Grundgesetze.sty for \LaTeX{}2e Documentation

Marcus Rossberg
University of Connecticut
marcus.rossberg@uconn.edu

Version 1.02 2014/06/16

Grundgesetze.sty is a \LaTeX{}2e package for typesetting Gottlob Frege’s begriffsschrift [concept-script] formalism in the style of his Grundgesetze der Arithmetik (1893/1903). Grundgesetze.sty was developed for the 2013 English edition.\textsuperscript{1} The package is based on Josh Parsons’s begriff.sty which renders the formalism in the style of Frege’s earlier work, Begriffsschrift (1879). It was amended by Richard G. Heck Jr., J. J. Green, Agustín Rayo, and Marcus Rossberg. Thanks to Philip Ebert for testing and suggestions. Note that Frege’s defined function symbols are not rendered by this package, but by J. J. Green’s fge.sty.

1 Options

At present the only package option is bguq, which causes the package to use the bguq font for an alternative universal quantifier (concavity), and this option accepts a value (being the size to be used, as in bguq=6, the default being 5). Of course, one must have the bguq font installed to use this option, but it is included in recent versions of the big \TeX{} distributions.

2 Basic Commands

\begin{verbatim}
\GGhorizontal The horizontal, —
\GGnot The negation-stroke, ¬
\GGconditional Conditional-stroke: called as \GGconditional{p}{q} yields \( q \) (i.e., ‘\( p \supset q \)’)
\GGquant Concavity: called as \GGquant{\mathfrak{a}} gives \( a \) (i.e., universal quantifier, ‘\( a \)’ is the quantified variable)
\GGjudge Judgement-stroke, \( \vdash \)
\GGdef Definition-stroke, \( \triangleq \)
\GGbracket Automatically scaling brackets, \GGbracket{\ldots} yields (\ldots) (see examples)
\GGsqbracket Analogous square brackets, [\ldots]
\end{verbatim}

A complete list of commands and synonyms in the package can be found in Table 4, and the lengths parameterising the appearance of the output in Table 5.

2.1 Examples

• \texttt{\textbackslash_GGjudge \textbackslash_GGquant{\textbackslant\textfrak{a}} \textfrak{a} = \textfrak{a}} yields
  \[ \vdash \textbackslash_GGquant{\textbackslant\textfrak{a}} \textfrak{a} = \textfrak{a} \]

• \texttt{\textbackslash_GGjudge \textbackslash_GGnot \textbackslash_GGquant{\textbackslant\textfrak{F}} \textbackslash_GGnot \textbackslash_GGquant{\textbackslant\textfrak{a}} \textfrak{F}a} yields
  \[ \vdash \textbackslash_GGnot \textbackslash_GGquant{\textbackslant\textfrak{F}} \textbackslash_GGnot \textbackslash_GGquant{\textbackslant\textfrak{a}} \textfrak{F}a \]

• \texttt{\textbackslash_GGjudge \textbackslash_GGconditional{\textbackslash_GGhorizontal p}} \{p\} yields
  \[ \vdash p \quad (\neg p) \]

• \texttt{\textbackslash_GGjudge \textbackslash_GGbracket{\textbackslash_GGconditional{\textbackslash_GGnot q}} \{q\} = \textbackslash_GGbracket{\textbackslash_GGconditional{\textbackslash_GGnot q}} \{\textbackslash_GGnot p\}} yields
  \[ \vdash \textbackslash_GGbracket{\textbackslash_GGconditional{\textbackslash_GGnot q}} \{q\} = \textbackslash_GGbracket{\textbackslash_GGconditional{\textbackslash_GGnot q}} \{\textbackslash_GGnot p\} \]

There are further examples, including Frege’s basic laws of logic, available for download on www.frege.info.

3 Advanced Typesetting

3.1 Left-alignment of terminal formulae: \texttt{\textbackslash_GGterm}

Conditional-strokes, negation-strokes, and concavities that are embedded in conditionals can result in a ragged appearance of the formula:

• \texttt{\textbackslash_GGjudge \textbackslash_GGconditional{p}} \{\textbackslash_GGconditional{q}} \{p\}} yields:
  \[ \vdash p \quad q \quad p \]

• \texttt{\textbackslash_GGjudge \textbackslash_GGconditional{Fa} \{\textbackslash_GGnot \textbackslash_GGquant{\textbackslant\textfrak{a}} \textbackslash_GGnot F \textfrak{a}} \textfrak{F}a} yields:
  \[ \vdash \textbackslash_GGconditional{Fa} \quad \textbackslash_GGnot \textbackslash_GGquant{\textbackslant\textfrak{a}} \textfrak{F}a \]

In Frege’s original work, the component formulae of conditionals are left-aligned. This can be achieved by marking “terminal formulae” using the command \texttt{\textbackslash_GGterm{\textbackslash_GGmath}}; the length \texttt{\textbackslash_GGlinewidth} specifies the distance of the terminal formula from the left end of the whole formula (typically, ‘\textbackslash ‘)
• \texttt{\setlength{\GGlinewidth}{9.2pt} \GGjudge} \\
\texttt{\GGconditional} \\
{\texttt{\{\GGterm{p}\}}} \\
{\texttt{\{\GGconditional{\GGterm{q}}\}}} \\
{\texttt{\{\GGterm{p}\}}} \\
yields: \\
\begin{array}{c}
\setlength{\GGlinewidth}{9.2pt} \\
\texttt{p} \\
\texttt{\GGconditional} \\
\{\GGterm{p}\} \\
\texttt{q} \\
\texttt{p}
\end{array}

• \texttt{\setlength{\GGlinewidth}{25.2pt}} \\
\texttt{\GGjudge} \texttt{\GGconditional{\GGterm{Fa}}} \\
\texttt{\{\GGnot} \texttt{\GGquant{\mathfrak{a}}} \texttt{\GGnot} \\
\texttt{\GGterm{F \mathfrak{a}}} \\
\texttt{\GGnot} \\
\texttt{\GGterm{F \mathfrak{a}}} \\
yields: \\
\begin{array}{c}
\setlength{\GGlinewidth}{25.2pt} \\
\texttt{\GGconditional{\GGterm{Fa}}} \\
\texttt{Fa}
\end{array}

\begin{tabular}{ll}
\text{negation-stroke} & \tau \\
\text{conditional-stroke} & \tau \\
\text{concavity} & \sim \\
\text{judgement-stroke:} & \ \ \\
\text{present} & \text{add .4pt} \\
\text{not present} & \text{subtract 2pt}
\end{tabular}

Table 1: Lengths of embedded symbols

The correct values for \texttt{\GGlinewidth} for each formula can be determined by adding up the lengths of the embedded symbols, as given in Table 1, or by using a GUI that allows producing \LaTeX and XML code for \textit{begriffsschrift} formulae via mouse-click and that will calculate and output the correct values. The GUI is available for download on \url{www.frege.info}.

3.2 Adding horizontal lengths manually: \texttt{\GGnot}, etc.

Readability is sometimes aided by moving, e.g., negations to the right end of the horizontal in a complex formula. For instance, Frege nearly always preferred the rendering displayed on the right in these types of formulae:

\begin{itemize}
\item \texttt{(a)} \begin{array}{c}
\setlength{\GGlinewidth}{9.2pt} \\
\texttt{\GGconditional{\GGterm{f(a)}}} \\
\texttt{\GGterm{f(a)}} \\
\texttt{\GGterm{f(a)}}
\end{array}
\item \texttt{(b)} \begin{array}{c}
\setlength{\GGlinewidth}{25.2pt} \\
\texttt{\GGconditional{\GGterm{g(a)}}} \\
\texttt{\GGterm{g(a)}} \\
\texttt{\GGterm{g(a)}} \\
\texttt{\GGterm{f(a)}} \\
\texttt{\GGterm{f(a)}} \\
\texttt{\GGterm{f(a)}} \\
\texttt{\GGterm{f(a)}}
\end{array}
\item \texttt{(c)} \begin{array}{c}
\setlength{\GGlinewidth}{9.2pt} \\
\texttt{\GGconditional{\GGterm{a}} \\
\texttt{\GGterm{b}}} \\
\texttt{\GGterm{f(b)}} \\
\texttt{\GGterm{f(a)}} \\
\texttt{\GGterm{a}} \\
\texttt{\GGterm{b}} \\
\texttt{\GGterm{f(b)}} \\
\texttt{\GGterm{f(a)}}
\end{array}
\end{itemize}
The right-hand formulae are produced by inserting commands for horizontals of the appropriate length directly at the position where the “space” should appear. The three right-hand formulae above are created in this way:

(a) \texttt{\textbackslash GGjudge \textbackslash Gconditional}
\begin{align*}
\texttt{\{\textbackslash GQuant\{\textbackslash mathfrak{a}\} \textbackslash Gnot f\{\textbackslash mathfrak{a}\}\}} \\
\texttt{\{\textbackslash GNoquant \textbackslash Gnot f(a)\}}
\end{align*}

(b) \texttt{\textbackslash GGjudge \textbackslash GConditional}
\begin{align*}
\texttt{\{\textbackslash GQuant\{\textbackslash mathfrak{a}\} \textbackslash GConditional\{f\{\textbackslash mathfrak{a}\}\}\{g\{\textbackslash mathfrak{a}\}\}\}} \\
\texttt{\{\textbackslash GNoquant \textbackslash GConditional\{f(a)\}\{g(a)\}\}}
\end{align*}

(c) \texttt{\textbackslash GGjudge \textbackslash GConditional}
\begin{align*}
\texttt{\{\textbackslash GNoquant \textbackslash Gnot f(a)\}} \\
\texttt{\{\textbackslash GConditional\{\textbackslash GNoquant \textbackslash Gnot f(b)\}\{\textbackslash GNoquant a=b\}\}}
\end{align*}

4 Comparison and compatibility with \texttt{begriff.sty}

Josh Parsons’s \texttt{begriff.sty}, on which \texttt{grundgesetze.sty} is based, is closer in appearance to Frege’s formalism as it is presented in Frege’s first book, \textit{Begriffsschrift} (1879). The corresponding commands were given different names so that both packages can be used in the same TeX document; see Table 2.

<table>
<thead>
<tr>
<th>\textit{begriff.sty} command</th>
<th>symbol</th>
<th>\textit{grundgesetze.sty} command</th>
</tr>
</thead>
<tbody>
<tr>
<td>\texttt{\textbackslash BGcontent}</td>
<td>\texttt{\textbackslash GHorizontal}</td>
<td></td>
</tr>
<tr>
<td>\texttt{\textbackslash BGnot}</td>
<td>\texttt{\textbackslash Gnot}</td>
<td></td>
</tr>
<tr>
<td>\texttt{\textbackslash BGconditional{p}{q}}</td>
<td>\texttt{\textbackslash Gnot}</td>
<td></td>
</tr>
<tr>
<td>\texttt{\textbackslash GConditional{f}{g}}</td>
<td></td>
<td></td>
</tr>
<tr>
<td>\texttt{\textbackslash BGquant{\textbackslash mathfrak{a}}}</td>
<td>\texttt{\textbackslash Gquant{\textbackslash mathfrak{a}}}</td>
<td></td>
</tr>
<tr>
<td>\texttt{\textbackslash BGassert}</td>
<td>\texttt{\textbackslash GJudge}</td>
<td></td>
</tr>
<tr>
<td>\texttt{\textbackslash BGbracket{\ldots}}</td>
<td>\texttt{\textbackslash Gbracket{\ldots}}</td>
<td></td>
</tr>
</tbody>
</table>

Table 2: Compatibility with \texttt{begriff.sty}

Also note the differences in alignment between \texttt{\textbackslash BGbracket} and \texttt{\textbackslash Gbracket} as shown in Table 3

\begin{align*}
\texttt{\textbackslash BGbracket} & \quad \{ \hat{\textit{f}}(\textit{z}) = \hat{\alpha}(\textit{g}(\textit{a})) \} = \texttt{\textbackslash Gbracket} \{ \hat{\textit{f}}(\textit{z}) = \hat{\alpha}(\textit{g}(\textit{a})) \} \\
\texttt{\textbackslash Gbracket:} & \quad \{ \hat{\textit{f}}(\textit{z}) = \hat{\alpha}(\textit{g}(\textit{a})) \} = \texttt{\textbackslash Gbracket} \{ \hat{\textit{f}}(\textit{z}) = \hat{\alpha}(\textit{g}(\textit{a})) \}
\end{align*}

Table 3: \texttt{\textbackslash BGbracket} and \texttt{\textbackslash Gbracket} alignment
4.1 Conversion of a \texttt{begriff.sty} document into a \texttt{grundgesetze.sty} document

A straightforward way to convert the a \LaTeX document that uses \texttt{begriff.sty} into one that uses \texttt{grundgesetze.sty} without manually exchanging the commands is to find and replace (using wrap search) “\texttt{BG}” by “\texttt{GG}”. Synonyms have been added to \texttt{grundgesetze.sty} to allow the use of all \texttt{begriff.sty} commands “translated” in this way (see Table 4).

<table>
<thead>
<tr>
<th>command</th>
<th>symbol</th>
<th>synonym / comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>\texttt{\GGterm{\ldots}}</td>
<td>-</td>
<td>(marks terminal formula)</td>
</tr>
<tr>
<td>\texttt{\GGhorizontal}</td>
<td>-</td>
<td>\texttt{\GGcontent}</td>
</tr>
<tr>
<td>\texttt{\GGjudge}</td>
<td>\GGassert</td>
<td>\texttt{\GGjudgealone, \GGassertlong, \GGassertalone}</td>
</tr>
<tr>
<td>\texttt{\GGjudgelong}</td>
<td>\GGassert</td>
<td>\texttt{\GGassertvar{\langle length \rangle}} (variable horizontal length, here: 6pt)</td>
</tr>
<tr>
<td>\texttt{\GGjudgevar{\langle length \rangle}}</td>
<td>\GGassert</td>
<td>\texttt{\GGassertvar{\langle length \rangle}} (variable horizontal length, here: 6pt)</td>
</tr>
<tr>
<td>\texttt{\GGdef}</td>
<td>\GGneg</td>
<td>\texttt{\GGnotalone}</td>
</tr>
<tr>
<td>\texttt{\GGdeflong}</td>
<td>\GGneg</td>
<td>\texttt{\GGnotalone}</td>
</tr>
<tr>
<td>\texttt{\GGdefvar{\langle length \rangle}}</td>
<td>\GGneg</td>
<td>\texttt{\GGnotalone}</td>
</tr>
<tr>
<td>\texttt{\GGnot}</td>
<td>\GGneg</td>
<td>\texttt{\GGnotalone}</td>
</tr>
<tr>
<td>\texttt{\GGdnot}</td>
<td>\GGneg</td>
<td>\texttt{\GGnotalone}</td>
</tr>
<tr>
<td>\texttt{\GGconditional{p}{q}}</td>
<td>\GGneg</td>
<td>\texttt{\GGnotalone}</td>
</tr>
<tr>
<td>\texttt{\GGquant{\mathfrak a}}</td>
<td>\GGneg</td>
<td>\texttt{\GGnotalone}</td>
</tr>
<tr>
<td>\texttt{\GGall{a}}</td>
<td>\GGneg</td>
<td>\texttt{\GGnotalone}</td>
</tr>
<tr>
<td>\texttt{\GGbracket{\ldots}}</td>
<td>\GGneg</td>
<td>\texttt{\GGnotalone}</td>
</tr>
<tr>
<td>\texttt{\GGsqbracket{\ldots}}</td>
<td>\GGneg</td>
<td>\texttt{\GGnotalone}</td>
</tr>
<tr>
<td>\texttt{\GGnot}</td>
<td>\GGneg</td>
<td>\texttt{\GGnotalone}</td>
</tr>
<tr>
<td>\texttt{\GGnodnot}</td>
<td>\GGneg</td>
<td>\texttt{\GGnotalone}</td>
</tr>
<tr>
<td>\texttt{\GGoddspace}</td>
<td>\GGneg</td>
<td>\texttt{\GGnotalone}</td>
</tr>
<tr>
<td>\texttt{\GGtinyspace}</td>
<td>\GGneg</td>
<td>\texttt{\GGnotalone}</td>
</tr>
<tr>
<td>\texttt{\GGtiniestspace}</td>
<td>\GGneg</td>
<td>\texttt{\GGnotalone}</td>
</tr>
</tbody>
</table>

Table 4: All commands (and synonyms) defined by the package
<table>
<thead>
<tr>
<th>length</th>
<th>default value</th>
<th>explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>\texttt{\textbackslash{GGthickness}}</td>
<td>0.4pt</td>
<td>thickness of horizontal and vertical lines</td>
</tr>
<tr>
<td>\texttt{\textbackslash{GGquantthickness}}</td>
<td>0.75\times \texttt{\textbackslash{GGthickness}}</td>
<td>thickness of the line of the quantifier “dish”. Note that this value is unused if the \texttt{bguq} option has been selected</td>
</tr>
<tr>
<td>\texttt{\textbackslash{beforelen}}</td>
<td>2.4pt</td>
<td>length of horizontal before quantifier, conditional, and negation</td>
</tr>
<tr>
<td>\texttt{\textbackslash{GAfterlen}}</td>
<td>2pt</td>
<td>length of horizontal after quantifier, conditional, negation, judgement-, and definition-stroke</td>
</tr>
<tr>
<td>\texttt{\textbackslash{GSpace}}</td>
<td>3pt</td>
<td>space between right end of horizontal and terminal formula</td>
</tr>
<tr>
<td>\texttt{\textbackslash{Glift}}</td>
<td>2pt</td>
<td>lift of horizontal from baseline</td>
</tr>
<tr>
<td>\texttt{\textbackslash{GLinewidth}}</td>
<td>(n/a)</td>
<td>total length from left end of formula (typically, \texttt{\textbackslash{judge}}) and the beginning of the terminal formula (see §3.1)</td>
</tr>
</tbody>
</table>

Table 5: Length parameters and their default values