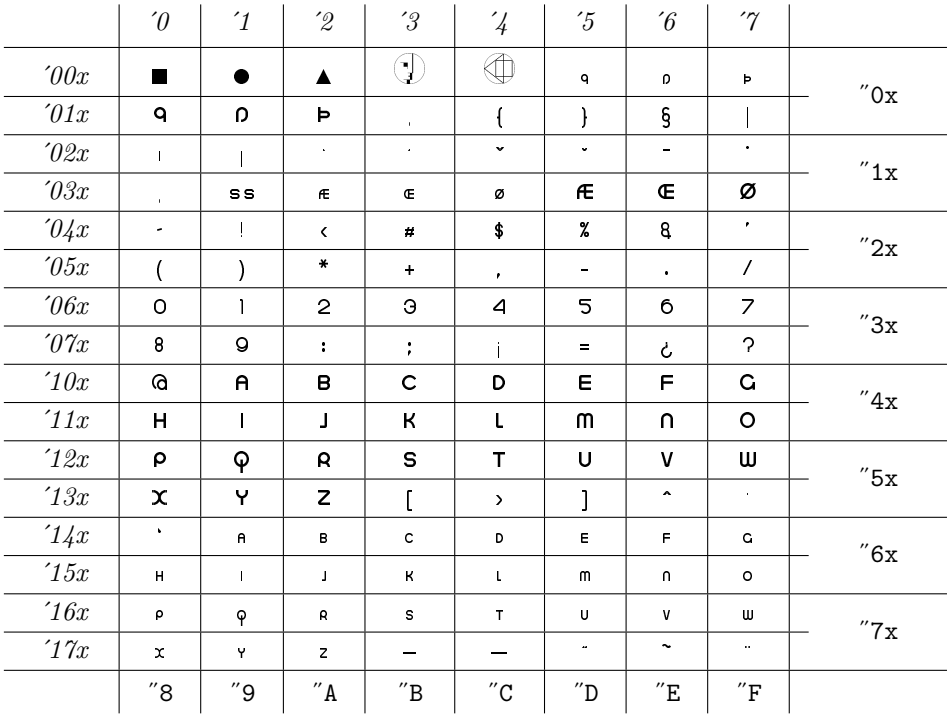

Figure 8: Letters and symbols in `fulbc8.mf` — bold, small-caps

	´0	´1	´2	´3	´4	´5	´6	´7	
´00x	■	●	▲			ð	ɳ	þ	″0x
´01x	ð	ɳ	þ	ˌ	{	}	§	|	
´02x	ˌ	|	ˋ	ˊ	˘	˜	¯	˙	″1x
´03x	¸	ß	œ	œ	ø	œ	œ	ø	
´04x	-	!	‹	#	$	%	ß	’	″2x
´05x	(	)	*	+	‚	-	.	/	
´06x	0	1	2	3	4	5	6	7	″3x
´07x	8	9	:	;	¡	=	¿	?	
´10x	@	a	b	c	d	e	f	g	″4x
´11x	h	i	j	k	l	m	n	o	
´12x	p	q	r	s	t	u	v	w	″5x
´13x	x	y	z	[	›	]	^	˙	
´14x	‘	a	b	c	d	e	f	g	″6x
´15x	h	i	j	k	l	m	n	o	
´16x	p	q	r	s	t	u	v	w	″7x
´17x	x	y	z	–	—	˝	~	¨	
	″8	″9	″A	″B	″C	″D	″E	″F	

Figure 9: Letters and symbols in `fulbst8.mf` — bold, strict

E Index

The Numbers written in *italic* refer to the pages, where a macros usage is descriped, while those in `typewrite` refer to line numbers in the files, mentioned before, where the definition is, while *slanted* shows the places it is used. Normal letters refer to pages, wether it be descriptions or usage.

F Change History

Below is listed the changes made to the `universal` font and the LaTeX support package `uni`, from version 1.0 to 2.0.