Linear Logic symbols for Computer Modern

Emmanuel Beffara

April 20, 2010

1 Documentation

1.1 Installation

The installation consists in copying the `.mf` files from the `mf` subdirectory somewhere in Metafont's search path and the `.sty` and `.fd` files from the `latex` subdirectory somewhere in LaTeX's search path.

Call `TEXMF` the base directory of your installation (on Unix this should be something like `/usr/share/texmf` or `~/texmf` for a single user installation). Then copy the directories as follows:

- copy `mf` as `TEXMF/fonts/source/public/cmll`
- copy `latex` as `TEXMF/tex/latex/cmll`

If you want to install the fonts in Type1 format:

- copy `type1` as `TEXMF/fonts/type1/public/cmll`
- copy `tfm` as `TEXMF/fonts/tfm/public/cmll`
- copy `cmll.map` as `TEXMF/fonts/map/dvips/misc/cmll.map`

You may have to update TeX's databases by running `texhash` or `mktexlsr` (this is for Kpathsea-based distributions like teTeX) or a similar command. You may also have to register the map file for the Type1 fonts using `updmap`, depending on your installation. Linux users using Debian or Ubuntu can avoid all this by using the package at `http://iml.univ-mrs.fr/~beffara/soft/` (updated more often than TexLive).

1.2 Package loading

The package is loaded by simply saying `\usepackage{cmll}`. The variant of the font that should be used is chosen using the following package options:

`cm`	Computer Modern serif
`ss`	Computer Modern sans serif
`euler`	AMS Euler
`emu`	emulation mode (see section 1.4)
`auto`	automatic selection among the above (default, see section 1.4)

1.3 Symbols

The `cmll` font defines a handful of symbols useful in linear logic that were not defined in other fonts and packages.

Unary operators:

$\oc$	`\oc`	$\wn$	`\wn`
$\shpos$	`\shpos`	$\shneg$	`\shneg`
$\shift$	`\shift`		

Binary operators:

$\parr$	`\parr`	$\invamp$	`\invamp`
$\with$	`\with`		

Large operators:

$\bigparr$	`\bigparr`	$\biginvamp$	`\biginvamp`
$\bigwith$	`\bigwith`		

Binary relations:

$\coh$	`\coh`	$\scoh$	`\scoh`
$\incoh$	`\incoh`	$\sincoh$	`\sincoh`
$\Perp$	`\Perp`	$\simperp$	`\simperp`
$\multimapinv$	`\multimapinv`	$\multimapboth$	`\multimapboth`
$\nmultimap$	`\nmultimap`	$\nmultimapinv$	`\nmultimapinv`
$\nmultimapboth$	`\nmultimapboth`		

Letter-like symbols:

$\Bot$	`\Bot`	$\simbot$	`\simbot`

The symbols !, ? and & are actually characters from the standard Computer Modern fonts declared with a new math code to get proper spacing. See the following examples:

`A \& B`	$A\&B$	`A \with B`	$A \with B$
`A = !B`	$A =!B$	`A = \oc B`	$A = \oc B$
`A = ?B`	$A =?B$	`A = \wn B`	$A = \wn B$

The names `\parr` and `\invamp` are synonyms, the names `\bigparr` and `\biginvamp` too. The macros `\biginvampemu` and `\bigwithemu` provide emulations for the `\biginvamp` and `\bigwith` symbols, built using the normal symbols at different sizes (using the `relsize` package). In emulation mode (i.e. with the package option `emu`) the names `\biginvamp` and `\bigwith` are synonyms for them.

1.4 Emulation and automatic selection

If you use a math font family other than Computer Modern or Euler, you may have an ampersand symbol that does not match the reversed ampersand from any variant of CMLL. In this case, the `cmll` package provides a “poor man” version of the reversed ampersand and large ampersands, built using the standard `\&` symbol. Also provided are the "big" variants of `\with` and `\invamp` built using the normal version at a different size (using the `relsize` package). These versions can be used explicitly under the names `\invampemu`, `\bigwithemu` and `\biginvampemu`.

In the emulation mode, that is when the package is loaded with the `emu` option, these variants are used instead of the symbols provided by the font. Moreover, in emulation mode, any symbol that already exists is preserved by `cmll`, otherwise all symbols are redefined. This is useful for instance with `txfonts`, which already provides `\invamp` and `\multimapboth`. Note that this feature works only if the package `cmll` is loaded *after* loading any package that might define one of the symbols provided by CMLL.

The automatic selection works as follows:

- if `euler` is loaded, use the Euler variant,
- if `txfonts` or `pxfonts` is loaded, use the emulation mode,
- if the default font is `cmss`, use the `ss` variant,
- otherwise use the `cm` variant.

1.5 History

2010-04-20 New symbols: `\multimapinv`, `\nmultimap`, `\nmultimapinv`, `\nmultimapboth`.

2009-01-23 New symbols: `\multimapboth`, `\Perp`, `\Bot`, `\simperp`, `\simbot`. Synonyms `\invamp` and `\biginvamp` added for compatibility. New package option `emu`. Various fixes and code improvements.

2006-02-22 First public release.

2 Files

2.1 This document

```
⟨*driver⟩
\documentclass{ltxdoc}
\usepackage{array,cmll}
\newenvironment{symbols}[1]{%
  \par%
  \def\dosymbol##1{\leavevmode\hbox to .5\textwidth{%
    \kern.25\textwidth \hbox to 2em{\hss$##1$\hfil}%
    \texttt{\string##1}\hss}\penalty10}%
  \flushleft%
  #1\strut\\}{\endflushleft}
\begin{document}
\DocInput{cmll.dtx}
\end{document}
⟨/driver⟩
```

2.2 Font definitions

The font definition file is deduced from the ones for Computer Modern. We provide an NFSS entry named `cmllr` in medium and bold extended versions.

```
⟨*ucmllr⟩
\DeclareFontShape{U}{cmllr}{m}{n}{%
      <5><6><7><8><9>gen*cmllr%
      <10><10.95>cmllr10%
      <12><14.4>cmllr12%
      <17.28->cmllr17%
      }{}
\DeclareFontShape{U}{cmllr}{bx}{n}{%
      <5><6><7><8><9>gen*cmllbx%
      <10><10.95>cmllbx10%
      <12->cmllbx12%
      }{}
⟨/ucmllr⟩
```

The following is a definition for the sans-serif version, named `cmllss`.

```
⟨*ucmllss⟩
\DeclareFontShape{U}{cmllss}{m}{n}{%
      <-8>cmllss8%
      <9>cmllss9%
      <10>cmllss10
      <12><14.4>cmllss12%
      <17.28->cmllss17%
      }{}
\DeclareFontShape{U}{cmllss}{bx}{n}{%
      <->cmllssbx10}{}
⟨/ucmllss⟩
```

The following is a definition for the Euler-style version, named `eull`.

```
⟨*ueull⟩
\DeclareFontShape{U}{eull}{m}{n}{%
      <5><6><7><8><9>gen*eullr%
      <10->eullr10}{}
\DeclareFontShape{U}{eull}{bx}{n}{%
      <5><6><7><8><9>gen*eullbx%
      <10->eullbx10}{}
⟨/ueull⟩
```

2.3 The package

```
⟨*package⟩
\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{cmll}[2010/04/20 Linear Logic symbols for Computer Modern]
```

With the package option `emu`, symbols already defined are used and `\invamp` (if undefined) is made by rotation using an ampersand. This requires the `graphicx` package.

```
\let\cmll@ifemu=\iffalse
\DeclareOption{emu}{\let\cmll@ifemu=\iftrue}
```

The font is declared as a symbol font named `llsymbols`, in normal and bold versions. We provide package options to switch between the standard, sans-serif and Euler-style variants.

```
\def\cmll@style{auto}
\DeclareOption*{\edef\cmll@style{\CurrentOption}}
\ProcessOptions\relax
```

The following code is used to detect which family should be used. Euler is detected if its package is loaded, emulation is activated for txfonts and pxfonts, sans-serif is detected by looking at the default font family name.

```
\RequirePackage{ifthen}
\def\cmll@use@auto{%
  \@ifpackageloaded{euler}{%
    \def\cmll@style{euler}}{%
  \@ifpackageloaded{txfonts}{%
    \let\cmll@ifemu=\iftrue%
    \def\cmll@style{cm}}{%
  \@ifpackageloaded{pxfonts}{%
    \let\cmll@ifemu=\iftrue%
    \def\cmll@style{cm}}{%
  \ifthenelse{\equal{\rmdefault}{cmss}}{%
    \def\cmll@style{ss}}{%
    \def\cmll@style{cm}}%
  }}}%
  \csname cmll@use@\cmll@style\endcsname}
```

The following macros are used to set up the font families and symbols from other fonts.

```
\def\cmll@use@cm{%
  \DeclareSymbolFont{llsymbols}{U}{cmllr}{m}{n}%
  \SetSymbolFont{llsymbols}{bold}{U}{cmllr}{bx}{n}%
  \DeclareMathSymbol{\with}{\mathbin}{operators}{`\&}%
  \DeclareMathSymbol{\oc}{\mathord}{operators}{`!}%
  \DeclareMathSymbol{\wn}{\mathord}{operators}{`?}}
```

Here is the sans-serif variant.

```
\def\cmll@use@ss{%
  \DeclareSymbolFont{llsymbols}{U}{cmllss}{m}{n}%
  \SetSymbolFont{llsymbols}{bold}{U}{cmllss}{bx}{n}%
  \DeclareMathSymbol{\with}{\mathbin}{operators}{`\&}%
  \DeclareMathSymbol{\oc}{\mathord}{operators}{`!}%
  \DeclareMathSymbol{\wn}{\mathord}{operators}{`?}}
```

And here is the Euler variant.

```
\def\cmll@use@euler{%
  \DeclareSymbolFont{llsymbols}{U}{eull}{m}{n}%
  \SetSymbolFont{llsymbols}{bold}{U}{eull}{bx}{n}%
  \DeclareMathSymbol{\with}{\mathbin}{EulerFraktur}{"26}%
  \DeclareMathSymbol{\oc}{\mathord}{EulerFraktur}{"21}%
  \DeclareMathSymbol{\wn}{\mathord}{EulerFraktur}{"3F}}
```

Finally we activate the proper variant.

```
\csname cmll@use@\cmll@style\endcsname
```

In some cases it is preferable to emulate the `\invamp` by rotating the ampersand symbol. Here is a robust definition of this rotation:

```
\newcommand\invampemu{%
  \mathbin{\mathchoice%
    {\rotatebox[origin=c]{180}{$\&$}}%
    {\rotatebox[origin=c]{180}{$\&$}}%
    {\rotatebox[origin=c]{180}{$\scriptstyle\&$}}%
    {\rotatebox[origin=c]{180}{$\scriptscriptstyle\&$}}%
  }}
```

We may also want to emulate the big versions of the ampersand. In this case we change ther text size appropriately for each style, using the `relsize` package. This version is reasonable at normal size and becomes approximative when math is composed in small or large sizes.

```
\newcommand\bigwithemu{%
  \mathop{\mathchoice%
    {\vcenter{\hbox{\relsize{+4}$\&$}}}%
    {\vcenter{\hbox{\relsize{+2}$\&$}}}%
    {\vcenter{\hbox{\relsize{+0.5}$\&$}}}%
    {\vcenter{\hbox{\relsize{-1}$\&$}}}%
  }}
```

For the large inversed ampersand, we call the `\invamp` macro, which is supposed to be defined (either as a proper character or as an emulation as above). When it is emulated, this makes two nested `\mathchoice`s, which is not very efficient.

```
\newcommand\biginvampemu{%
  \mathop{\mathchoice%
    {\vcenter{\hbox{\relsize{+4}$\invamp$}}}%
    {\vcenter{\hbox{\relsize{+2}$\invamp$}}}%
    {\vcenter{\hbox{\relsize{+0.5}$\invamp$}}}%
    {\vcenter{\hbox{\relsize{-1}$\invamp$}}}}}
```

The actual commands `\invamp`, `\bigwith` and `\biginvamp` can be defined in various ways depending on the setup.

```
\cmll@ifemu
  \@ifundefined{invamp}{%
    \RequirePackage{graphicx}%
    \let\invamp=\invampemu
  }{}%
  \RequirePackage{relsize}%
  \let\bigwith=\bigwithemu
  \let\biginvamp=\biginvampemu
```

Already existing symbols are preserved in emulation mode.

```
  \def\cmll@declare@symbol#1#2#3#4{%
    \@ifundefined{#1}{%
      \expandafter\DeclareMathSymbol%
      \expandafter{\csname#1\endcsname}{#2}{#3}{#4}}{}}
```

```
\else
  \def\cmll@declare@symbol#1#2#3#4{%
    \expandafter\DeclareMathSymbol%
    \expandafter{\csname#1\endcsname}{#2}{#3}{#4}}
\fi
```

The new symbol definitions are the same for all variants.

```
\cmll@declare@symbol{invamp}{\mathbin}{llsymbols}{0}
\let\parr=\invamp
\cmll@declare@symbol{shpos}{\mathord}{llsymbols}{1}
\cmll@declare@symbol{shneg}{\mathord}{llsymbols}{2}
\cmll@declare@symbol{shift}{\mathord}{llsymbols}{3}
\cmll@declare@symbol{coh}{\mathrel}{llsymbols}{4}
\cmll@declare@symbol{scoh}{\mathrel}{llsymbols}{5}
\cmll@declare@symbol{incoh}{\mathrel}{llsymbols}{6}
\cmll@declare@symbol{sincoh}{\mathrel}{llsymbols}{7}
\cmll@declare@symbol{bigwith}{\mathop}{llsymbols}{8}
\cmll@declare@symbol{biginvamp}{\mathop}{llsymbols}{10}
\let\bigparr=\biginvamp
\cmll@declare@symbol{multimapboth}{\mathrel}{llsymbols}{12}
\cmll@declare@symbol{Bot}{\mathord}{llsymbols}{13}
\cmll@declare@symbol{Perp}{\mathrel}{llsymbols}{13}
\cmll@declare@symbol{simbot}{\mathord}{llsymbols}{14}
\cmll@declare@symbol{simperp}{\mathrel}{llsymbols}{14}
\cmll@declare@symbol{multimapinv}{\mathrel}{llsymbols}{15}
\cmll@declare@symbol{nmultimap}{\mathrel}{llsymbols}{16}
\cmll@declare@symbol{nmultimapinv}{\mathrel}{llsymbols}{17}
\cmll@declare@symbol{nmultimapboth}{\mathrel}{llsymbols}{18}
⟨/package⟩
```