

# The `ebproof` package

Emmanuel Beffara  
[manu@beffara.org](mailto:manu@beffara.org)

Version 2.0  
March 17, 2017

<b>Contents</b>			
<b>1 Introduction</b>	<b>1</b>	4.1 General shape . . . . .	4
<b>2 Environments</b>	<b>1</b>	4.2 Spacing . . . . .	5
<b>3 Statements</b>	<b>2</b>	4.3 Shape of inference lines . . . . .	6
3.1 Basic statements . . . . .	2	4.4 Format of conclusions . . . . .	7
3.2 Modifying proof trees . . . . .	3	4.5 Format of labels . . . . .	7
<b>4 Options</b>	<b>4</b>	4.6 Style macros . . . . .	8
		<b>5 License</b>	<b>9</b>
		<b>6 History</b>	<b>9</b>

## 1 Introduction

The `ebproof` package provides commands to typeset proof trees, in the style of sequent calculus and related systems:

$$\frac{\Gamma, A \vdash B}{\Gamma \vdash A \rightarrow B} \text{ abs} \quad \frac{\Gamma \vdash A}{\Gamma \vdash B} \text{ app}$$

```
\begin{prooftree}
  \hypo{ \Gamma, A &\vdash B }
  \infer1[abs]{ \Gamma &\vdash A \to B }
  \hypo{ \Gamma &\vdash A }
  \infer2[app]{ \Gamma &\vdash B }
\end{prooftree}
```

The structure is very much inspired by the `bussproofs` package, in particular for the postfix notation. I actually wrote `ebproof` because there were some limitations in `bussproofs` that I did not know how to lift, and also because I did not like some choices in that package (and also because it was fun to write).

## 2 Environments

The package provides the `prooftree` environment, in a standard and starred variants.

```
\begin{prooftree}[\langle options \rangle] \langle statements \rangle \end{prooftree}
```

Typeset the proof tree described by the  $\langle statements \rangle$ , as described in section 3. The  $\langle options \rangle$  provide default formatting options for the proof tree. This environment can be used either in math mode or in text mode. It produces a proof tree at the current position in the text flow.

`\begin{prooftree*}[\langle options \rangle] \langle statements \rangle \end{prooftree*}`

Typeset the proof on a line of its own; it is essentially equivalent to wrapping the `prooftree` environment inside a displayed math environment.

The starred version is used in situations when a single proof will be displayed. The non-starred version is useful in order to integrate the proof into some larger structure, like two parts of a formula:

$$\frac{\frac{\overline{\vdash A} \quad \overline{\vdash B}}{\vdash B, C}}{\vdash A \wedge B, C} \rightsquigarrow \frac{\overline{\vdash A} \quad \overline{\vdash B}}{\vdash A \wedge B}$$

```

\l[
  \begin{prooftree}
    \infer0{ \vdash A }
    \hypo{ \vdash B } \infer1{ \vdash B, C }
    \infer2{ \vdash A \wedge B, C }
  \end{prooftree}
\quad \rightsquigarrow \quad
\begin{prooftree}
  \infer0{ \vdash A } \hypo{ \vdash B }
  \infer2{ \vdash A \wedge B }
  \infer1{ \vdash A \wedge B, C }
\end{prooftree}
\r]

```

### 3 Statements

Statements describe proofs in postfix notation: when typesetting a proof tree whose last rule has, say, two premisses, you will first write statements for the subtree of the first premiss, then statements for the subtree of the second premiss, then a statement like `\infer2{\langle conclusion \rangle}` to build an inference with these two subtrees as premisses and the given text as conclusion.

Hence statements operate on a stack of proof trees. At the beginning of a `prooftree` environment, the stack is empty. At the end, it must contain exactly one tree, which is the one that will be printed.

Note that the commands defined in this section only exist right inside `prooftree` environments. If you have a macro with the same name as one of the statements, for instance `\hypo`, then this macro will keep its meaning outside `prooftree` environments as well as inside the arguments of a statement. If you really need to access the statements in another context, you can always call them by prefixing their names with `ebproof`, for instance as `\ebproofhypo`.

#### 3.1 Basic statements

The basic statements for building proofs are the following, where `\langle options \rangle` stands for arbitrary options as described in section 4.

`\hypo[\langle options \rangle]{\langle text \rangle}`

Push a new proof tree consisting only in one conclusion line, with no premiss and no line above, in other words a tree with only a leaf (`\hypo` stands for *hypothesis*).

`\infer[\langle options \rangle]{\langle arity \rangle}[\langle label \rangle]{\langle text \rangle}`

Build an inference step by taking some proof trees from the top of the stack, assembling them with a rule joining their conclusions and putting a new conclusion below. The `\langle arity \rangle` is the number of sub-proofs, it may be any number including 0 (in this case there will be a line above the conclusion but no sub-proof). If `\langle label \rangle` is present, it is used as the label on the right of the inference line; it is equivalent to using the `right label` option.

The  $\langle text \rangle$  in these statements is the contents of the conclusion at the root of the tree that the statements create. It is typeset in math mode by default but any kind of formatting can be used instead, using the `template` option. The  $\langle label \rangle$  text is formatted in horizontal text mode by default.

Each proof tree has a vertical axis, used for alignment of successive steps. The position of the axis is deduced from the text of the conclusion at the root of the tree: if  $\langle text \rangle$  contains the alignment character `&` then the axis is set at that position, otherwise the axis is set at the center of the conclusion text. The `\infer` statement makes sure that the axis of the premiss is at the same position as the axis of the conclusion. If there are several premisses, it places the axis at the center between the left of the leftmost conclusion and the right of the rightmost conclusion:

$\frac{\frac{\frac{\vdash A, B, C}{A \vdash B, C}}{A, B \vdash C} \quad D \vdash E}{A, B, D \vdash C, E} \quad \frac{A, B \vdash C, D, E}{A \vdash B, C, D, E}$	<pre> \begin{prooftree}   \hypo{ &amp;\vdash A, B, C }   \infer1{ A &amp;\vdash B, C }   \infer1{ A, B &amp;\vdash C }   \hypo{ D &amp;\vdash E }   \infer2{ A, B, D &amp;\vdash C, E }   \infer1{ A, B &amp;\vdash C, D, E }   \infer1{ A &amp;\vdash B, C, D, E } \end{prooftree} </pre>
---	--

`\ellipsis{ $\langle label \rangle$ }{ $\langle text \rangle$ }`

Typeset vertical dots, with a label on the right, and a new conclusion. No inference lines are inserted.

$\begin{array}{l} \Gamma \vdash A \\ \vdots \text{foo} \\ \Gamma \vdash A, B \end{array}$	<pre> \begin{prooftree}   \hypo{ \Gamma &amp;\vdash A }   \ellipsis{foo}{ \Gamma &amp;\vdash A, B } \end{prooftree} </pre>
---	--

### 3.2 Modifying proof trees

The following additional statements may be used to affect the format of the last proof tree on the stack:

`\rewrite{ $\langle code \rangle$ }`

Rewrite the proof while preserving its size and alignment. The  $\langle code \rangle$  is typeset in horizontal mode, with the following control sequences defined:

- `\treebox` is a box register that contains the original material,
- `\treemark{ $\langle name \rangle$ }` expands as the position of a given mark with respect to the left of the box.

A simple use of this statement is to change the color of a proof tree:

$\frac{\frac{\Gamma, A \vdash B}{\Gamma \vdash A \rightarrow B} \text{ abs} \quad \Gamma \vdash A}{\Gamma \vdash B} \text{ app}$	<pre> \begin{prooftree}   \hypo{ \Gamma, A &amp;\vdash B }   \infer1[abs]{ \Gamma &amp;\vdash A \to B }   \rewrite{\color{red}\box\treebox}   \hypo{ \Gamma &amp;\vdash A }   \infer2[app]{ \Gamma &amp;\vdash B } \end{prooftree} </pre>
--	---

Note the absence of spaces inside the call to `\rewrite`, because spaces would affect the position of the tree box. Note also that explicit use of `\treebox` is required to actually draw the subtree. Not using it will effectively not render the subtree, while still reserving its space in the enclosing tree:

$$\frac{\Gamma \vdash A}{\Gamma \vdash B} \text{ app}$$

```

\begin{prooftree}
  \hypo{ \Gamma, A &\vdash B }
  \infer1[abs]{ \Gamma &\vdash A \to B }
  \rewrite{}
  \hypo{ \Gamma \vdash A }
  \infer2[app]{ \Gamma \vdash B }
\end{prooftree}

```

This kind of manipulation is useful for instance in conjunction with the `beamer` package to allow revealing subtrees of a proof tree progressively in successive slides of a given frame.

`\delims{⟨left⟩}{⟨right⟩}`

Put left and right delimiters around the whole sub-proof, without changing the alignment (the spacing is affected by the delimiters, however). The `⟨left⟩` text must contain an opening occurrence of `\left` and the `⟨right⟩` text must contain a matching occurrence of `\right`. For instance, `\delims{\left(}{\right)}` will put the sub-proof between parentheses.

$$\frac{A_1 \vee \dots \vee A_n \quad \left( \begin{array}{c} [A_i] \\ \vdots \\ B \end{array} \right)_{1 \leq i \leq n}}{B}$$

```

\begin{prooftree}
  \hypo{ A_1 \vee \cdots \vee A_n }
  \hypo{ [A_i] }
  \ellipsis{}{ B }
  \delims{ \left( }{ \right)_{\{1 \leq i \leq n\}} }
  \infer2{ B }
\end{prooftree}

```

## 4 Options

The formatting of trees, conclusion texts and inference rules is affected by options, specified using the L<sup>A</sup>T<sub>E</sub>X3 key-value system. All options are in the `ebproof` module in the key tree. They can be set locally for a proof tree or for a single statement using optional arguments in the associated commands.

`\ebproofset{⟨options⟩}`

Set some options. The options will apply in the current scope; using this in preamble will effectively set options globally. Specific options may also be specified for each proof tree and for each statement in a proof tree, using optional arguments.

### 4.1 General shape

The options in this section only make sense at the global level and at the proof level. Changing the proof style inside a `proof` environment has undefined behaviour.

`proof style=⟨name⟩`

Set the general shape for representing proofs. The following styles are provided:

upwards

This is the default style. Proof trees grow upwards, with conclusions below and premisses above.

`downwards`

Proof trees grow downwards, with conclusions above and premisses below.

$$\frac{\frac{\Gamma \vdash B}{\Gamma \vdash A \rightarrow B} \text{ abs} \quad \Gamma \vdash A}{\Gamma, A \vdash B} \text{ app}$$

```

\begin{prooftree}[proof style=downwards]
  \hypo{ \Gamma, A &\vdash B }
  \infer1[abs]{ \Gamma &\vdash A \to B }
  \hypo{ \Gamma &\vdash A }
  \infer2[app]{ \Gamma &\vdash B }
\end{prooftree}

```

In the optional argument of `prooftree` environments, proof styles can be specified directly, without prefixing the name by “`proof style=`”. For instance, the first line of the example above could be written `\begin{prooftree}[downwards]` equivalently.

`center=<boolean>`

If set to `true`, the tree produced by the `prooftree` environment will be vertically centered around the text line. If set to `false`, the base line of the tree will be the base line of the conclusion. The default value is `true`.

$$\frac{}{A \vdash A} \quad \frac{\Gamma, A \vdash B}{\Gamma \vdash A \rightarrow B}$$

```

\begin{prooftree}[center=false]
  \infer0{ A \vdash A }
\end{prooftree}
\quad
\begin{prooftree}[center=false]
  \hypo{ \Gamma, A \vdash B }
  \infer1{ \Gamma \vdash A \to B }
\end{prooftree}

```

## 4.2 Spacing

`separation=<dimension>`

The horizontal separation between sub-proofs in an inference. The default value is `1.5em`.

$$\frac{\frac{A \quad B}{C} \quad \frac{D \quad E \quad F}{G} \quad H}{K}$$

```

\begin{prooftree}[separation=0.5em]
  \hypo{ A } \hypo{ B } \infer2{ C }
  \hypo{ D } \hypo{ E } \hypo{ F } \infer3{ G }
  \hypo{ H } \infer[separation=3em]3{ K }
\end{prooftree}

```

`rule margin=<dimension>`

The spacing above and below inference lines. The default value is `0.7ex`.

$$\frac{\frac{\Gamma, A \vdash B}{\Gamma \vdash A \rightarrow B} \text{ abs} \quad \Gamma \vdash A}{\Gamma \vdash B} \text{ app}$$

```

\begin{prooftree}[rule margin=2ex]
  \hypo{ \Gamma, A &\vdash B }
  \infer1[abs]{ \Gamma &\vdash A \to B }
  \hypo{ \Gamma &\vdash A }
  \infer2[app]{ \Gamma &\vdash B }
\end{prooftree}

```

### 4.3 Shape of inference lines

`rule style=<name>`

Set the shape of inference lines. The following values are provided:

`simple`

A simple horizontal rule is drawn. This is the default style.

`no rule`

No inference line is drawn. A single space of the length of `rule margin` is inserted.

`double`

A double line is drawn.

`dashed`

A single dashed line is drawn.

The precise rendering is influenced by parameters specified below. Arbitrary new shapes can be defined using the `rule code` option described below and the `\ebproofnewrulestyle` command described in section 4.6.

In the optional argument of the `\infer` statement, rule styles can be specified directly, without prefixing the style name by “`rule style=`”. For instance, `\infer[dashed]` is equivalent to `\infer[rule style=dashed]`.

$$\frac{\frac{\Gamma \vdash A \rightarrow B \quad \frac{\Delta \vdash A \quad B \vdash B}{\Delta, !A \multimap B \vdash B}}{\Gamma, \Delta \vdash B}}{\Gamma \cup \Delta \vdash B}$$

```
\begin{prooftree}
\hypo{ \Gamma &\vdash A \to B }
\infer[no rule]1{ \Gamma &\vdash {!A} \multimap B }
\hypo{ \Delta &\vdash A }
\infer[rule thickness=2pt]1{ \Delta &\vdash {!A} }
\infer0{ B \vdash B }
\infer[dashed]2{ \Delta, {!A} \multimap B \vdash B }
\infer2{ \Gamma, \Delta &\vdash B }
\infer[double]1{ \Gamma \cup \Delta &\vdash B }
\end{prooftree}
```

`rule thickness=<dimension>`

The thickness of inference lines. It is 0.4pt by default.

`rule separation=<dimension>`

The distance between the two lines in the `double` rule style. It is 2pt by default.

`rule dash length=<dimension>`

The length of dashes in the `dashed` rule style. It is 0.2em by default.

`rule dash space=<dimension>`

The space between dashes in the `dashed` rule style. It is 0.3em by default.

`rule code=<code>`

This option is used to define an arbitrary shape for rules. The `<code>` is used to render the rule, it is executed in vertical mode in a `\vbox` whose `\hsize` is set to the width of the rule. Margins above and below are inserted automatically (they can be removed by setting `rule margin` to `0pt`).

$\frac{\Gamma \vdash A}{\Gamma \vdash A, \dots, A \quad \Delta, A, \dots, A \vdash \Theta} \Gamma, \Delta \vdash \Theta$	<pre> \begin{prooftree}[rule code={\hbox{\tikz   \draw[decorate,decoration={snake,amplitude=.3ex]}   (0,0) -- (\hsize,0);}}]   \hypo{ \Gamma &amp;\vdash A }   \infer1{ \Gamma &amp;\vdash A, \ldots, A }   \hypo{ \Delta, A, \ldots, A \vdash \Theta }   \infer2{ \Gamma, \Delta \vdash \Theta } \end{prooftree} </pre>
--	--

Note that this example requires the `tikz` package, with the `decorations.pathmorphing` library for the `snake` decoration.

#### 4.4 Format of conclusions

```

template=<code>
left template=<code>
right template=<code>

```

Defines how conclusions are formatted. The code is arbitrary T<sub>E</sub>X code, composed in horizontal mode. The macro `\inserttext` is used to insert the actual text passed to the `\hypo` and `\infer` statements. The `template` value is used for conclusions with no alignment mark. The `left template` and `right template` values are used on the left and right side of the alignment mark when it is present. The default value for `template` is simply `$(\inserttext)$`, so that conclusions are set in math mode. The default values for `left template` and `right template` are similar, with spacing assuming that a relation symbol is put near the alignment mark, so that `\infer1{A &\vdash B}` is spaced correctly.

$\frac{\text{( foo )} \quad \text{( bar )}}{\text{( quux )} \quad \text{( baz )}}$	<pre> \begin{prooftree}[template=(\textbf{\inserttext})]   \hypo{ foo }   \hypo{ bar }   \infer1{ baz }   \infer2{ quux } \end{prooftree} </pre>
--	--

#### 4.5 Format of labels

```

left label=<text>
right label=<text>

```

The text to use as the labels of the rules, on the left and on the right of the inference line. Using the second optional argument in `\infer` is equivalent to setting the `right label` option with the value of that argument.

$$\lambda \frac{\Gamma, A \vdash B}{\Gamma \vdash A \rightarrow B} \text{abs} \quad \text{app} \quad \frac{\Gamma \vdash A}{\Gamma \vdash B}$$

```

\begin{prooftree}
  \hypo{ \Gamma, A \vdash B }
  \infer[left label=\lambda$]1[abs]
    { \Gamma \vdash A \to B }
  \hypo{ \Gamma \vdash A }
  \infer[left label=@]2[app]{ \Gamma \vdash B }
\end{prooftree}

```

`left label template=code`  
`right label template=code`

Defines how rule labels are formatted. The code is arbitrary T<sub>E</sub>X code, composed in horizontal mode. The macro `\inserttext` can be used to insert the actual label text, as defined by the options above. The default values are simply `\inserttext` so that labels are set in plain text mode.

`label separation=dimension`

The spacing between an inference lines and its labels. The default value is `0.5em`.

`label axis=dimension`

The height of the horizontal axis used for aligning the labels with the rules. The default value is `0.5ex`.

## 4.6 Style macros

The following commands allow for the definition of custom styles using the basic style options, in a way similar to PGF’s “styles” and L<sup>A</sup>T<sub>E</sub>X3’s “meta-keys”. This allows setting a bunch of options with the same values in many proofs using a single definition.

`\ebproofnewstyle{name}{options}`

Define a new style option *name* that sets the given *options*.

For instance, the following code defines a new option `small` that sets various parameters so that proofs are rendered smaller.

$$\frac{\Gamma, A \vdash B}{\Gamma \vdash A \rightarrow B} \quad \Gamma \vdash A \quad \Gamma \vdash B$$

```

\ebproofnewstyle{small}{
  separation = 1em, rule margin = .5ex,
  template = \footnotesize\inserttext$ }
\begin{prooftree}[small]
  \hypo{ \Gamma, A \vdash B }
  \infer1{ \Gamma \vdash A \to B }
  \hypo{ \Gamma \vdash A } \infer2{ \Gamma \vdash B }
\end{prooftree}

```

`\ebproofnewrulestyle{name}{options}`

Define a new rule style. The *options* part includes options used to set how to draw rules in the new style.

The option `rule code` is useful in this command as it allows to define arbitrary rule styles. For instance, the squiggly rule example above could be turned into a new rule style `zigzag` with the following code:

$$\frac{\Gamma \vdash A}{\Gamma \vdash A, \dots, A \quad \Delta, A, \dots, A \vdash \Theta} \quad \Gamma, \Delta \vdash \Theta$$

```

\ebproofnewrulestyle{zigzag}{
  rule code = {\hbox{\tikz
    \draw[decorate,decoration={snake,amplitude=.3ex}]
      (0,0) -- (\hsize,0);}}
\begin{prooftree}
  \hypo{ \Gamma &\vdash A }
  \infer1{ \Gamma &\vdash A, \ldots, A }
  \hypo{ \Delta, A, \ldots, A \vdash \Theta }
  \infer[zigzag]2{ \Gamma, \Delta \vdash \Theta }
\end{prooftree}

```

## 5 License

This work may be distributed and/or modified under the conditions of the L<sup>A</sup>T<sub>E</sub>X Project Public License, either version 1.3 of this license or (at your option) any later version. The latest version of this license is in

<http://www.latex-project.org/lppl.txt>

and version 1.3 or later is part of all distributions of L<sup>A</sup>T<sub>E</sub>X version 2005/12/01 or later.

This work has the LPPL maintenance status ‘maintained’.

The Current Maintainer of this work is Emmanuel Beffara.

This work consists of the files `ebproof.sty` and `ebproof.tex`.

## 6 History

This section lists the principal evolutions of the package, in reverse chronological order.

**Version 2.0** A complete rewrite of the code using the L<sup>A</sup>T<sub>E</sub>X3 programming environment. The incompatible changes from the user’s point of view are the following:

- Proof statements are now writtten in lowercase (i.e. `\Infer` is now written `\infer` etc.) but the syntax is otherwise unchanged. The old uppercase commands still work but produce a deprecation warning, they will be removed in a future version.
- New styles are now defined using `\ebproofnewstyle` and `\ebproofnewrulestyle`. The previous method using PGF styles does not work anymore (because PGF is not used anymore).

The new commands and options are the following:

- The statement `\rewrite` generalizes `\Alter`,
- The option `label axis` controls vertical alignment of labels.

**Version 1.1** A bugfix release. In `template` options, one now uses `\inserttext` instead of `#1` for the text arguments, which improves robustness.

**Version 1.0** The first public release.