Actuarial angle symbol for life contingencies and financial mathematics*

Vincent Goulet†
vincent.goulet@act.ulaval.ca

June 13, 2019

Abstract

The package actuarialangle provides commands to typeset the "angle" symbol denoting a duration n in actuarial notation, as in $\overline{n}|$, and an overhead angle bracket, as in $\langle\overline{xy}\rangle$.

1 Introduction

This package defines commands to typeset two symbols used in actuarial notation for life contingencies and financial mathematics. The first is the "angle" denoting a duration in the present value of an insurance or annuity: $\overline{n}|$. The second is an overhead angle bracket (or "roof") used to emphasize joint status when ambiguity is possible: $\langle\overline{xy}\rangle$. The bracket is normally used with a precedence number above. Facilities to position such numbers are provided by the package actuarialsymbol (Beauchemin and Goulet, 2017).

For additional details on actuarial notation for life contingencies, see Bowers et al. (1997).

2 Package options

The package offers the following options:

*This document corresponds to actuarialangle v2.1, dated 2019/06/13.
†École d'actuariat, Université Laval.

`thinspace` insert a thin space of 1mu between the material under the angle and the right descender; this is the default starting with version 2.0 of the package;

`thickspace` insert a thicker space of 2mu between the material under the angle and the right descender; this was the value used in versions of the package prior to 2.0;

`nobracket` do not define command `\overanglebracket` and its alias `\group`; this option also prevents loading of package pict2e.

We provide the option `nobracket` in case the bracket symbol is not needed and pict2e (Gäßlein et al., 2016) interferes with other packages. Loading the package with

```
\usepackage[thickspace,nobracket]{actuarialangle}
```

yields the behaviour of versions prior to 2.0, but for the defunct features mentioned in section 4.

3 Package features

`\actuarialangle` In math mode, the command

```
\actuarialangle{⟨duration⟩}
```

composes an angle symbol around ⟨*duration*⟩. This is the "raw" command that does not insert any space between ⟨*duration*⟩ and the right descender of the angle. The symbol scales gracefully if the command is used outside of a first-level subscript.

```
\actuarialangle{n} \quad
a_{\actuarialangle{n}}
```

$\overline{n|} \quad a_{\overline{n|}}$

`\angl` `\angln` `\anglr` `\anglk` Users are expected to typeset angle symbols with the command

```
\angl{⟨duration⟩}
```

In contrast to `\actuarialangle`, this command inserts some thin space (by default or with package option `thinspace`) or thick space (with package option `thickspace`) between {⟨*duration*⟩} and the right descender.

```
% with option thinspace:
\angl{n} \quad a_{\angl{n}}
```
$\overline{n|} \quad a_{\overline{n|}}$

```
% with option thickspace:
\angl{n} \quad a_{\angl{n}}
```
$\overline{n|} \quad a_{\overline{n|}}$

Commands `\angln`, `\anglr` and `\anglk` are shortcuts for the common cases `\angl{n}`, `\angl{r}` and `\angl{k}`, respectively.

`\overanglebracket` `\group` The command

`\overanglebracket{`⟨*statuses*⟩`}`

composes an angle bracket ("roof") above ⟨*statuses*⟩. The rule thickness and spacing relative to the statuses match those of the angle symbol. The command `\group` is a convenient alias for `\overanglebracket`.

```
\group{xy} \quad
A_{\group{xy}:\angln}
```
$\overbrace{xy}^{\cdot} \quad A_{\overbrace{xy}^{\cdot}:\overline{n|}}$

4 Defunct features

Versions prior to 2.0 of the package included the undocumented commands

`\topprecedence` (with alias `\lift`)
`\vartopprecedence`
`\bottomprecedence`
`\varbottomprecedence`

to typeset precedence numbers above and below statuses in subscript of an actuarial symbol. These features have been moved — and improved on the way — to package actuarialsymbol (Beauchemin and Goulet, 2017).

A Implementation

This appendix contains the annotated source code of the package. Most readers can stop reading here.

A.1 Package options

`\ifacta@thinspace` `\ifacta@nobracket` Two flags are defined to keep track of the spacing between the material under the angle and right descender, and whether or not the package should define the command `\overanglebracket` and load package pict2e.

```
\newif\ifacta@thinspace \acta@thinspacetrue
\newif\ifacta@bracket   \acta@brackettrue
```

\DeclareOption

Declaration of the package options and processing. Defaults are `thinspace` and to define the bracket.

```
\DeclareOption{thinspace}{\acta@thinspacetrue}
\DeclareOption{thickspace}{\acta@thinspacefalse}
\DeclareOption{nobracket}{\acta@bracketfalse}
\ProcessOptions
```

A.2 Variable extra space, rule thickness and vertical gap

\acta@overbarkern@fontdimen
\acta@overbarkern@family
\acta@overbarrule@fontdimen
\acta@overbarrule@family
\acta@overbarvgap@fontdimen
\acta@overbarvgap@family

We first deal with Type 1 math fonts. We define the `\fontdimen`'s and families used for the thickness of the `\overline` rule and the amount of vertical gap between the rule and the content, as well as the extra white space above the rule.

```
\def\acta@overbarkern@fontdimen{8}
\def\acta@overbarkern@family{\thr@@}
\def\acta@overbarrule@fontdimen{8}
\def\acta@overbarrule@family{\thr@@}
\def\acta@overbarvgap@fontdimen{8}
\def\acta@overbarvgap@family{\thr@@}
```

\acta@overbarkern

We want a macro to take a math style, e.g., `\displaystyle`, and then to expand into `\fontdimen8\textfont3` (for Type 1 fonts). This will serve as the extra space. The name "overbarkern" comes from the corresponding LuaTeX primitive and OpenType Math table entry.

```
\def\acta@overbarkern#1{%
  \fontdimen\acta@overbarkern@fontdimen
  \ifx#1\displaystyle
    \textfont
  \else
    \ifx#1\textstyle
      \textfont
    \else
      \ifx#1\scriptstyle
        \scriptfont
      \else
        \scriptscriptfont
      \fi
    \fi
  \fi
  \acta@overbarkern@family
```

```
}
```

`\acta@overbarrule` The `\acta@overbarrule` macro is similar. This will serve as the rule thickness. The name "overbarrule" comes from the corresponding LuaTeX primitive and OpenType Math table entry.

```
\def\acta@overbarrule#1{%
  \fontdimen\acta@overbarrule@fontdimen
  \ifx#1\displaystyle
    \textfont
  \else
    \ifx#1\textstyle
      \textfont
    \else
      \ifx#1\scriptstyle
        \scriptfont
      \else
        \scriptscriptfont
      \fi
    \fi
  \fi
  \acta@overbarrule@family
}
```

`\acta@overbarvgap` The `\acta@overbarvgap` macro is similar. When `\acta@overbarrule@fontdimen` and `\acta@overbarvgap@fontdimen` coincide, we use three times the rule thickness as the vertical gap; otherwise we use the different `\fontdimen` specified by the latter.

```
\def\acta@overbarvgap#1{%
  \ifx\acta@overbarrule@fontdimen\acta@overbarvgap@fontdimen
    \thr@@
  \fi
  \fontdimen\acta@overbarvgap@fontdimen
  \ifx#1\displaystyle
    \textfont
  \else
    \ifx#1\textstyle
      \textfont
    \else
      \ifx#1\scriptstyle
        \scriptfont
      \else
        \scriptscriptfont
      \fi
    \fi
```

```
  \fi
  \acta@overbarvgap@family
}
```

A.3 Actuarial angle

The code for `\actuarialangle`, `\acta@angle`, `\angl` and the underlying macros were given to the present author by a colleague many years ago. The original author is unknown. Some of the comments below are his or hers.

`\actuarialangle` We first define the "raw" user level command.

```
\DeclareRobustCommand{\actuarialangle}{\mathpalette\acta@angle}
```

The operation of `\mathpalette` ensures that proper sizing the command is ever used outside of a first-level subscript.

`\acta@angle` Next we define the real workhorse.

```
\def\acta@angle#1#2{%
  \mathord{%
```

Add a bit of preceding space.

```
    \mkern1mu%
```

We need many nested boxes here: first a vbox to stack the horizontal rule (with some extra space on top) of the angle and the symbol; second an hbox to position the symbol and the right descender of the angle side-to-side; third a vbox to insert spacing between the horizontal rule and the symbol.

```
    \vbox{%
      \kern\acta@overbarkern#1%
      \hrule \@height\acta@overbarrule#1%
      \hbox{%
        \vbox{%
```

The amount of vertical gap is the normal space for `\overline`.

```
          \kern\acta@overbarvgap#1%
          \hbox{$\m@th#1#2$}%
        }%
```

Make the right-hand rule extending down to the depth of a parenthesis even if the symbol under the angle does not have a descender.

```
        \setbox\z@\hbox{$#1\mathstrut$}%
        \vrule \@width\acta@overbarrule#1\@depth\dp\z@
      }%
    }%
```

Finishing touch is a bit of following space.

```
    \mkern1mu%
  }%
}
```

\angl \angln \anglr \anglk

Finally, we define the main user level function `\angl` and shortcuts for common cases.

```
\ifacta@thinspace
  \newcommand*\angl[1]{{\actuarialangle{#1\mkern1mu}}}
\else
  \newcommand*\angl[1]{{\actuarialangle{#1\mkern2mu}}}
\fi
\newcommand*\angln{{\angl n}}
\newcommand*\anglr{{\angl r}}
\newcommand*\anglk{{\angl k}}
```

A.4 Over angle bracket

The code of this section is executed only if `\acta@bracket` is `true`, that is when the package is *not* loaded with option `nobracket`.

```
\ifacta@bracket
```

Drawing the angle bracket requires package pict2e (Gäßlein et al., 2016) to get arbitrary slopes and neat line joins in paths.

```
  \RequirePackage{pict2e}
```

\overanglebracket

Here is the user level command.

```
  \DeclareRobustCommand{\overanglebracket}{%
    \mathpalette\acta@anglebracket}
```

We use `\mathpalette` as above.

\acta@anglebracket

The workhorse is `\acta@anglebracket`. It builds the bracket symbol with path lines.

```
  \def\acta@anglebracket#1#2{%
```

We construct the whole symbol in the same style as `\acta@angle`, i.e., a bit of preceding and following space, to prevent clashing into the trailing colon.

```
    \mathord{%
      \mkern1mu%
```

We setup the rule thickness here.

```
      \linethickness{\acta@overbarrule#1}%
```

Box zero contains the material under the bracket. The width of this box will determine the width of the flat part of the bracket and the height, the length of the descenders of the bracket. Hence we store these values.

```
    \setbox\z@\hbox{$\m@th#1#2$}%
    \dimen\z@\wd\z@ \dimen\tw@\ht\z@
```

Box two contains the bracket itself. It is drawn in three parts stitched together; the first and third parts are expressed in a dimension relative to `\dimen2` whereas the central part is expressed relative to `\dimen0`.

```
    \setbox\tw@\hbox{%
      \unitlength\dimen\tw@
      \begin{picture}(0.4,0)
        \polyline(0.4001,0)(0.4,0)(0,-0.8)
      \end{picture}%
      \unitlength\dimen\z@
      \begin{picture}(1,0)
        \put(0,0){\line(1,0){1}}
      \end{picture}%
      \unitlength\dimen\tw@
      \begin{picture}(0.4,0)
        \polyline(-0.0001,0)(0,0)(0.4,-0.8)
      \end{picture}%
    }%
```

We store the total width of the whole bracket to center the material under it, as follows.

```
    \dimen@\wd\tw@
```

The box containing the whole symbol. The lineskip between the bracket and the statuses is the same as in `\acta@angle`, plus half the rule thickness `\@halfwidth`.

```
    \vbox{%
      \baselineskip\z@
      \lineskip\acta@overbarvgap#1%
      \advance\lineskip\@halfwidth
      \lineskiplimit\lineskip
```

The extra space is the same as in `\acta@angle`, plus half the rule thickness `\@halfwidth`.

```
      \kern\acta@overbarkern#1%
      \kern\@halfwidth
      \box\tw@
      \hbox to\dimen@{\hss\unhbox\z@\hss}%
    }%
```

Finish with a bit of following space.

```
      \mkern1mu%
    }%
  }
```

`\group` Alias for `\overanglebracket`.

```
  \let\group\overanglebracket
\fi
```

A.5 Compatibility with unicode-math and OpenType math fonts

`\AtBeginDocument` The unicode-math package is to be loaded *after* other math setup packages, e.g., amsmath and mathtools. This usually implies that unicode-math is loaded after actuarialangle. So we delay the adjustments for unicode-math.

```
\AtBeginDocument{%
  \@ifpackageloaded{unicode-math}{%
```

When the unicode-math package is loaded, OpenType math font will be used and the compile engine must be either LuaTeX or XeTeX. We test against the XeTeX primitive `\XeTeXcharclass` to find out which engine is used.

```
    \ifx\XeTeXcharclass\@undefined
```

In this case the engine is LuaTeX. We redefine `\acta@overbarkern`, `\acta@overbarrule` and `\acta@overbarvgap` to be the LuaTeX primitives `\Umathoverbarkern`, `\Umathoverbarrule` and `\Umathoverbarvgap`, respectively. We also undefine the fontdimen and family parameters.

```
      \global\let\acta@overbarkern\Umathoverbarkern
      \global\let\acta@overbarrule\Umathoverbarrule
      \global\let\acta@overbarvgap\Umathoverbarvgap
      \global\let\acta@overbarkern@fontdimen\@undefined
      \global\let\acta@overbarkern@family\@undefined
      \global\let\acta@overbarrule@fontdimen\@undefined
      \global\let\acta@overbarrule@family\@undefined
      \global\let\acta@overbarvgap@fontdimen\@undefined
      \global\let\acta@overbarvgap@family\@undefined
    \else
```

Otherwise the engine is XeTeX. We use `\fontdimen54\<X>font2` for the rule thickness. We should use `\fontdimen53` and `\fontdimen55` for the vertical gap and the extra space, respectively, but XeTeX seems to have made a mistake here (see the discussion on StackExchange). So, for the vertical gap, we fallback to three times the rule thickness. For the extra space, we fallback to the rule thickness.

```
      \gdef\acta@overbarkern@fontdimen{54}% XeTeX mistake? Use 55?
      \gdef\acta@overbarkern@family{\tw@}%
      \gdef\acta@overbarrule@fontdimen{54}%
      \gdef\acta@overbarrule@family{\tw@}%
      \gdef\acta@overbarvgap@fontdimen{54}% XeTeX mistake? Use 53?
      \gdef\acta@overbarvgap@family{\tw@}%
    \fi
  }{}%
}
```

References

D. Beauchemin and V. Goulet. *Actuarial symbols of life contingencies and financial mathematics*, 2017. URL https://www.ctan.org/pkg/actuarialsymbol/.

N. L. Bowers, H. U. Gerber, J. C. Hickman, D. A. Jones, and C. J. Nesbitt. *Actuarial Mathematics*. Society of Actuaries, Shaumburg, IL, second edition, 1997. ISBN 0-9389594-6-8.

H. Gäßlein, R. Niepraschk, and J. Tkadlec. *The pict2e package*, 2016. URL https://www.ctan.org/pkg/pict2e/.

Version history