

Users Guide

GFnotation.tex (version 2.9, 29.03.2015)

1. Description

This package implements macros to typeset formulas and inferences in Gottlob Frege’s notation in the style of his “Begriffsschrift, eine der arithmetischen nachgebildete Formelsprache des reinen Denkens” (see [1]) or in the style of his main work “Grundgesetze der Arithmetik” (see [2]). Here is an example of a single formula: $\vdash \frac{\vdash A}{\vdash^a f(a)}$. And this is a complete inference:

$$\frac{\frac{\frac{\vdash A}{\vdash B}}{\vdash B}}{\vdash A}$$

The package has two input formats, the so-called “symbolic representation” and the “short form.” The short form outputs the formulas either in the style of Frege’s *Begriffsschrift* or in the style that is used in his *Grundgesetze*. The above formulas are in the style of the *Begriffsschrift*; here is the first one in the style of the *Grundgesetze*: $\vdash \frac{\vdash A}{\vdash^a f(a)}$.

Note: This users guide shows how to work with the package. It does not explain the constructions and symbols in Frege’s notation.

2. Elements of Frege’s notation

a) **Letters.** Frege used lowercase roman letters, a, b, c, \dots , uppercase Greek letters A, B, Γ, \dots , lowercase Greek letters $\alpha, \beta, \gamma, \dots$, and Fraktur letters $\mathfrak{a}, \mathfrak{e}, \mathfrak{f}, \dots$.

b) **Formulas (Begriffsschrift).** The following notation is used by Frege to build formulas:

Content stroke: $\text{—}A.$	Judgment: $\vdash A.$
Negation: $\neg A.$	Affirmation: $\text{—}A.$
Condition: $\frac{\text{—}B.}{\text{—}A}$	Generality: $\text{—}^a \Phi(a).$
Condition stroke: $\text{—} .$	Definition: $\text{—} ((A) \equiv B).$
Substitution: $a f(a).$	

c) **Inferences (Begriffsschrift).** For inferences two types of lines are used:

Single line: — . Double lines: = .

Separator between inference chains:

d) **Terminal symbols (Begriffsschrift).** Four special constructions appear in the *Begriffsschrift*:

$$\delta \left(\begin{array}{c} x(\alpha) \\ | \\ f(\delta, \alpha) \end{array} \right), \quad \frac{\gamma}{\beta} f(x_\gamma, y_\beta), \quad \frac{\gamma}{\beta} f(x_\gamma, z_\beta), \quad \frac{\delta}{\epsilon} f(\delta, \epsilon).$$

e) **Formulas (Grundgesetze).** The following notation is used by Frege to build formulas:

Content stroke: $\text{—}A.$	Judgment: $\vdash A.$
Negation: $\neg A.$	Affirmation: $\text{—}A.$
Condition: $\frac{\text{—}B.}{\text{—}A}$	Generality: $\text{—}^a \Phi(a).$
Condition stroke: $\text{—} .$	Definition: $\text{—} A = B.$

4. Shortcuts for letters

The characters that Frege used (except the roman letters) are coded with two letter shortcuts. The first letter indicates how the second letter is output. Uppercase Greek letters are introduced by ‘g’, lowercase Greek letters by ‘k’, and the Fraktur letters by ‘d’: `\gA$` outputs A , `\ka$` outputs α , and `\da$` outputs \mathfrak{a} .

Here is the list of the uppercase Greek letters; the coding follows [6].

<code>\gA\$</code> : A	<code>\gB\$</code> : B	<code>\gG\$</code> : Γ	<code>\gD\$</code> : Δ	<code>\gE\$</code> : E
<code>\gZ\$</code> : Z	<code>\gH\$</code> : H	<code>\gJ\$</code> : Θ	<code>\gI\$</code> : I	<code>\gK\$</code> : K
<code>\gL\$</code> : Λ	<code>\gM\$</code> : M	<code>\gN\$</code> : N	<code>\gX\$</code> : Ξ	<code>\gO\$</code> : O
<code>\gP\$</code> : Π	<code>\gR\$</code> : P	<code>\gS\$</code> : Σ	<code>\gT\$</code> : T	<code>\gU\$</code> : U
<code>\gF\$</code> : Φ	<code>\gQ\$</code> : X	<code>\gY\$</code> : Ψ	<code>\gW\$</code> : Ω	

For the Fraktur letters only a subset of the alphabets is provided; these are the letters that appear in the *Begriffsschrift* or in part 1 of Vol. 1 of the *Grundgesetze*:

<code>\da\$</code> : \mathfrak{a}	<code>\dc\$</code> : \mathfrak{c}	<code>\dd\$</code> : \mathfrak{d}	<code>\de\$</code> : \mathfrak{e}	<code>\df\$</code> : \mathfrak{f}
<code>\dg\$</code> : \mathfrak{g}	<code>\dm\$</code> : \mathfrak{m}	<code>\dn\$</code> : \mathfrak{n}	<code>\do\$</code> : \mathfrak{o}	<code>\ddp\$</code> : \mathfrak{p}
<code>\dq\$</code> : \mathfrak{q}	<code>\dr\$</code> : \mathfrak{r}	<code>\dt\$</code> : \mathfrak{t}	<code>\du\$</code> : \mathfrak{u}	<code>\dF\$</code> : \mathfrak{f}

Note: Commands like `\dp` or `\do` shouldn’t change their plain TeX meaning. Therefore the commands for the letters ‘o’ and ‘p’ are build with two ‘d’s.

You can add more letters using a simple `\def`. For example, to add \mathfrak{s} declare `\def\ds{\frak s}`.

5. Input of formulas using the “symbolic representation” (Begriffsschrift)

This input format gives complete control over the output of the formulas. But it requires some amount of typing. All formulas are built from symbols and three of them are grouped into the argument of a control symbol. The control symbol is the `*`. (The original meaning as discretionary multiplication is kept in the new control word `\discretionarytimes`.)

In total there are 35 symbols. In the following list the parentheses contain the sequence `*"#` if symbol ‘#’ is explained. The first collection lists the important symbols that are used for hand-coding the formulas in Frege’s notation.

- _ () is the empty space;
- (—) is a horizontal line for the content stroke and the inference rule;
- ~ (⊖) is a horizontal line with centered negation;
- + (⊕) is a horizontal line with two centered negation indicators (affirmation);
- : (⊢) is the then-part of the condition;
- ’ (⊣) is the if-part of the condition;
- ! (⊥) is a vertical line for the condition stroke;
- [(⊤) is a vertical line placed left and starts a horizontal line;
- | (⊥) is the vertical line for the substitution;
- 3 (⊥⊥) are two vertical lines used for *definitions*;
- = (⊥⊥) are two horizontal lines.

The following symbols extend the basic list and can be used to fine-tune the output.

- " () is a skip, that is, no output at all;
- (⊖) negation but the indication of negation is moved to the left;
-) (⊖) negation with the indicator moved to the right;
- < (⊖) negation: the indicator is placed at the left end;
- > (⊖) negation: the indicator is placed at the right end;
- ~ (⊕) affirmation with indicators moved to the left;

/ (\neg) affirmation with indicators moved to the right;

] (\lvert) is a vertical line placed right, that is, only the judgment stroke is printed.

For the generality a letter must be given. Therefore the parentheses contain the sequence `*#a.` (`*@a` for `@`). Instead of the letter ‘a’ any other letter can be used that is available in the Fraktur font (see Section 4). A triple for the generality must be started with one of the following symbols `.`, `'`, `@`. If it is not the `@` then the symbol after the letter must be one of `.`, `'`, `;`.

`.` ($\overset{a}{-}$) is a short vertical line;

`,` ($\overset{a}{\neg}$) is a short vertical line with a centered negation indicator;

`'` ($\overset{a}{\neg}$) represents negation but the indicator is moved to the left;

`;` ($\overset{a}{\neg}$) is a short vertical line with affirmation;

`@` ($\overset{a}{\neg}$) signals (without any output) that a letter for the generality follows.

Note: The letter ‘ β ’ is special. It uses the content of the token register `\dsz`. For example, with the default `\dsz={\xi}` `*.β` outputs $\overset{\xi}{-}$. Of course, β must be a letter so for plain T_EX the command `\catcode'\beta=11` must be entered before `\input GFnotation` is done.

Example 1. The formula $\lvert \overset{a}{-} A$ of Section 1 can be coded in the

following way:

```
\vtop{%
  \hbox{\strut\*[-:\*--- $\gA$}
  \hbox{\strut\*_'\*a. $f(\da)$}}
```

The last list contains symbols that are used by the short form. The symbols do not try to have a mnemonic representation for their output. Most of these symbols can “change” and stand for one of two symbols from the above lists.

0 negation either moved to the left or the right: `0=(`)

1 affirmation: `1=^/`

2 again affirmation: `2=^+`

4 not required content strokes: `4=-"`

5 not required empty space: `5=_"`

6 judgment strokes: `6=[]`

7 negated generality: `7=' ,`

8 again negation: `8=<~`

9 represents a content stroke that has only one third of its usual length;

`*` is a skip whose length is multiplied by 5/3 to get back into alignment after a 9;

`$` is used only with a generality and represents either a short content stroke or an empty space.

6. Input for inference chains in the “symbolic representation” (Begriffsschrift)

In order to typeset complete inferences several formulas must be positioned in a chain and the rule for the inference has to be entered. The construction of Example 1 with `\vtop`, `\hbox`, and `\strut` is shortened to a single command `\nlp`. The terminal symbols become an argument of a second command `\ce`.

1. `\nlp<n>` starts a new line and sets the following text `<n>` units to the right;

2. `\ce{<s>}` outputs the string `<s>` with some spacing at the end of a line.

Example 2. This is the implementation of the inference in Section 1, that is

$$\frac{\begin{array}{l} \vdash A \\ \vdash B \end{array}}{\vdash A} :$$

```
\nlp4\*_[-\*-:-\ce{\$gA\$}
\nlp4\*_--\*_'\-ce{\$gB\$}
\nlp4\*_[-\*---\ce{\$gB\$}
\nlp4\*---\*---\*---
\nlp4\*_[-\*---\ce{\$gA\$} .
```

For complete inferences with substitutions more commands are required:

3. `\ci{<s>}` is like `\ce` but must be used in the line and spaces that follow are ignored;
4. `\nlpc<n>{<s>}` acts like `\nlp<n>` and sets the string `<s>` to the left in the white space;
5. `\fono{<m>}` outputs `<m>` as the number of the formula at the right end of the previous line;
6. `\rep<n>{<f>}` repeats the `*-formula <f>` `<n>` times;
7. `\bcc<n>/<p><m>:<c><f>\ecc` typesets an inference rule. `<n>` is the position to the right, `<m>` is a formula number used in the inference, and `<f>` is the rule itself. The parameter `<p>` is either 'l', 'm', or 'r' to position the formula number to the left, in the middle, or to the right. And `<c>` is either '.' or ':' to output one or two colons after the formula number.

Example 3. The following code

```
\nlp4\ci{\hss\thinspace58\hss}\rep3\*_--\*_'\-ce{\$f(b)\$}
\nlpc4{\$f(A)\$}\*_|\_--\ci{\$f(A)\$}\rep2\*_--\*_!\_--\*_'\-ce{\$g(b)\$}
\nlp4\*_|\_--\ci{\$g(A)\$}\rep2\*_--\*_'\-ce{\$f(\da)\$}
\nlpc4{\$c\$}\*_|\_--\ci{\$b\$} \rep3\*_--\*_'\-ce{\$g(\da)\$}
\bcc4/m30:.\rep3\*_--\null \rep6\*_--\ecc
\nlpc4{\$a\$}\*_|\_--\ci{\$f(b)\$} \rep2\*_--\*_'\-ce{\$f(\da)\$}
\nlpc4{\$c\$}\*_|\_--\ci{\$g(b)\$} \rep2\*_--\*_!\_--\*_'\-ce{\$g(\da)\$}
\nlpc4{\$b\$}\*_|\_--\ci{\$f(\da)\$} \*_--\*_!\_--\*_'\-ce{\$f(b)\$}
\nlp4\*_|\_--\*_'\-ce{\$g(\da)\$} \*_--\*_'\-ce{\$g(b)\$}
\fono{59}
```

outputs formula 59 of the *Begriffsschrift*:

$$\begin{array}{l} f(A) \quad | \quad \begin{array}{l} \vdash f(A) \\ \vdash g(A) \end{array} \\ c \quad | \quad b \\ (30) : \\ a \quad | \quad \begin{array}{l} f(b) \\ g(b) \end{array} \\ c \quad | \quad \begin{array}{l} f(a) \\ g(a) \end{array} \\ b \quad | \quad \begin{array}{l} f(a) \\ g(a) \end{array} \end{array} \quad \frac{\begin{array}{l} \vdash f(b) \\ \vdash g(b) \end{array}}{\vdash f(a)} \quad \frac{\begin{array}{l} \vdash f(a) \\ \vdash g(a) \end{array}}{\vdash f(b)} \quad \frac{\begin{array}{l} \vdash f(b) \\ \vdash g(b) \end{array}}{\vdash g(b)} \quad (59.)$$

One more command is required for collections of inferences:

8. `\decollator<l>` outputs a thick line of width `0.<l>\hsize`. The line is centered. It is preceded and followed by vertical space.

Example 4. Here are some inferences of the *Begriffsschrift*:

$$\begin{array}{c}
 1 \qquad \qquad \qquad \begin{array}{c} \text{---} \\ | \\ \text{---} \end{array} \begin{array}{c} a \\ b \\ a \end{array} \\
 \\
 (8) : \qquad \qquad \qquad \text{---} \\
 d \mid a \qquad \qquad \qquad \begin{array}{c} \text{---} \\ | \\ \text{---} \end{array} \begin{array}{c} a \\ a \\ b \end{array}
 \end{array} \tag{26}$$

$$\begin{array}{c}
 26 \qquad \qquad \qquad \text{---} \\
 b \mid \begin{array}{c} \text{---} \\ | \\ \text{---} \end{array} \begin{array}{c} a \\ b \\ a \end{array} \qquad \qquad \qquad \begin{array}{c} \text{---} \\ | \\ \text{---} \end{array} \begin{array}{c} a \\ a \\ a \\ b \\ a \end{array} \\
 \\
 (1) :: \qquad \qquad \qquad \text{---} \\
 \qquad \qquad \qquad \begin{array}{c} \text{---} \\ | \\ \text{---} \end{array} \begin{array}{c} a \\ a \end{array}
 \end{array} \tag{27}$$

The command `\decollator4` appears after formula 26. █

7. Input of formulas using the “short form”

This input format reduces the amount of typing compared to the symbolic representation. It follows given rules and produces as an intermediate step the code of the symbolic representation.

The input is recursively entered using four constructions:

1. `=<sign>{<formula>}` to add a signed content stroke in front of a formula;
2. `<sign><sign>{<formula><sign>{<formula>}}` to build a signed condition from two signed formulas;
3. `*<sign><character><sign>{<formula>}` to add a signed concavity in front of a signed formula;
4. `!\<nameofdef><arguments>` to insert a previously defined subformula.

Here `<formula>` is either a valid formula constructed with the four rules or a terminal string (both are typeset in math mode) and `<sign>` is one of the following symbols:

- . represents no sign;
- represents the negation;
- + represents the affirmation.

In rule 4 `<nameofdef>` is together with the preceding backslash a defined control word that has at most three arguments given in `<arguments>`. The control word must define a part of a formula. This rule implements *macro expansion* to avoid retyping of often used parts in formulas.

Most formulas are created by rules 2 and 3; rule 1 is only needed to type a single line.

Important note: The braces around `<formula>` can be omitted if the formula is a terminal string that consists of a single token. Double curly braces are required around terminal strings that have more than five tokens. But single braces must always be used for formulas in rules 1–3, which are not terminal strings. (In Example 5 below case i shows that a single token needs no braces. The cases ii and iii show the use of single and double curly braces.)

Formulas that are created by the above rules must be given to the following commands:

1. `\frege<j>{<f>}` where `<f>` is a formula created by the above stated rules and `<j>` is either ‘.’ or ‘|’.
2. `\formula p<n><j>{<f>}` adds a position parameter `<p>` to move the formula `<n>` units to the right.

The formula `<f>` is output as a judgment if `<j>` equals |. The command `\frege` shall be used in running text, `\formula` in vertical mode. But `\frege` doesn’t start a paragraph; use `\indent` (or `\noindent`) if `\frege` is the first item in a paragraph.

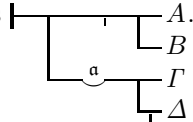
Example 5. Here are a few examples for all four rules:

- i) `\frege|{-\gA}` outputs $\vdash A$.
- ii) `\frege|{-.\gA-\{g(\ka,\kb)\}}` outputs $\vdash g(\alpha, \beta)$.
- iii) `\frege|{-a+\{f(\da)\}}` outputs $\vdash^a f(a)$.
- iv) `\def\0#1{*.\a.\#1(\da)}` together with `\frege|{-.\gA-\{!\0f\}}` outputs $\vdash^a f(a)$.

The output of iv) is nearly the formula of Section 1 ($\vdash^a A$); it can be coded (without `\0`) as `\frege|{-.\gA.*.\a.\{f(\da)\}}`. ■

Of course, a formula in Frege's notation that has several lines is not easy to read in a sequential form. The periods and braces do not help to make the structure of that formula easy to recognize. I suggest to copy a Frege formula from left to right and from bottom to top and to use more than one line for the sequential coding.

Example 6. This example shows how to construct in a structured manner the short form of a given formula, which is written in Frege's notation. The formula is



Step 1: The formula is a judgment and starts at the left with a condition, whose upper line contains a negation indicator. So the coding starts with

```
\frege|{-.\{.\}}
```

Or write it in two lines to make the structure visible:

```
\frege|{-.\{
.\}}
```

Step 2: The bottom part starts with a concavity, so the input becomes

```
\frege|{-.\{
.\*.\a.\}}
```

Step 3: A simple condition comes after the concavity

```
\frege|{-.\{
.\*.\a.\.\gG-\gD\}}
```

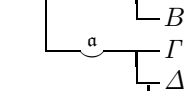
And the final step is to add the condition in the upper part:

```
\frege|{-.\{.\gA.\gB\
.\*.\a.\.\gG-\gD\}}
```

Note: The input

```
\frege|{-.\{.\gA.\gB\
.\*.\a.\.\gG-\gD\}}
```

outputs $\vdash^a f(a)$, which looks very similar to the given formula but, of



course, the negation indicator in the upper part has moved a little bit to the right. ■

8. Inferences of the Begriffsschrift in the “short form”

Five commands are available to typeset inferences with the short form:

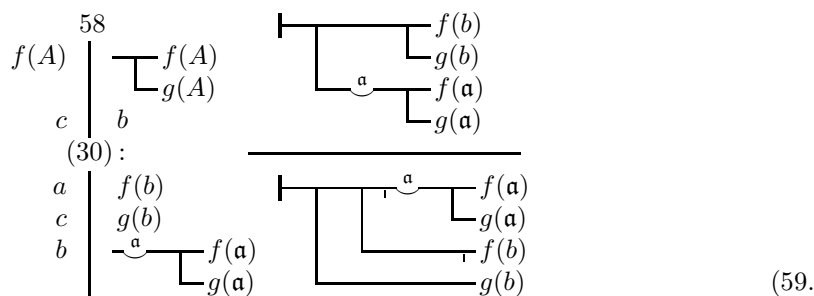
1. `\outof p<n>,<o>"<m>"with<r>\thatis` starts the inference chain. The parameter `<n>` is the number of units the output is moved to the right, `<m>` is a number of a formula, `<o>` is an offset to position the number `<m>` `<o>` lines lower, and `<r>` is either empty or a space separated list of pairs `v:F` to indicate the substitution of `v` by the formula `F`.
2. `\use p<n>"<m>"` is a shortcut of `\outof` if the parameters `<o>` and `<r>` are not needed; `<o>` is set 0 and `<r>` is empty.
3. `\followswith p<n>"<m>"a<c>p<N>s<l>` draws the inference rule and can be expressed in the `\bcc` macro of Section 6, no 7, as `\bcc{<n>}/m{<m>}:<c>\rep{<N>-<n>}*_--\rep{<l>}*---\ecc`.
4. `\substituting p<n> <r>\whichgives` has the usual position parameter `<n>` and a list of substitutions `<r>` in the above described form `v:F`.
5. `\named "<m>"` outputs the number `<m>` as the number of the previous formula.

An inference chain is typically typed with the sequence:

```
\outof ... with ... \thatis
\formula ...
\followswith ...
\substituting ... \whichgives
\formula ...
\named ...
```

or if no substitution in the first formula is required, the `\outof` line is replaced by the simpler `\use`.

Example 7. The formula of Example 3



has this coding:

```
\outof p4,0"58"with f(A):{..{f(A)}.{g(A)}}
c:b
\thatis
\formula p8|{..{f(b)}.{g(b)}}
.*.a.{f(\da)}.{g(\da)}}}
\followswith p4"30"a.p8s6
\substituting p4 a:{f(b)}
c:{g(b)}
b:{*a.{f(\da)}.{g(\da)}}}
\whichgives
\formula p8|{..{*a.{f(\da)}.{g(\da)}}}
-f(b)}}
.g(b)}}
\named "59"
```


12. Inferences of the Grundgesetze

The list entry 1c of Section 10 needs some explanation as it extends the list of available symbols for inferences. Two of the symbols have no output (? and >). In the following list the text in parentheses contains the sequence *?#" or *?># for the symbol '#'.
 ? (must be the first in the triple) signals that the next two symbols build an inference line in the style of the *Grundgesetze*. Such symbols use 50% more space than ordinary symbols.

- (—) is a single horizontal stroke;
- . (·) is a single (centered) period;
- = (≡) is a double stroke;
- * (—) is a single stroke in the height of the upper line of the double stroke;
- _ () is an empty space;
- " () is a skip;
- > has no output; the next symbol must be a *transition-sign*:
 - x (×) it must follow a >;
 - u (⏟) it must follow a >.

Inferences of the *Grundgesetze* are separated by a new type of line:

1. \separator outputs a thick line with a centered dot.

Example 10. Here is an inference chain from the *Grundgesetze*:

$$\begin{array}{c}
 \text{III} \quad \begin{array}{c} \vdash \text{---} \overset{f}{\text{---}} \text{---} f(a) \\ \vdash \text{---} \text{---} f(a) \\ \vdash \text{---} \text{---} a = a \end{array} \\
 \quad \quad \quad \times \\
 \quad \quad \quad \begin{array}{c} \vdash \text{---} \text{---} a = a \\ \vdash \text{---} \overset{f}{\text{---}} \text{---} f(a) \\ \vdash \text{---} \text{---} f(a) \end{array} \quad (\alpha) \\
 \hline
 \text{I} \quad \begin{array}{c} \vdash f(a) \\ \vdash f(a) \\ \underbrace{\quad} \\ \vdash \overset{f}{\text{---}} \text{---} f(a) \\ \vdash \text{---} \text{---} f(a) \end{array} \quad (\beta) \\
 (\alpha): \quad \hline
 \vdash a = a \quad \quad \quad \text{(IIIe)}
 \end{array}$$

And this is the code:

```

\use p3"III"
\formula p5|{.-{*f.f.{{\df(a)}.{\df(a)}}}--{a=a}}
\nlp{6.5}\*?>x
\formula p5|{..{a=a}.{*f.f.{{\df(a)}.{\df(a)}}}}
\named "$\ka$"
\separator
\use p3"I"
\formula p5|{..{f(a)}.{f(a)}}
\nlp5\*?_"\*?>u
\formula p5|{*f.f.{{\df(a)}.{\df(a)}}}
\named "$\kb$"
\bcc3/m$\ka$:. \*__"\rep3\*?--\ecc
\formula p5|{=. {a=a}}
\named "IIIe"

```

13. Terminal symbols of the Grundgesetze

The *Grundgesetze* has not such complex constructions as the *Begriffsschrift* (see Section 9). But it contains a lot of special symbols.

- | | |
|---|---|
| 1. $\backslash G! : \backslash \xi$. | 16. $\backslash Gfm : u \simeq q$. |
| 2. $\backslash kasl : \dot{\alpha} \Phi(\alpha)$. | 17. $\backslash Gsm : p \simeq q$. |
| 3. $\backslash G\bar{\cdot} : a \sim b$. | 18. $\backslash G< : p \nabla q$. |
| 4. $\backslash GI : Ip$. | 19. $\backslash Gy : p \underline{\Delta} q$. |
| 5. $\backslash G> : \rangle p$. | 20. $\backslash Ge : p \dot{\Delta} q$. |
| 6. $\backslash GU : \Downarrow p$ (or $\Downarrow p$). | 21. $\backslash Gc : p \dot{\Delta} q$. |
| 7. $\backslash Gn : \#u$ (or $\#u$). | 22. $\backslash Gmf : p \simeq q$. |
| 8. $\backslash GO : \emptyset$. | 23. $\backslash G* : *t$. |
| 9. $\backslash G1 : 1$. | 24. $\backslash G6 : \ddot{\sigma}s$. |
| 10. $\backslash Gbf$ or $\backslash Gff : f$. | 25. $\backslash Gf : fs$. |
| 11. $\backslash G- : \underline{\Delta} q$. | 26. $\backslash GF : s \#u$. |
| 12. $\backslash G\bar{\cdot} : \simeq q$. | 27. $\backslash G1 : s \dot{\Delta} u$. |
| 13. $\backslash G_ : p \underline{\Delta} q$. | 28. $\backslash Gp : ps$. |
| 14. $\backslash Guu : \supset$. | 29. $\backslash Gss$ or $\backslash Gsz : s \# p$. |
| 15. $\backslash G; : o; a$. | |

The symbols are constructed with the Computer Modern and $\mathcal{A}\mathcal{M}\mathcal{S}$ - TEX fonts. But with the external flag $\backslash fgefontknown$ the second symbol in numbers 6 and 7 is selected (see Section 3).

Several symbols are rotated and this requires the package `rotate.tex`. The output of unrotated symbols is activated by default; the external flag $\backslash let\showrotation=t$ must be given to activate rotation (see Section 3).

14. Typesetting definitions

The following command is provided to typeset definitions:

- $\backslash gfdefinition p<n><J> <j>\{<f>\}=<e>$ outputs the formula $\backslash frege<j>\{<f>\}$ in parentheses and sets it equal to the new symbol $<e>$ if the parameter $<J>$ is either $|$ to type a judgment or $.$ for a definition. In the style of the *Grundgesetze* the parameter $<J>$ can also be $:\cdot$. In this case write $<j><f>$ and this is interpreted as a math formula and not as a formula created by $\backslash frege$. The parameter $<n>$ is the position parameter to move the output the given amount of units to the right.

The output of a definition depends on the style.

Example 11. The definition in formula 69 of the *Begriffsschrift* is:

$$\Vdash \left(\left(\begin{array}{c} \overline{\delta} \quad \overline{\alpha} \\ \hline \begin{array}{c} F(\alpha) \\ f(\delta, \alpha) \\ F(\delta) \end{array} \end{array} \right) \equiv \begin{array}{c} \delta \\ | \\ \alpha \end{array} \left(\begin{array}{c} F(\alpha) \\ f(\delta, \alpha) \end{array} \right) \right)$$

In the style of the *Grundgesetze* the output is:

$$\Vdash \overline{\delta} \begin{array}{c} \overline{\alpha} \\ \hline \begin{array}{c} F(\alpha) \\ f(\delta, \alpha) \\ F(\delta) \end{array} \end{array} = \begin{array}{c} \delta \\ | \\ \alpha \end{array} \left(\begin{array}{c} F(\alpha) \\ f(\delta, \alpha) \end{array} \right)$$

And here is the source (note that $\backslash 1$ is a normal macro and needs no macro expansion):

```
\def\1#1{\gfvererbe\kd\ka{#1(\ka)}{f(\kd,\ka)}}
\gfdefinition p4. .{* .d. {..{* .a. {..{F(\da)}
. {f(\dd,\da)}}}}
. {F(\dd)}}}=\{1F}
```

Example 12. Here is an example of the *Grundgesetze*, in which the <J> is a colon.

$$\vDash \dot{\epsilon}(o \frown (a \frown \epsilon)) = o; a \quad (\Xi)$$

And this is the source:

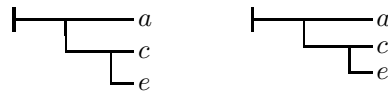
```
\gfdefinition p4: \kes1(o\G^(a\G^\ke))={o\G;a}
\named "$\gX$"
```

15. Commands to improve page breaks

The formulas are never broken. A page or column break must appear at a decollator, separator, or after an inference rule. To gain some space the formulas can be typeset in a smaller height and depth. (`\offinterlineskip` should be set.)

1. `\gfbssetzeabstand<h><d>` sets the height and depth between the lines to <h>pt and <d>pt, resp.;
2. `\gfbsreduziereabstandum<nn>` reduces the height and depth each by <nn>% (<nn> is a two-digit number);
3. `\gfbsabstandzuruecksetzen` sets height and depth to the values of `\strutbox`.

Example 13. The right formula uses `\gfbsreduziereabstandum20`:



And this is the code:

```
\offinterlineskip
\setbox0=\hbox to .3\hsize{\vtop to 30pt{%
  \nlp4\*_[-\*--:\*---\ce{\$a\$}
  \nlp4\*___\*_'\-*--:\ce{\$c\$}
  \nlp4\*___\*___\*_'\ce{\$e\$}\vss}\hss}
\setbox1=\hbox to .3\hsize{\vtop to 30pt{%
  \nlp1\*_[-\*--:\*---\ce{\$a\$}
  \gfbsreduziereabstandum20
  \nlp1\*___\*_'\-*--:\ce{\$c\$}
  \nlp1\*___\*___\*_'\ce{\$e\$}\vss}\hss}
\centerline{\box0 \box1 }
\gfbsabstandzuruecksetzen
```

16. List of parameters

Nearly all aspects of the lines that build the structure of Frege's notation is controlled by parameters. They can be changed to fit better with a certain type of font. But note that a couple of tests have been made with these values to guarantee, for example, that the affirmation can be placed in front of a quantification.

Here is the list of registers that make up the style of the *Begriffsschrift*.

1. `\gfbsvolleeinheit = 0.57 em`; width of one `*` triple
2. `\gfbsraise = 0.5 ex`; height of the horizontal line
3. `\gfbsneg = 0.25 ex`; height of the negation indicator
4. `\gfbsstrichdicke = 0.5 pt`; thickness of the horizontal line
5. `\gfbsurteildicke = 1 pt`; thickness of the judgment stroke
6. `\gfbsuht = 1.5 ex`; height of the judgment stroke
7. `\gfbsudp = 0.5 ex`; depth of the judgment stroke

8. `\gfbsersetzungdicke = 0.8 pt`; thickness of the substitution line
9. `\gfbsefdefdicke = 0.75 pt`; thickness of the definition strokes
10. `\gfbsschlussabstand = 2.5 pt`; distance of double lines in an inference
11. `\gfbsschlussdicke = 0.8 pt`; thickness of the second line in an inference
12. `\gfbmdpabstand = 0.2 em`; distance between colons in front of an inference
13. `\gfbsmaxanzahlzeilen = 25`; the maximal amount of lines in a single formula of the short form (the package declares for each line one or two token and skip registers; two registers of each kind are needed if substitutions are allowed);
14. `\gfbsegnepct = 70`; when `\ifgfbsegnedirect` is set to false the negation indicator has only a height of this percentage so that a gap occurs.

And this is the list of additional dimen parameters for the style of the *Grundgesetze*.

15. `\gfggstrichdicke = 0.58 pt`; common line thickness
16. `\gfggraise = 0.14 ex`; height of horizontal line
17. `\gfggneg = 0.47 ex`; height of negation indicator
18. `\gfgguht = 1.4 ex`; height of the judgment stroke
19. `\gfggudp = 0.9 ex`; depth of the judgment stroke
20. `\gfggdpabstand = 0.06 em`; distance between colons in front of an inference

17. Related work

Other people have worked on the problem to typeset Frege’s notation in \TeX too. J. Parsons has developed a package *begriff.sty* for \LaTeX (see [7]). Later the package was improved by Q. Pamp ([8]). A team that translated the *Grundgesetze* created its own package from *begriff.sty* and called it *grundgesetze.sty* (see [9]). It uses the fonts [5] that J. J. Green has developed for the special terminal symbols in the *Grundgesetze*.

18. References

- [1] Gottlob Frege, *Begriffsschrift, eine der arithmetischen nachgebildete Formelsprache des reinen Denkens* (Halle an der Saale: Louis Nebert, 1879)
“Begriffsschrift, a formula language, modeled upon that of arithmetic, for pure thought,” in: Jean van Heijenoort (ed.), *From Frege to Gödel: A Sourcebook in Mathematical Logic, 1879–1931*, (Cambridge, MA: Harvard University Press, 1967), 1–82; translation by S. Bauer-Mengelberg
- [2] Gottlob Frege, *Grundgesetze der Arithmetik — begriffsschriftlich abgeleitet* (Jena: Hermann Pohle, Volume 1 1893, Volume 2 1903)
Basic Laws of Arithmetic (Oxford: Oxford Univ. Press, 2013); translation by Philip A. Ebert, Marcus Rossberg, Crispin Wright
- [3] American Mathematical Society, *amstex*, <http://www.ams.org/publications/authors/tex/tex> or <http://www.ctan.org/tex-archive/macros/amstex> (accessed: 2014-11-29)
- [4] *rotate*, <http://www.ctan.org/tex-archive/macros/plain/contrib/misc> (accessed: 2014-11-29)
- [5] J. J. Green, *fge*, <http://www.ctan.org/tex-archive/fonts/fge> (accessed: 2014-11-29)
- [6] Silvio Levy, “Using Greek fonts with \TeX ,” *TUGboat* **9,1** (1988), 20–24
- [7] Josh Parsons, *begriff.sty*, <http://www.ctan.org/tex-archive/macros/latex/contrib/begriff> (accessed: 2014-11-29)
- [8] Quirin Pamp, *frege.sty*, <http://www.ctan.org/tex-archive/macros/latex/contrib/frege> (accessed: 2014-11-29)
- [9] Marcus Rossberg, *grundgesetze.sty*, <http://www.ctan.org/tex-archive/macros/latex/contrib/grundgesetze> (accessed: 2014-11-29)