Configurable LTL math operators with the ltl package*

Malte Schmitz
malte@schmitz-sh.de

May 4, 2015

Abstract

The ltl package contains a set of macros for typesetting operators used in formulas of the linear temporal logic (LTL) in the Manna/Pnueli or modern character based notation. This package provides options to switch between characters and symbols as operators. As a backend for the symbols either the ltlfonts by Matteo Slanina or Ti*k*Z drawings provided in this package can be used.

1 Introduction

Put text here.

2 Usage

Put text here.

3 Showcase

3.1 Direct Symbol Usage

circle: $\bigcirc x$
circle with minus: $\ominus x$
circle with tilde: $\circledsim x$

*This document corresponds to ltl v0.3, dated 2015/04/03.

diamond: $\Diamond x$
diamond with minus: $\diamondminus x$
square: $\Box x$
square with minus: $\boxminus x$

3.2 Semantic Interface Usage with symbols and without weakindex

until: $x \mathcal{U} x$
weak until: $x \mathcal{W} x$
release: $x \mathcal{R} x$
since: $x \mathcal{S} x$
weak since: $x \mathcal{B} x$ (back: $x \mathcal{B} x$)
globally: $\Box x$
finally: $\Diamond x$ (eventually: $\Diamond x$)
next: $\bigcirc x$
weak next: $\overline{\bigcirc} x$
previous: $\ominus x$ (prev: $\ominus x$)
weak previous: $\overline{\ominus} x$ (weak prev: $\overline{\ominus} x$)
trigger: $x \mathcal{T} y$ (past release: $x \mathcal{T} y$)
past globally: $\boxminus x$
past finally: $\diamondminus x$ (once: $\diamondminus x$)
predict: $\rhd x$
record: $\lhd x$
stop: $\mathrm{stop}\, x$
implication: $x \to x$ (imp: $x \to x$)
equal: $x \leftrightarrow x$ (equ: $x \leftrightarrow x$)
and: $x \wedge x$
or: $x \vee x$
false: false
true: true
not: $\neg$

3.3 Semantic Interface Usage with symbols and with weakindex

until: $x \mathcal{U} x$
weak until: $x \mathcal{U}_{\mathrm{w}} x$
release: $x \mathcal{R} x$
since: $x \mathcal{S} x$
weak since: $x \mathcal{S}_{\mathrm{w}} x$ (back: $x \mathcal{S}_{\mathrm{w}} x$)
globally: $\Box x$
finally: $\Diamond x$ (eventually: $\Diamond x$)
next: $\bigcirc x$
weak next: $\bigcirc_{\mathrm{w}} x$

previous: $\ominus x$ (prev: $\ominus x$)
weak previous: $\ominus_{\mathrm{w}} x$ (weak prev: $\ominus_{\mathrm{w}} x$)
trigger: $x \mathcal{T} y$ (past release: $x \mathcal{T} y$)
past globally: $\boxminus x$
past finally: $\diamondminus x$ (once: $\diamondminus x$)
predict: $\rhd x$
record: $\lhd x$
stop: $\mathrm{stop}\, x$
implication: $x \to x$ (imp: $x \to x$)
equal: $x \leftrightarrow x$ (equ: $x \leftrightarrow x$)
and: $x \wedge x$
or: $x \vee x$
false: false
true: true
not: $\neg$

3.4 Semantic Interface Usage with characters and without weakindex

until: $x \mathcal{U} x$
weak until: $x \mathcal{W} x$
release: $x \mathcal{R} x$
since: $x \mathcal{S} x$
weak since: $x \mathcal{B} x$ (back: $x \mathcal{B} x$)
globally: $\mathcal{G} x$
finally: $\mathcal{F} x$ (eventually: $\mathcal{F} x$)
next: $\mathcal{X} x$
weak next: $\overline{\mathcal{X}} x$
previous: $\mathcal{P} x$ (prev: $\mathcal{P} x$)
weak previous: $\overline{\mathcal{P}} x$ (weak prev: $\overline{\mathcal{P}} x$)
trigger: $x \mathcal{T} y$ (past release: $x \mathcal{T} y$)
past globally: $\mathcal{H} x$
past finally: $\mathcal{O} x$ (once: $\mathcal{O} x$)
predict: $\rhd x$
record: $\lhd x$
stop: $\mathrm{stop}\, x$
implication: $x \to x$ (imp: $x \to x$)
equal: $x \leftrightarrow x$ (equ: $x \leftrightarrow x$)
and: $x \wedge x$
or: $x \vee x$
false: false
true: true
not: $\neg$

3.5 Semantic Interface Usage with characters and with weakindex

until: $x \mathcal{U} x$
weak until: $x \mathcal{U}_{\mathrm{w}} x$
release: $x \mathcal{R} x$
since: $x \mathcal{S} x$
weak since: $x \mathcal{S}_{\mathrm{w}} x$ (back: $x \mathcal{S}_{\mathrm{w}} x$)
globally: $\mathcal{G} x$
finally: $\mathcal{F} x$ (eventually: $\mathcal{F} x$)
next: $\mathcal{X} x$
weak next: $\mathcal{X}_{\mathrm{w}} x$
previous: $\mathcal{P} x$ (prev: $\mathcal{P} x$)
weak previous: $\mathcal{P}_{\mathrm{w}} x$ (weak prev: $\mathcal{P}_{\mathrm{w}} x$)
trigger: $x \mathcal{T} y$ (past release: $x \mathcal{T} y$)
past globally: $\mathcal{H} x$
past finally: $\mathcal{O} x$ (once: $\mathcal{O} x$)
predict: $\triangleright x$
record: $\triangleleft x$
stop: $\mathrm{stop}\, x$
implication: $x \to x$ (imp: $x \to x$)
equal: $x \leftrightarrow x$ (equ: $x \leftrightarrow x$)
and: $x \wedge x$
or: $x \vee x$
false: false
true: true
not: $\neg$

4 Installing ltlfonts

ltlfonts is a free font developed by Matteo Slanina containing mathematical symbols for typesetting formulas of linear temporal logic (LTL) in the Manna/Pnueli notation.

If ltlfonts is loaded this package will use the symbols of LTLFonts instead of the TikZ drawings provided in this package to define its macros.

ltlfonts can be downloaded from `http://theory.stanford.edu/~matteo/ltlfonts/`

To use it with TeX, LaTeX and dvips you can just copy all the files from the zip file (ignoring all folders) next to your tex file and run

```
latex yourfile.tex
dvips -u +ltlfonts.map yourfile.tex</pre>
```

To use it with PdfTeX or PdfLaTeX you have to install the font into your LaTeX system first. To do so follow these steps:

Find your local TeX tree (texmf-local). In TeX Live and MaxTeX the default is `/usr/local/texlive/texmf-local`. In MicTeX the default is `%APPDATA%\MikTeX\2.9`

Copy these files to the given directories:

- `fonts/afm/matteo/ltlfonts/ltlfonts.afm`
- `fonts/map/dvips/matteo/ltlfonts.map`
- `fonts/tfm/matteo/ltlfonts/ltlfonts.tfm`
- `fonts/type1/matteo/ltlfonts/ltlfonts.pfb`
- `fonts/type1/matteo/ltlfonts/ltlfonts.pfm`
- `tex/latex/ltlfonts/ltlfonts.sty`
- `tex/latex/ltlfonts/Ultlfonts.fd`

After getting your new files into their proper location, you must update the so-called "TeX filename database".

- on MikTeX run `initexmf --update-fndb`
- on MacTeX run `mktexlsr`
- on TeXLive run `mktexlsr`

After recording the new files, the last step is to update so-called "map" files with the information about the new font.

On MikTex run `initexmf --edit-config-file updmap`. A configuration file gets opened in your default editor. Add the line

```
Map ltlfonts.map
```

(Don't worry if the file is initially empty.)

On TeX Live and MacTeX run

```
updmap-sys --enable Map=ltlfonts.map
```

ltlfonts is now installed and can be used. As a test you can compile this LaTeX code:

```
\documentclass{article}
\usepackage{ltlfonts}
\begin{document}
LTLFonts provides some nice boxes and circles. For example:
\[ \LTLsquare \LTLdiamond \LTLcircle a, \LTLcircleminus
\LTLdiamondminus \LTLsquareminus \LTLcircletilde b, \LTLsquarehat
\LTLdiamondminushat c \]
\end{document}
```

The ltlfonts package provides a set of LTL symbols. It does not provide any semantically named commands and it does not allow switching to characeter based LTL symbols. It just lets you use the LTL symbols of the LTLFonts font in a LaTeX document.

5 Implementation

We start loading some required packages. xkeyval is used to handle the options. amsmath provides `\operatorname` which handles the spacing around the declared operators automatically, amssymb provides some additional symbols some operator macros are based on and Ti*k*Z is used to draw the LTL symbols.

```
\RequirePackage{xkeyval}
\RequirePackage{amsmath}
\RequirePackage{amssymb}
\RequirePackage{tikz}
```

5.1 Ti*k*Z symbol operators

We now define the symbols using Ti*k*Z. The macros are named after the macros defined by the LTLFonts package. Every macro is only defined unless it is already defined. This way we use the macros of the LTLFonts package if it is loaded. The `\tikz` command with `baseline` option is used to create "inline" graphics as this technique called in the manual:

> Normally, the lower end of the picture is put on the baseline of the surrounding text. Using this option, you can specify that the picture should be raised or lowered such that the given height is on the baseline.

The value defaults to `0pt`.

`LTLcircle` Draws a circle, e.g. for the LTL next operator.

```
\ifx\LTLcircle\undefined
  \DeclareRobustCommand{\LTLcircle}{\operatorname{%
    \tikz[baseline,rounded corners=0pt,shorten >=0pt,shorten <=0pt]{
      \draw[line width=.12ex]
        (0,.6ex) circle (.8ex);
    }}}{}
\fi
```

`LTLcircleminus` Draws a circle with minus in it, e.g. for LTL previous resp. past next operator.

```
\ifx\LTLcircleminus\undefined
```

```
  \DeclareRobustCommand{\LTLcircleminus}{\operatorname{%
    \tikz[baseline,rounded corners=0pt,shorten >=0pt,shorten <=0pt]{
      \draw[line width=.12ex]
        (0,.6ex) circle (.8ex);
      \draw[line width=.09ex,line cap=round]
        (-.4ex,.6ex) -- (.4ex,.6ex);
    }}}{}
\fi
```

`LTLcircletilde` Draw a circle with tilde in it, e.g. for LTL weak previous resp weak past next operator.

```
\ifx\LTLcircletilde\undefined
  \DeclareRobustCommand{\LTLcircletilde}{\operatorname{%
    \tikz[baseline,rounded corners=0pt,shorten >=0pt,shorten <=0pt]{
      \draw[line width=.12ex]
        (0,.6ex) circle (.8ex);
      \draw[line width=.09ex,line cap=round,rounded corners=0.2ex]
        (-.4ex,.55ex) -- (-.2ex,.8ex) -- (.2ex,.4ex) -- (.4ex,.65ex);
    }}}{}
\fi
```

`LTLdiamond` Draws diamond, e.g. for the LTL finally resp. eventually operator.

```
\ifx\LTLdiamond\undefined
  \DeclareRobustCommand{\LTLdiamond}{\operatorname{%
    \tikz[baseline,rounded corners=0pt,shorten >=0pt,shorten <=0pt]{
      \draw[line width=.12ex,line join=round]
        (0ex,.6ex) -- (.95ex,1.55ex) -- (1.9ex,.6ex) -- (.95ex,-.35ex) -- cycle;
    }}}{}
\fi
```

`LTLdiamondminus` Draws a diamond with minus in it, e.g. for the LTL past finally resp. past eventually resp. once operator.

```
\ifx\LTLdiamondminus\undefined
  \DeclareRobustCommand{\LTLdiamondminus}{\operatorname{%
    \tikz[baseline,rounded corners=0pt,shorten >=0pt,shorten <=0pt]{
      \draw[line width=.12ex,line join=round]
        (0ex,.6ex) -- (.95ex,1.55ex) -- (1.9ex,.6ex) -- (.95ex,-.35ex) -- cycle;
      \draw[line width=.09ex,line cap=round]
        (.5ex,.6ex) -- (1.3ex,.6ex);
    }}}{}
\fi
```

`LTLsquare` Draws a square, e.g. for the LTL globally operator.

```
\ifx\LTLsquare\undefined
  \DeclareRobustCommand{\LTLsquare}{\operatorname{%
```

```
    \tikz[baseline,rounded corners=0pt,shorten >=0pt,shorten <=0pt]{
      \draw[line width=.12ex,line join=round]
        (0ex,-.2ex) -- (0ex,1.3ex) -- (1.5ex,1.3ex) -- (1.5ex,.-.2ex) -- cycle;
    }}}{}
\fi
```

`LTLsquareminus` Draws a square with minus in it, e.g. for the LTL past globally operator.

```
\ifx\LTLsquareminus\undefined
  \DeclareRobustCommand{\LTLsquareminus}{\operatorname{%
    \tikz[baseline,rounded corners=0pt,shorten >=0pt,shorten <=0pt]{
      \draw[line width=.12ex,line join=round]
        (0ex,-.2ex) -- (0ex,1.3ex) -- (1.5ex,1.3ex) -- (1.5ex,.-.2ex) -- cycle;
      \draw[line width=.09ex,line cap=round]
        (.35ex,.6ex) -- (1.15ex,.6ex);
    }}}{}
\fi

\DeclareMathOperator{\LTLcirclew}{\LTLcircle_w}
\DeclareMathOperator{\LTLcircleminusw}{\LTLcircleminus_w}
\DeclareMathOperator{\LTLcircleoverline}{\overline{\LTLcircle}}
\DeclareMathOperator{\LTLcircleminusoverline}{\overline{\LTLcircleminus}}
```

5.2 Character operators

```
\newcommand{\ltl@operatorfont@mathcal}[1]{\mathcal{#1}}
\let\ltl@operatorfont\relax

\DeclareMathOperator{\LTLu}{\ltl@operatorfont{U}}
\DeclareMathOperator{\LTLuw}{\ltl@operatorfont{U}_w}
\DeclareMathOperator{\LTLw}{\ltl@operatorfont{W}}
\DeclareMathOperator{\LTLr}{\ltl@operatorfont{R}}
\DeclareMathOperator{\LTLs}{\ltl@operatorfont{S}}
\DeclareMathOperator{\LTLsw}{\ltl@operatorfont{S}_w}
\DeclareMathOperator{\LTLb}{\ltl@operatorfont{B}}
\DeclareMathOperator{\LTLf}{\ltl@operatorfont{F}}
\DeclareMathOperator{\LTLg}{\ltl@operatorfont{G}}
\DeclareMathOperator{\LTLx}{\ltl@operatorfont{X}}
\DeclareMathOperator{\LTLxw}{\ltl@operatorfont{X}_w}
\DeclareMathOperator{\LTLwx}{\overline{\ltl@operatorfont{X}}}
\DeclareMathOperator{\LTLp}{\ltl@operatorfont{P}}
\DeclareMathOperator{\LTLpw}{\ltl@operatorfont{P}_w}
\DeclareMathOperator{\LTLwp}{\overline{\ltl@operatorfont{P}}}
\DeclareMathOperator{\LTLt}{\ltl@operatorfont{T}}
\DeclareMathOperator{\LTLh}{\ltl@operatorfont{H}}
\DeclareMathOperator{\LTLo}{\ltl@operatorfont{O}}
```

5.3 Semantic Macros for LTL Operators

```
\let\LTLuntil\LTLu
\let\LTLrelease\LTLr
\let\LTLsince\LTLs
\let\LTLtrigger\LTLt
\let\LTLpastrelease\LTLtrigger

\let\LTLweakuntil\relax
\let\LTLweaksince\relax
\let\LTLglobally\relax
\let\LTLfinally\relax
\let\LTLnext\relax
\let\LTLweaknext\relax
\let\LTLprevious\relax
\let\LTLweakprevious\relax
\let\LTLpastglobally\relax
\let\LTLpastfinally\relax

\newcommand{\ltl@define}{%
  \ifKV@ltl@mathcal%
    \let\ltl@operatorfont\ltl@operatorfont@mathcal
  \else
    \let\ltl@operatorfont\relax
  \fi
  \ifKV@ltl@weakindex%
    \let\LTLweakuntil\LTLuw
    \let\LTLweaksince\LTLsw
  \else
    \let\LTLweakuntil\LTLw
    \let\LTLweaksince\LTLb
  \fi
  \ifKV@ltl@characters%
    \let\LTLglobally\LTLg
    \let\LTLfinally\LTLf
    \let\LTLnext\LTLx
    \let\LTLprevious\LTLp
    \ifKV@ltl@weakindex
      \let\LTLweaknext\LTLxw
      \let\LTLweakprevious\LTLpw
    \else
      \let\LTLweaknext\LTLwx
      \let\LTLweakprevious\LTLwp
    \fi
    \let\LTLpastglobally\LTLh
    \let\LTLpastfinally\LTLo
  \else
    \let\LTLglobally\LTLsquare
    \let\LTLfinally\LTLdiamond
    \let\LTLnext\LTLcircle
    \let\LTLprevious\LTLcircleminus
    \ifKV@ltl@weakindex
```

```
      \let\LTLweakprevious\LTLcircleminusw
      \let\LTLweaknext\LTLcirclew
    \else
      \let\LTLweakprevious\LTLcircleminusoverline
      \let\LTLweaknext\LTLcircleoverline
    \fi
    \let\LTLpastglobally\LTLsquareminus
    \let\LTLpastfinally\LTLdiamondminus
  \fi
  \let\LTLeventually\LTLfinally
  \let\LTLprev\LTLprevious
  \let\LTLweakprev\LTLweakprevious
  \let\LTLonce\LTLpastfinally
  \let\LTLback\LTLweaksince
}
```

5.4 Option Handling

We now declare the xkeyval boolean keys. The optional parameter `true` is the default value that gets used if you only specify the key with a value.

```
\define@boolkeys{ltl}{characters,mathcal,weakindex}[true]
```

This macro processes the keys and values passed by the user to the class or package.

```
\ProcessOptionsX<ltl>
\ltl@define
```

`ltlsetup`

```
\newcommand{\ltlsetup}[1]{%
  \setkeys{ltl}{#1}%
  \ltl@define%
}
```

5.5 Further macros

auxilaries

```
\DeclareMathOperator{\LTLpredict}{\rhd}
\DeclareMathOperator{\LTLrecord}{\lhd}
\DeclareMathOperator{\LTLstop}{stop}
\DeclareMathOperator{\LTLimplication}{\rightarrow}
\DeclareMathOperator{\LTLequivalent}{\leftrightarrow}
\DeclareMathOperator{\LTLand}{\wedge}
\DeclareMathOperator{\LTLor}{\vee}
\DeclareMathOperator{\LTLfalse}{false}
\DeclareMathOperator{\LTLtrue}{true}
```

```
168 \DeclareMathOperator{\LTLnot}{\neg}
169 \let\LTLimp\LTLimplication
170 \let\LTLequ\LTLequivalent
```

Change History

v0.1

General: Initial version 1

v0.2

General: Macro collection transformed into a LaTeX package. . . 1

Index

Numbers written in italic refer to the page where the corresponding entry is described; numbers underlined refer to the code line of the definition; numbers in roman refer to the code lines where the entry is used.