PROOF TREES IN LaTeX

MARCO BENINI

1. Introduction

Writing proofs in natural deduction or in similar, tree-like calculi, is always a challenge: from the typographical point of view, these proofs are complex objects that cannot be simply typeset using the standard LaTeX commands. Thus, many packages have been developed: Sam Buss's `bussproofs.sty`, `http://math.ucsd.edu/~sbuss/ResearchWeb/bussproofs/`; Makoto Tatsuta's `proof.sty`, `http://research.nii.ac.jp/~tatsuta/proof-sty.html`; and `prooftree.sty` by Paul Taylor, `http://mirror.ctan.org/macros/generic/proofs/taylor`.

All these packages have their merits and weaknesses. For example, Buss's package is extremely flexible but inference rules with more than five assumptions cannot be directly typeset. On the other hand, Tatsuta's package provides a very simple set of commands doing a fine job, but customisation is very limited. Taylor's package provides a natural syntax for writing proofs, but customisation is limited, and the package has an expire date.

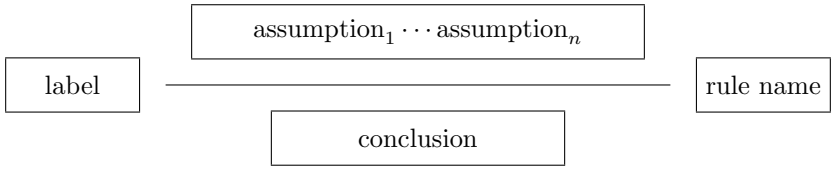
The package presented in the following provides most of the features which are already present in Buss's package, coupled with some new ones. This package uses a syntax which is closer to Tatsuta's one, but almost all the typesetting process is parametric, so that each bit of a proof can be customised at will.

The graphical appearance of a proof is similar to the one obtained using Taylor's package, but the additional features allow to set up the graphical output to follow the style of some of the standard textbooks, e.g., A.S. Troelstra and H. Schwichtenberg, *Basic Proof Theory*, Cambridge University Press (2000).

2. Basic Commands

The package is invoked by putting `\usepackage{prftree.sty}` in the preamble of the document, and installation reduces to put the file `prftree.sty` somewhere in the LaTeX search path.

A proof tree constructs a box with the following internal structure:



In turn, each assumption is typeset as a box which has usually the shape of another proof tree, while the rule name and the label are typeset in a text box, and the conclusion in a math box. The aspect of the proof line is controlled by suitable options, as is the presence of the rule name and of the label. Options cover other aspects of the graphical rendering of a proof tree, as it will be explained later. The basic command to build a proof tree is `\prftree`.

For example, the proof of $A \supset \neg\neg A$ in natural deduction is:

$$\dfrac{\dfrac{\dfrac{[A] \quad [\neg A]}{\bot}\supset\mathrm{E}}{\neg\neg A}\supset\mathrm{I}}{A \supset \neg\neg A}\supset\mathrm{I}$$

This proof is generated by the following LaTeX code:

```
\begin{displaymath}
  \prftree[r]{$\scriptstyle\supset\mathrm{I}$}
  {\prftree[r]{$\scriptstyle\supset\mathrm{I}$}
    {\prftree[r]{$\scriptstyle\supset\mathrm{E}$}
      {\prfboundedassumption{A}}
      {\prfboundedassumption{\neg A}}
      {\bot}}
    {\neg\neg A}}
  {A \supset \neg\neg A}
\end{displaymath}
```

In general, the syntax of the `\prftree` command is:

`\prftree`[options] ⋯ [options]{assumption$_1$} ⋯ {assumption$_n$}{conclusion}

Assumptions are optional and there may be any number of them. Each assumption may contain a proof tree, which is typeset independently: the order allows to use indentation to help reading the source. The conclusion is mandatory, and it is supposed to be a formula.

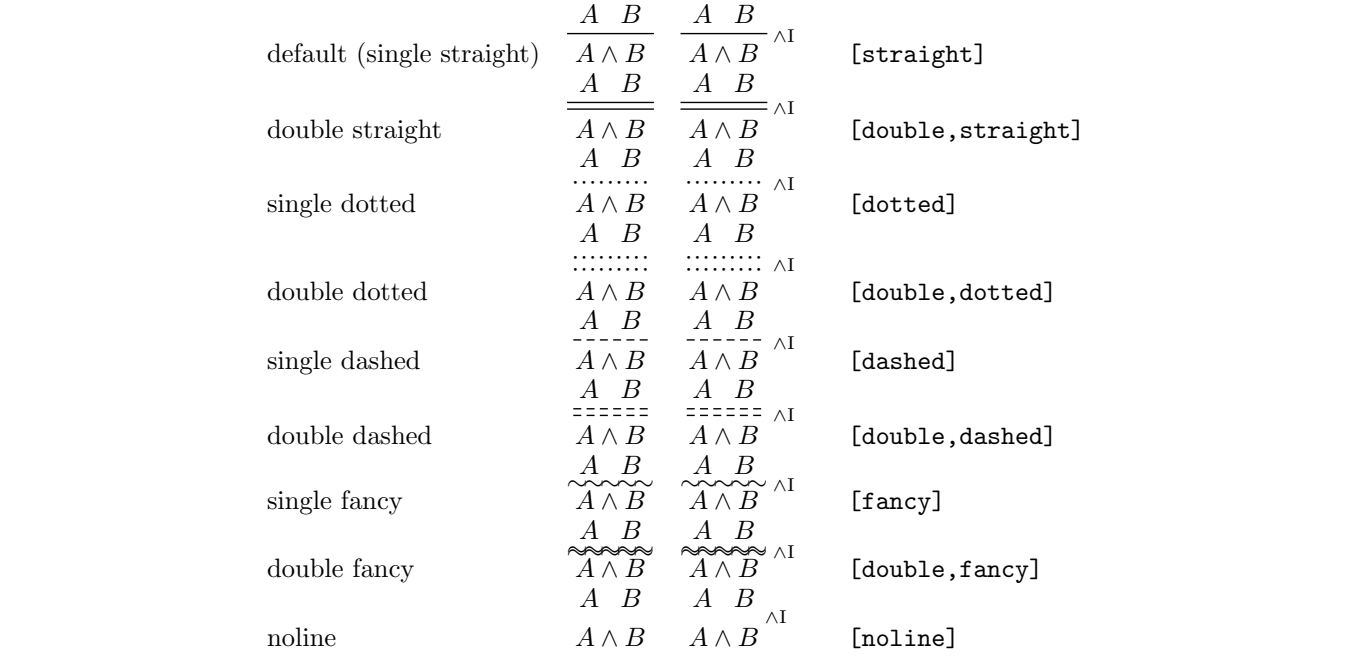
Assumptions and the conclusion are typeset in a display style math environment. Options control the way the proof is generated: in the example, the `r` option has been used to signal that the first argument of `\prftree` is the name of the inference rule.

The available options are:

- [**r**], [**rule**], [**by rule**], [**by**], [**right**]: the first argument after the options is the rule name, which is typeset in text mode;
- [**l**], [**left**], [**label**]: the first argument after the options is the label of the rule, which is typeset in text mode. If a rule name is present, the first argument is the rule name, and the second one is the label;
- [**straight**], [**straight line**], [**straightline**]: makes the proof line solid;
- [**dotted**], [**dotted line**], [**dottedline**]: makes the proof line dotted;
- [**dashed**], [**dashed line**], [**dashedline**]: makes the proof line dashed;
- [**f**], [**fancy**], [**fancy line**], [**fancyline**]: the proof line will be fancy;
- [**s**], [**single**], [**single line**], [**singleline**]: makes the proof line single;
- [**d**], [**double**], [**double line**], [**doubleline**]: makes the proof line double;
- [**noline**]: suppresses the proof line (prevails over all other line options);
- [**summary**]: renders the proof line as the summary symbol (prevails over all other line options except **noline**).

By default the proof line is straight and single. Options may be written in sequence, as in `[r,f,d]`, which means that the proof tree will have a rule name, and the proof line will be fancy and double, or separately, as in `[r][f][d]`, or even as a combination, like `[r][f,d]`. Options are evaluated left-to-right, so `[d,s]` is the same as `[s]`, while `[noline,straight,d]` is the same as `[noline]`.

The conjunction introduction rule illustrates the various line options:



These examples are implemented in an array whose cells have the form

```
\prftree[option]{A}{B}{A \wedge B} &
\prftree[option,r]{$\scriptstyle\wedge\mathrm{I}$}
```

in which the option part is the one on the right of the picture.

An assumption is a special proof tree, built by the command:

`\prfassumption`{formula}

Similarly, a bounded assumption is produced by the command:

\prfboundedassumption{formula}

as in the previous example.

Although it is possible to type assumptions directly as argument of \prftree, it is better to use the commands above: as explained later, since a proof tree is a box with an internal structure, the assumption commands take care of building this structure appropriately, while the direct typing does not, which may produce unexpected results.

Similarly, axioms are produced by the commands

\prfaxiom{axiom}

and

\prfbyaxiom{name}{axiom}

For example, the axiom stating that equality is reflexive, is

$$\frac{}{\forall x\, x = x} \quad \frac{}{\forall x\, x = x}\,\text{refl}$$

and they are generated by the LaTeX code

```
\prfaxiom{\forall x\, x = x}
\prfbyaxiom{refl}{\forall x\, x = x}
```

Finally, a proof summary is used to summarise a proof. The corresponding command is:

\prfsummary[name]{assumption$_1$}$\cdots${assumption$_n$}{conclusion}

The name of the proof is optional, while the assumptions and the conclusion are treated as in \prftree. When present, the proof name is typeset in text mode.

For example, `\prfsummary{\forall x\, x = x}` produces

$$\begin{array}{c}\vdots\\ \forall x\, x = x\end{array}$$

while `\prfsummary[name]{A(x)}{B(y)}{B(y) \wedge A(x)}` gives

$$\begin{array}{cc} A(x) & B(y) \\ & \vdots\ \text{name} \\ & \vdots \\ B(y) \wedge A(x) & \end{array}$$

In general, a proof tree is a TeX box containing all the pieces of the tree, with strict bounds: for example,

$$\boxed{\begin{array}{cc} A(x) & B(y) \\ & \vdots\ \text{name} \\ & \vdots \\ B(y) \wedge A(x) & \end{array}}$$

3. Parameters

A number of parameters may be used to control the typesetting of proof trees. They may be changed globally or locally, following the usual scoping rules of TeX. In this respect, remember that each assumption is typeset independently, so parameters may be changed on a sub-proof basis, as will be done in most examples.

There are various TeX dimensions that influence how proofs are constructed:

- `\prflinepadbefore` (default 0.3ex): the space between the bottom line of assumptions and the proof line
- `\prflinepadafter` (default 0.3ex): the space between the proof line and the top of the conclusion;
- `\prflineextra` (default 0.3em): the length which extends on the left and on the right the proof line so that it is slightly longer than the largest between the conclusion and the list of (direct) assumptions;
- `\prflinethickness` (default 0.12ex): the thickness of the proof line;
- `\prfemptylinethickness` (default 4 times the line thickness): in the rare case when the line is empty, but there are assumptions, this is the distance between the assumptions and the conclusion;
- `\prfrulenameskip` (default 0.2em): the space between the proof line and the rule name;
- `\prflabelskip` (default 0.2em): the space between the proof label and the proof line;
- `\prfinterspace` (default .8em): the space between two subsequent assumptions in the assumption list;
- `\prfdoublelineinterspace` (default 0.06ex): the space between the two lines of a double line.

For example,

$$\dfrac{\dfrac{\dfrac{[A] \quad [\neg A]}{\bot}\supset\mathrm{E}}{\neg\neg A}\supset\mathrm{I}}{A \supset \neg\neg A}\supset\mathrm{I}$$

is typeset by

```
\prflinepadafter=0ex
\prftree[r]{$\supset$I}
  {\prftree[r]{$\supset$I}
    {\prftree[r]{$\supset$E}
      {\prfboundedassumption{A}}
      {\prfboundedassumption{\neg A}}
      {\bot}}
    {\neg\neg A}}
  {A \supset \neg\neg A}
```

Similarly, `\prflineextra=-.4em` and `\prfrulenameskip=.8em` produce:

$$\dfrac{\dfrac{\dfrac{[A] \quad [\neg A]}{\bot}\supset\mathrm{E}}{\neg\neg A}\supset\mathrm{I}}{A \supset \neg\neg A}\supset\mathrm{I}$$

Also, `\prflinethickness=3pt` and `\prfdoublelineinterspace=2pt` in the upper sub-proof generate:

$$\dfrac{\dfrac{\dfrac{[A]\quad[\neg A]}{\bot}\supset\mathrm{E}}{\neg\neg A}\supset\mathrm{I}}{A\supset\neg\neg A}\supset\mathrm{I}$$

The corresponding code is

```
\prftree[r]{$\supset$I}
  {\prftree[r]{$\supset$I}
    {\prflinethickness=3pt
      \prfdoublelineinterspace=2pt
      \prftree[r,d]{$\supset$E}
      {\prfboundedassumption{A}}
      {\prfboundedassumption{\neg A}}
      {\bot}}
    {\neg\neg A}}
  {A \supset \neg\neg A}
```

Line thickness does not affect dashed, dotted, and fancy lines, but interline space does: in the example, `\prfdoublelineinterspace=4pt` on a fancy line produces

$$\dfrac{\dfrac{\dfrac{[A]\quad[\neg A]}{\bot}\supset\mathrm{E}}{\neg\neg A}\supset\mathrm{I}}{A\supset\neg\neg A}\supset\mathrm{I}$$

Fancy lines are drawn by the `\prffancyline` command. This can be redefined: as a guideline, the package defines it as

```
\def\prffancyline{\cleaders\hbox to .63em%
  {\hss\raisebox{-.5ex}[.2ex][0pt]{$\sim$}\hss}\hfill}
```

Label spacing works exactly as rule name spacing. Actually, it is possible to have a proof with both a label and a rule name:

$$[\bot\mathrm{E}\ \textit{will not work here!}]\ \dfrac{\dfrac{\dfrac{[A]\quad[\neg A]}{\bot}\supset\mathrm{E}}{\neg\neg A}\supset\mathrm{I}}{A\supset\neg\neg A}\supset\mathrm{I}$$

which has been typeset by

```
\prftree[r]{$\supset$I}
  {\prflabelskip=.7em
    \prftree[r,l]{$\supset$I}
            {[\textsl{$\bot\mathrm{E}$ will not work here!}]}
    {\prftree[r]{$\supset$E}
      {\prfboundedassumption{A}}
      {\prfboundedassumption{\neg A}}
```

```
    {\bot}}
  {\neg\neg A}}
{A \supset \neg\neg A}
```

The `\prfinterspace` controls the distance between assumptions. Specifically, this is the space between the *boxes* containing two assumptions.

Consider the following example

$$\dfrac{\dfrac{\dfrac{\dfrac{[A \to (B \to C)] \quad [A]}{B \to C} \quad \dfrac{[A \to B] \quad [A]}{B}}{C}}{A \to C}}{(A \to B) \to (A \to C)}}{(A \to (B \to C)) \to ((A \to B) \to (A \to C))}$$

Although the assumptions in the top line are well spaced, the two sub-proofs on the top are too close. This can be corrected in two different ways: by putting an explicit space, via `\hspace`, in front of the second sub-proof, or after the first sub-proof—remember, they are just boxes

$$\dfrac{\dfrac{\dfrac{\dfrac{[A \to (B \to C)] \quad [A]}{B \to C} \qquad \dfrac{[A \to B] \quad [A]}{B}}{C}}{A \to C}}{(A \to B) \to (A \to C)}}{(A \to (B \to C)) \to ((A \to B) \to (A \to C))}$$

otherwise, putting `\prfinterspace` = 1.5em before the sub-proof whose conclusion is C, one obtains the more pleasant

$$\dfrac{\dfrac{\dfrac{\dfrac{[A \to (B \to C)] \quad [A]}{B \to C} \quad \dfrac{[A \to B] \quad [A]}{B}}{C}}{A \to C}}{(A \to B) \to (A \to C)}}{(A \to (B \to C)) \to ((A \to B) \to (A \to C))}$$

The `Strut` option of the package controls a subtle point about spacing around a proof line: assumptions and conclusion are usually typeset so that the height and the depth of their box is at least the one of `\mathstrut`. In this way, adjacent proofs will have their proof lines aligned (well, whenever they don't have huge conclusions). But, as signalled by Dominic Hughes, sometimes one wants the height and the depth to be the "real" ones, especially when there are no characters/symbols with a positive depth: this forces the perceived space above and below the proof line to be exactly the values of `\prflinepadbefore` and `\prflinepadafter`. This behaviour can be achieved by calling the package with the `STRUT` option. Alternatively, one may use the `\prfSTRUToptionfalse` command to locally force this behaviour, and `\prfSTRUToptiontrue` to return to the standard one. Similarly, the `STRUTlabel`

package option, together with the pair of commands `\prfSTRUTlabeloptiontrue` and `\prfSTRUTlabeloptionfalse`, operate on rule names and rule labels.

The rendering of bounded assumptions is modified by `\prfboundedstyle`. When `\prfboundedstyle` $= 0$, the format of the assumption is [formula], which is the default behaviour; with `\prfboundedstyle` $= 1$, the formula is cancelled by a horizontal line; with `\prfboundedstyle` > 1, the custom `\prfdiscargedassumption` command is invoked:

$$[A(x)] \qquad \cancel{A(x)} \qquad \langle A(x) \rangle$$

The `\prfdiscargedassumption` can be freely redefined. The package provides a reference implementation:

```
\def\prfdiscargedassumption#1{\left\langle{#1}\right\rangle}
```

Proof summaries are drawn according to `\prfsummarystyle`. The default value is 0, which produces a vertical dotted line. Setting `\prfsummarystyle` $= 1$ produces a huge Π, while `\prfsummarystyle` $= 2$ produces a $\prod$. The value 3 uses a $\mathcal{D}$ as the derivation symbol. Values greater than 3 force the summary to be rendered by the `\prffancysummarybox` command.

$$\begin{array}{lcccc}
 & & B(x) & A(y) \quad D(x) & \\
 & \vdots & \vdots & \vdots & \text{name} \\
\texttt{\textbackslash prfsummarystyle} = 0 & \forall x.\, x = x & A(x) & B(x) \wedge C(x) & \\
 & & B(x) & A(y) \quad D(x) & \\
 & \prod & \prod & \prod & \text{name} \\
\texttt{\textbackslash prfsummarystyle} = 1 & \forall x.\, x = x & A(x) & B(x) \wedge C(x) & \\
 & & B(x) & A(y) \quad D(x) & \\
 & \prod & \prod & \prod & \text{name} \\
\texttt{\textbackslash prfsummarystyle} = 2 & \forall x.\, x = x & A(x) & B(x) \wedge C(x) & \\
 & & B(x) & A(y) \quad D(x) & \\
 & \mathcal{D} & \mathcal{D} & \mathcal{D} & \text{name} \\
\texttt{\textbackslash prfsummarystyle} = 3 & \forall x.\, x = x & A(x) & B(x) \wedge C(x) & \\
 & & B(x) & A(y) \quad D(x) & \\
 & \bigtriangledown & \bigtriangledown & \bigtriangledown & \text{name} \\
\texttt{\textbackslash prfsummarystyle} = 4 & \forall x.\, x = x & A(x) & B(x) \wedge C(x) &
\end{array}$$

The fancy summary box is composed by the `\prffancysummarybox` command. This can be modified at will. The package defines it as

```
\newbox\prf@@fancysummarybox\newdimen\prf@@fancysymmarylen
\def\prffancysummarybox{%
  \sbox{\prf@@fancysummarybox}{\Huge$\bigtriangledown$}%
  \prf@@fancysymmarylen\ht\prf@@fancysummarybox%
  \advance\prf@@fancysymmarylen\dp\prf@@fancysummarybox%
  \sbox{\prf@@fancysummarybox}{%
    \raisebox{.25\prf@@fancysymmarylen}[.8\prf@@fancysymmarylen]%
    [0pt]{\usebox{\prf@@fancysummarybox}}}%
  \prf@@fancysymmarylen\wd\prf@summary@label%
  \ifdim\prf@@fancysymmarylen>\z@\relax%
```

```
    \prf@@fancysymmarylen\wd\prf@@fancysummarybox%
    \wd\prf@summary@label.4em%
    \hbox to\prf@@fancysymmarylen{%
      \usebox\prf@@fancysummarybox}\kern-.4em%
      \box\prf@summary@label%
  \else\usebox\prf@@fancysummarybox\fi}
```

The assumptions, conclusions, labels, and rule names are drawn using the following commands, which may be redefined:

```
\def\prfConclusionBox#1{%
  \hbox{$\displaystyle\begingroup#1\endgroup%
\def\prfAssumptionBox#1{%
  \hbox{$\displaystyle\begingroup#1\endgroup%
    \ifprfSTRUToption\mathstrut\fi$}}
\def\prfRuleNameBox#1{\hbox{\begingroup#1\endgroup%
    \ifprfSTRUTlabeloption\strut\fi}}
\def\prfLabelBox#1{\hbox{\begingroup#1\endgroup%
    \ifprfSTRUTlabeloption\strut\fi}}
```

It is not advisable to change these commands in a radical way, unless one understands how the graphical engine works.

4. Labels and References

As discharged assumptions are often hard to track in a proof, the package provides a mechanism to label them and to reference them inside a proof tree. A reference is made up of three pieces: the *label*, which is the name to denote the reference inside the text, the *reference value*, which is the value denoted by the label, and the *anchor*, which is the graphical rendering of the value aside the labelled point of the proof.

For example,

$$\dfrac{\dfrac{\dfrac{[A]^{1}\quad[\neg A]^{2}}{\bot}\supset\mathrm{E}}{\neg\neg A}\supset\mathrm{I}_{2}}{A\supset\neg\neg A}\supset\mathrm{I}_{1}$$

is generated by the following code

```
\begin{prfenv}
  \prftree[r]{$\supset\mathrm{I}_{\prfref<assum:A>}$}
  {\prftree[r]{$\supset\mathrm{I}_{\prfref<assum:not_A>}$}
    {\prftree[r]{$\supset$E}
      {\prfboundedassumption<assum:A>{A}}
      {\prfboundedassumption<assum:not_A>{\neg A}}
      {\bot}}
    {\neg\neg A}}
  {A \supset \neg\neg A}
\end{prfenv}
```

The labels are `assum:A` and `assum:not_A`, the reference values are 1 and 2, respectively, and the anchors are these values on the discharged assumptions on the top of the proof. The references to these labels are the values in the rule names.

The `prfenv` environment delimits the scope of labels: the `\end{prfenv}` declaration makes the labels still available for reference, but numbering of new labels will restart from 1. Enclosing a proof tree in a `prfenv` environment is not mandatory: in such case, labels will be global to the document.

Sometimes, labels require two compilation steps to be correctly generated: in fact, as LaTeX labels, forward references may be undefined in the first compilation step. The package issues a warning in this case, and display a `??` for the invalid reference. Also, notice how the assumption reference mechanism is analogous to LaTeX labels, but it is independent from it.

A reference to a label is made by the `\prfref`⟨label⟩ command: its argument is a label, i.e., a string of text following the same rules as the argument of the LaTeX `\label` command. As in the `\ref` command, the resulting value has no formatting.

A labelled assumption is generated by the following commands:

`\prfassumption`⟨[option]label⟩{assumption}
`\prfboundedassumption`⟨[option]label⟩{assumption}

The first one acts as `\prfassumption` but also declares the assumption label and decorates the assumption text with the anchor. The second one does the same on bounded assumptions.

The generation of labels is controlled by the option value:

- **n**, **number**, **arabic**: generates a number (default);
- **r**, **roman**: generates a lowercase roman number;
- **R**, **Roman**: generates an uppercase roman number;
- **a**, **alph**, **alpha**, **alphabetic**: produces a lowercase letter;
- **A**, **Alph**, **Alpha**, **Alphabetic**: produces an uppercase letter;
- **f**, **s**, **function**, **symbol**, **function symbol**: produces a footnote symbol, as in Section C.8.4 of Lamport's, *LaTeX: A document preparation system*;
- **l**, **label**: tells that the label has not to be defined. This is used to generate a labelled assumption sharing the label with another one, which declares the value and the format.

Except for **l** and **label**, all the options are used to format the anchor following the standard LaTeX way available for counters. No multiple options are allowed.

For example, the disjunction elimination rule is a perfect way to illustrate the reason behind the **label** option, i.e., the need to discharge a pair of assumptions:

$$\dfrac{\Gamma \quad \Gamma,[A]^{1} \quad \Gamma,[B]^{1} \\ \vdots \quad \vdots \quad \vdots \\ A\vee B \quad C \quad C}{C}\vee\mathrm{E}_{1}$$

```
\prftree[r]{$\vee\mathrm{E}_{\prfref<assum:orE>}$}
{\prfsummary{\Gamma}{A \vee B}}
{\prfsummary{\Gamma,
    \prfboundedassumption<assum:orE>{A}}{C}}
{\prfsummary{\Gamma,
    \prfboundedassumption<[l]assum:orE>{B}}{C}}{C}
```

If a label is declared more than once, a warning is issued when the **label** option is not used: although this is not a mistake, it may indicate that a label is reused when it should not.

The same example can be used to show how the other options work:

$$\dfrac{\Gamma \quad \Gamma,[A]^{1} \quad \Gamma,[B]^{1} \\ \prod \quad \prod \quad \prod \\ A\vee B \quad C \quad C}{C}\vee\mathrm{E}_{1} \qquad \dfrac{\Gamma \quad \Gamma,[A]^{i} \quad \Gamma,[B]^{i} \\ \prod \quad \prod \quad \prod \\ A\vee B \quad C \quad C}{C}\vee\mathrm{E}_{i} \qquad \dfrac{\Gamma \quad \Gamma,[A]^{I} \quad \Gamma,[B]^{I} \\ \prod \quad \prod \quad \prod \\ A\vee B \quad C \quad C}{C}\vee\mathrm{E}_{I}$$

$$\dfrac{\Gamma \quad \Gamma,[A]^{a} \quad \Gamma,[B]^{a} \\ \prod \quad \prod \quad \prod \\ A\vee B \quad C \quad C}{C}\vee\mathrm{E}_{a} \qquad \dfrac{\Gamma \quad \Gamma,[A]^{A} \quad \Gamma,[B]^{A} \\ \prod \quad \prod \quad \prod \\ A\vee B \quad C \quad C}{C}\vee\mathrm{E}_{A} \qquad \dfrac{\Gamma \quad \Gamma,[A]^{*} \quad \Gamma,[B]^{*} \\ \prod \quad \prod \quad \prod \\ A\vee B \quad C \quad C}{C}\vee\mathrm{E}_{*}$$

Also, as the `\prfboundedstyle` varies, the resulting proof trees are:

$$\dfrac{\Gamma \quad \Gamma,[A]^{1} \quad \Gamma,[B]^{1}}{\begin{array}{ccc} \bigtriangledown & \bigtriangledown & \bigtriangledown \\ A\vee B & C & C \end{array}}\vee\mathrm{E}_{1} \qquad \dfrac{\Gamma \quad \Gamma,\cancel{A}^{1} \quad \Gamma,\cancel{B}^{1}}{\begin{array}{ccc} \bigtriangledown & \bigtriangledown & \bigtriangledown \\ A\vee B & C & C \end{array}}\vee\mathrm{E}_{1} \qquad \dfrac{\Gamma \quad \Gamma,\langle A\rangle^{1} \quad \Gamma,\langle B\rangle^{1}}{\begin{array}{ccc} \bigtriangledown & \bigtriangledown & \bigtriangledown \\ A\vee B & C & C \end{array}}\vee\mathrm{E}_{1}$$

The `prfassumptioncounter` is the LaTeX counter used to generate the assumption values. It contains the last used value, and initially, it is set to 0. By modifying its value, e.g., to `\setcounter{prfassumptioncounter}{1}`,

$$\dfrac{\Gamma \quad \Gamma,[A]^{\dagger} \quad \Gamma,[B]^{\dagger}}{\begin{array}{ccc} \prod & \prod & \prod \\ A\vee B & C & C \end{array}}\vee\mathrm{E}_{\dagger}$$

A labelled assumption box is graphically constructed by the package command `\prflabelledassumptionbox` which can be redefined if needed. It takes two arguments: the assumption and the anchor. Its standard definition is

```
\def\prflabelledassumptionbox#1#2{%
  \setbox\prf@fancybox\hbox{${#1}$}%
  \prf@tmp\wd\prf@fancybox%
  \setbox\prf@fancybox\hbox{$\box\prf@fancybox^{#2}$}%
  \wd\prf@fancybox\prf@tmp%
  \prf@assumption{\box\prf@fancybox}}
```

Moreover, also a labelled and bounded assumption is graphically rendered by the same command. There is just one exception: when `\prfboundedstyle` > 1. In fact, since that style is controlled by a command that can be redefined, the same must hold for references in that style. The command which is called in this case is `\prflabelleddiscargedassumption` which can be redefined if needed; its standard definition in the package is

```
\def\prflabelleddiscargedassumption#1#2{%
  \prflabelledassumptionbox{\left\langle{#1}\right\rangle}{#2}}
```

Also proof summaries can be labelled and referenced. The syntax extends the `\prfsummary` command:

`\prfsummary`⟨[option]label⟩[name]{assumption1}$\cdots${assumption$_n$}{conclusion}

The reference argument works in the same way as the corresponding one for assumptions, and the options are the same.

$$\begin{array}{ccccc} \begin{array}{c} A\quad B \\ \vdots \\ {\scriptstyle 1} \end{array} & \begin{array}{c} A\quad B \\ \prod\nolimits_{2} \end{array} & \begin{array}{c} A\quad B \\ \prod\nolimits^{3} \end{array} & \begin{array}{c} A\quad B \\ \mathcal{D}_{4} \end{array} & \begin{array}{c} A\quad B \\ \bigtriangledown_{5} \end{array} \\ A\wedge B & A\wedge B & A\wedge B & A\wedge B & A\wedge B \end{array}$$

These examples have been generated by the following code snippet:

```
{\prfsummarystyle=X
 \prfsummary<proof:aX>{A}{B}{A \wedge B}}
```

The `[option]` part of the label specification is optional, and it works exactly as the option field of labelled assumptions. This is best illustrated by an example:

$$\dfrac{A\quad B}{A\wedge B}\,\mathrm{i}\qquad \dfrac{A\quad B}{A\wedge B}\,\mathrm{II}\qquad \dfrac{A\quad B}{A\wedge B}\,\ddagger\qquad \dfrac{A\quad B}{A\wedge B}\,\mathrm{d}$$

$$\dfrac{A\quad B}{A\wedge B}\,\mathrm{E}\qquad \dfrac{A\quad B}{A\wedge B}\,\ddagger$$

These examples have been generated by the following code snippet:

```
{\prfsummarystyle=1
 \prfsummary<[r]proof:bX>{A}{B}{A \wedge B}}
```

and the last line uses the `label` option.

The value of the summary labelling is controlled by the `prfsummarycounter` counter, which is initially 0 and contains the last used value.

5. Simplified Commands

The basic commands illustrated so far allow to control proof trees in all aspects, but they tend to be verbose in practise. Thus, a number of abbreviations are provided to make handier the writing of proofs. Since they may collide with other packages, these macros are activated by suitable options. Multiple options can be used at the same time.

5.1. **Natural deduction.** By loading the package with the `ND` option, the following abbreviations are available, which correspond to the inference rules of natural deduction calculi:

- `\NDA`: assumption;
- `\NDAL`: labelled assumption;
- `\NDD`: discharged assumption;
- `\NDDL`: labelled discharged assumption;
- `\NDP`: generic proof tree;
- `\NDAX`: a generic axiom rule;

$$\frac{}{x = x}\,\mathrm{ax} \quad ;$$

- `\NDANDI`: conjunction introduction

$$\frac{A \quad B}{A \wedge B}\,\wedge\mathrm{I} \quad ;$$

- `\NDANDER`, `\NDANDEL`, `\NDANDE`: conjunction elimination right, left, and unspecified, respectively

$$\frac{A \wedge B}{A}\,\wedge\mathrm{E}_1 \qquad \frac{A \wedge B}{B}\,\wedge\mathrm{E}_2 \quad ;$$

- `\NDORIR`, `\NDORIL`, `\NDORI`: disjunction introduction right, left, and unspecified, respectively

$$\frac{A}{A \vee B}\,\vee\mathrm{I}_1 \qquad \frac{B}{A \vee B}\,\vee\mathrm{I}_2 \quad ;$$

- `\NDOREL`, `\NDORE`: disjunction elimination, possibly labelled

$$\frac{A \vee B \quad \begin{array}{c}[A]^1\\ \vdots\\ C\end{array} \quad \begin{array}{c}[B]^1\\ \vdots\\ C\end{array}}{C}\,\vee\mathrm{E}^1 \qquad \frac{A \vee B \quad \begin{array}{c}A\\ \vdots\\ C\end{array} \quad \begin{array}{c}B\\ \vdots\\ C\end{array}}{C}\,\vee\mathrm{E} \quad ;$$

- `\NDIMPIL`, `\NDIMPI`: implication introduction, possibly labelled

$$\frac{\begin{array}{c}[A]^1\\ \vdots\\ B\end{array}}{A \to B}\,\to\mathrm{I}^1 \qquad \frac{\begin{array}{c}A\\ \vdots\\ B\end{array}}{A \to B}\,\to\mathrm{I} \quad ;$$

- `\NDIMPE`: implication elimination

$$\frac{A \to B \quad A}{B}\,\to\mathrm{E} \quad ;$$

- `\NDNOTIL`, `\NDNOTI`: negation introduction, possibly labelled

$$\dfrac{\begin{array}{c}[A]^1\\ \vdots\\ \bot\end{array}}{\neg A}\,\neg\mathrm{I}^1 \qquad \dfrac{\begin{array}{c}A\\ \vdots\\ \bot\end{array}}{\neg A}\,\neg\mathrm{I}\ ;$$

- `\NDNOTE`: negation elimination

$$\dfrac{\neg A \quad A}{\bot}\,\neg\mathrm{E}\ ;$$

- `\NDALLI`: universal quantifier introduction

$$\dfrac{A}{\forall x.\, A}\,\forall\mathrm{I}\ ;$$

- `\NDALLE`: universal quantifier elimination

$$\dfrac{\forall x.\, A}{A[t/x]}\,\forall\mathrm{E}\ ;$$

- `\NDEXI`: existential quantifier introduction

$$\dfrac{A[t/x]}{\exists x.\, A}\,\exists\mathrm{I}\ ;$$

- `\NDEXEL`, `\NDEXE`: existential quantifier elimination, possibly labelled

$$\dfrac{\exists x.\, A \quad \begin{array}{c}[A]^1\\ \vdots\\ B\end{array}}{B}\,\exists\mathrm{E}^1 \qquad \dfrac{\exists x.\, A \quad \begin{array}{c}A\\ \vdots\\ B\end{array}}{B}\,\exists\mathrm{E}\ ;$$

- `\NDTI`: truth introduction

$$\dfrac{}{\top}\,\top\mathrm{I}\ ;$$

- `\NDFE`: falsity elimination

$$\dfrac{\bot}{A}\,\bot\mathrm{E}\ ;$$

- `\NDLEM`: law of Excluded Middle

$$\dfrac{}{A \vee \neg A}\,\mathrm{lem}\ .$$

The labels, when present, are the first argument, the rest being the assumptions and, finally, the conclusion. The rules do not have a fixed format, so extensions are allowed, e.g., on conjunction elimination or disjunction introduction.

For example, the proof

$$\dfrac{\dfrac{}{A \vee \neg A}\,\mathrm{lem} \qquad \dfrac{\begin{array}{c}[A]^2\\ \vdots\\ \neg\neg A \supset A\end{array}}{\neg\neg A \supset A}\,{\supset}\mathrm{I} \qquad \dfrac{\dfrac{\dfrac{[\neg\neg A]^1 \quad [\neg A]^2}{\bot}\,{\supset}\mathrm{E}}{A}\,\bot\mathrm{E}}{\neg\neg A \supset A}\,{\supset}\mathrm{I}^1}{\neg\neg A \supset A}\,\vee\mathrm{E}^2$$

is typeset in abbreviated form by the following code

```
\NDOREL{simp:notA}{\NDLEM{A \vee \neg A}}
{\NDIMPI{\NDDL{[l]simp:notA}{A}}{\neg\neg A \supset A}}
{\NDIMPIL{simp:notnotA}
  {\NDFE{\NDIMPE{\NDDL{simp:notnotA}{\neg\neg A}}
      {\NDDL{simp:notA}{\neg A}}{\bot}}{A}}
  {\neg\neg A \supset A}}
{\neg\neg A \supset A}
```

5.2. **Sequents.** Similarly, by loading the package with the `SEQ` option, the following abbreviations are available, which roughly correspond to the inference rule of sequent calculi:

- `\SEQA`: assumption;
- `\SEQD`: bounded assumption (not normally used, but handy to have in case of fancy calculi);
- `\SEQP`: generic proof;
- `\SEQAX`: axiom rule

$$\frac{}{A \Rightarrow A}\,\mathrm{Ax}\ ;$$

- `\SEQLF`: left falsity

$$\frac{}{\bot \Rightarrow}\,\mathrm{L}\bot\ ;$$

- `\SEQLW`, `\SEQRW`: left and right weakening

$$\frac{\Gamma \Rightarrow \Delta}{A, \Gamma \Rightarrow \Delta}\,\mathrm{LW}\quad \frac{\Gamma \Rightarrow \Delta}{\Gamma \Rightarrow \Delta, A}\,\mathrm{LW}\ ;$$

- `\SEQLC`, `\SEQRC`: left and right contraction

$$\frac{A, A, \Gamma \Rightarrow \Delta}{A, \Gamma \Rightarrow \Delta}\,\mathrm{LC}\quad \frac{\Gamma \Rightarrow \Delta, A, A}{\Gamma \Rightarrow \Delta, A}\,\mathrm{RC}\ ;$$

- `\SEQLAND`, `\SEQLANDL`, `\SEQLANDR`: left conjunction; the `L` and `R` variants specify which side of the conjunction is introduced

$$\frac{A, \Gamma \Rightarrow \Delta}{A \wedge B, \Gamma \Rightarrow \Delta}\,\mathrm{L}\wedge_1\quad \frac{B, \Gamma \Rightarrow \Delta}{A \wedge B, \Gamma \Rightarrow \Delta}\,\mathrm{L}\wedge_2\ ;$$

- `\SEQRAND`: right conjunction

$$\frac{\Gamma \Rightarrow \Delta, A \quad \Gamma \Rightarrow \Delta, B}{\Gamma \Rightarrow \Delta, A \wedge B}\,\mathrm{R}\wedge\ ;$$

- `\SEQLOR`: left disjunction

$$\frac{A, \Gamma \Rightarrow \Delta \quad B, \Gamma \Rightarrow \Delta}{A \vee B, \Gamma \Rightarrow \Delta}\,\mathrm{L}\vee\ ;$$

- `\SEQROR`, `\SEQRORL`, `\SEQRORR`: right disjunction; the `R` and `L` variants specify which side of the disjunction is introduced

$$\frac{\Gamma \Rightarrow \Delta, A}{\Gamma \Rightarrow \Delta, A \vee B}\,\mathrm{R}\vee_1\quad \frac{\Gamma \Rightarrow \Delta, B}{\Gamma \Rightarrow \Delta, A \vee B}\,\mathrm{R}\vee_2\ ;$$

- `\SEQLIMP`: left implication

$$\frac{\Gamma \Rightarrow \Delta, A \quad B, \Gamma \Rightarrow \Delta}{A \to B, \Gamma \Rightarrow \Delta}\,\mathrm{L}{\to}\ ;$$

- `\SEQRIMP`: right implication

$$\frac{A, \Gamma \Rightarrow \Delta, B}{\Gamma \Rightarrow, \Delta, A \to B}\,\mathrm{R}{\to}\ ;$$

- `\SEQLALL`: left universal quantification

$$\frac{A[t/x], \Gamma \Rightarrow \Delta}{\forall x.\, A, \Gamma \Rightarrow \Delta}\,\mathrm{L}{\forall}\ ;$$

- `\SEQRALL`: right universal quantification

$$\frac{\Gamma \Rightarrow \Delta, A}{\Gamma \Rightarrow \Delta, \forall x.\, A}\,\mathrm{R}{\forall}\ ;$$

- `\SEQLEX`: left existential quantification

$$\frac{A, \Gamma \Rightarrow \Delta}{\exists x.\, A, \Gamma \Rightarrow \Delta}\,\mathrm{L}{\exists}\ ;$$

- `\SEQREX`: right existential quantification

$$\frac{\Gamma \Rightarrow \Delta, A[t/x]}{\Gamma \Rightarrow \Delta, \exists x.\, A}\,\mathrm{R}{\exists}\ ;$$

- `\SEQCUT`: cut rule

$$\frac{\Gamma \Rightarrow \Delta, A \quad A, \Gamma' \Rightarrow \Delta'}{\Gamma\Gamma' \Rightarrow \Delta\Delta'}\,\mathrm{Cut}\ .$$

5.3. **Equality.** Invoking the `EQ` option defines the following inference rules:

- `\EQREFL`: reflexivity

$$\frac{}{t = t}\,\mathrm{refl}\ ;$$

- `\EQSYM`: symmetry

$$\frac{t = s}{s = t}\,\mathrm{sym}\ ;$$

- `\EQTRANS`: transitivity

$$\frac{t = s \quad s = r}{t = r}\,\mathrm{trans}\ ;$$

- `\EQSUBST`: the substitution rule

$$\frac{t = s \quad A[t/x]}{A[s/x]}\,\mathrm{subst}\ .$$

5.4. **Implication.** Since the implication symbol is usually represented either as $\to$ or as $\supset$, the package allows to choose which representation to use. By default, implication is $\to$, but loading the package with the `[IMP]` option switches to $\supset$. The same effect is obtained by the commands `\prfIMPOptiontrue` (implication is $\supset$) and `prfIMPOptionfalse` (implication is $\to$).

5.5. **Martin-Löf Type Theory and Homotopy Type Theory.** Invoking the package with the `ML` option enables the support for these type theories. This part is derived from Roberta Bonacina's PhD dissertation, which used this package in an essential way to develop proof trees in Homotopy Type Theory.

Enabling the option `ML` defines a number of symbols which are useful to have. However, since they may conflict with other packages, they can be disabled invoking the option `MLnodef`. These operators are

- `\type`: the symbol : correctly spaced as a mathematical binary operation;
- `\universe`: the symbol for universes;
- `\judgementaldef` and `\propositionaldef`: the symbols $:\equiv$ and $:=$ spaced as mathematical binary operations;
- `\emptytype` ($\mathbf{0}$), `\unittype` ($\mathbf{1}$), `\booleantype` ($\mathbf{2}$): these symbols are ordinary operators typeset in mathematical boldface font;
- `\context` (ctx), `\identitytype` (Id), `\refl` (refl), `\axiomofchoice` (AC), `\accessibility` (acc), `\ap` (ap), `\apd` (apd), `\basepoint` (base), `\biinv` (biinv), `\cardtype` (Card), `\cocone` (cocone), `\cons` (cons), `\contr` (contr), `\equivtype` (Equiv), `\ext` (ext), `\fiber` (fib), `\funext` (funext), `\glue` (glue), `\happly` (happly), `\hom` (hom), `\id` (id), `\idtoeqv` (idtoeqv), `\im` (im), `\idtoiso` (idtoiso), `\ind` (ind), `\inj` (inj), `\inl` (inl), `\inr` (inr), `\iscontr` (isContr), `\isequiv` (isequiv), `\ishae` (ishae), `\isotoid` (istoid), `\isprop` (isProp), `\isset` (isSet), `\ker` (ker), `\LEM` (LEM), `\linv` (linv), `\listtype` (List), `\loopcons` (loop), `\Map` (Map), `\merid` (merid), `\nil` (nil), `\ordtype` (Ord), `\pair` (pair), `\pred` (pred), `\pr` (pr), `\Prop` (Prop), `\qinv` (qinv), `\rec` (rec), `\rinv` (rinv), `\seg` (seg), `\Set` (Set), `\Succ` (succ), `\sup` (sup), `\total` (total), `\transport` (transport), `\ua` (ua), `\Wtype` (W), `\transportconst` (transportconst): these symbols are ordinary operators, typeset in the mathematical sans-serif font; their graphical appearance is in brackets.

The large number of inference rules is listed below: they cover the structural part of the theories, plus most of the usual inductive types, comprehending also some higher-order inductive types. To each rule is associated a rule name, which is available as a command: the convention is that the rule name is obtained appending `rule` to the name of the inference rule. In general, the command to typeset a rule conforms to the standard name in the book *Homotopy Type Theory*. The name as typeset, is shown in brackets:

- `\MLctxEMP` (ctx−EMP),
  `\MLctxEXT` (ctx−EXT): context manipulation;
- `\MLVble` (Vble): variable introduction;
- `\MLSubst` (Subst), `\MLWkg` (Wkg): substitution and weakening;
- `\MLEQrefl` ($\equiv$−refl), `\MLEQsym` ($\equiv$−sym), `\MLEQtrans` ($\equiv$−trans),
  `\MLEQsubst` ($\equiv$−subst), `\MLEQsubsteq` ($\equiv$−subst−eq): structural rules about judgemental equality;
- `\MLUintro` ($\mathcal{U}$−intro), `\MLUcumul` ($\mathcal{U}$−cumul), `\MLUcumuleq` ($\mathcal{U}$−cumul−eq): type universe;
- `\MLpiform` (Π−form), `\MLpiformeq` (Π−form−eq),
  `\MLpiintro` (Π−intro), `\MLpiintroeq` (Π−intro−eq),
  `\MLpielim` (Π−elim), `\MLpielimeq` (Π−elim−eq),
  `\MLpicomp` (Π−comp), `\MLpiuniq` (Π−uniq): dependent function types;

- \MLKintro ($k-$intro): generic rule for constant introduction;
- \MLsigmaform ($\Sigma-$form), \MLsigmaintro ($\Sigma-$intro), \MLsigmaelim ($\Sigma-$elim), \MLsigmacomp ($\Sigma-$comp), \MLsigmauniq ($\Sigma-$uniq): dependent pair types;
- \MLplusform ($+-$form), \MLplusintrol ($+-\text{intro}_1$), \MLplusintror ($+-\text{intro}_2$), \MLpluselim ($+-$elim), \MLpluscompl ($+-\text{comp}_1$), \MLpluscompr ($+-\text{comp}_2$), \MLplusuniq ($+-$uniq): coproduct types;
- \MLzeroform ($\mathbf{0}-$form), \MLzeroelim ($\mathbf{0}-$elim), \MLzerouniq ($\mathbf{0}-$uniq): the empty type;
- \MLunitform ($\mathbf{1}-$form), \MLunitintro ($\mathbf{1}-$intro), \MLunitelim ($\mathbf{1}-$elim), \MLunitcomp ($\mathbf{1}-$comp), \MLunituniq ($\mathbf{1}-$uniq): the unit type;
- \MLnatform ($\mathbb{N}-$form), \MLnatintrozero ($\mathbb{N}-\text{intro}_1$), \MLnatintrosucc ($\mathbb{N}-\text{intro}_2$), \MLnatelim ($\mathbb{N}-$elim), \MLnatcompzero ($\mathbb{N}-\text{comp}_1$), \MLnatcompsucc ($\mathbb{N}-\text{comp}_2$), \MLnatuniq ($\mathbb{N}-$uniq): the natural number type;
- \MLidform ($=-$form), \MLidintro ($=-$intro), \MLidelim ($=-$elim), \MLidcomp ($=-$comp), \MLiduniq ($=-$uniq): identity types;
- \MLwform ($\mathsf{W}-$form), \MLwintro ($\mathsf{W}-$intro), \MLwelim ($\mathsf{W}-$elim), \MLwcomp ($\mathsf{W}-$comp), \MLwuniq ($\mathsf{W}-$uniq): W types;
- \MLListform ($\mathsf{List}-$form), \MLListintron ($\mathsf{List}-\text{intro}_1$), \MLListintroc ($\mathsf{List}-\text{intro}_2$), \MLListelim ($\mathsf{List}-$elim), \MLListcompn ($\mathsf{List}-\text{comp}_1$), \MLListcompc ($\mathsf{List}-\text{comp}_2$), \MLListuniq ($\mathsf{List}-$uniq): List types;
- \MLfunext ($\Pi-$ext): function extensionality;
- \MLuniv ($\mathcal{U}_i-$univ): univalence;
- \MLSform ($\mathbb{S}^1-$form), \MLSintro ($\mathbb{S}^1-$intro), \MLSelim ($\mathbb{S}^1-$elim), \MLScomp ($\mathbb{S}^1-$comp), \MLSuniq ($\mathbb{S}^1-$uniq), \MLSpeqintro ($\mathbb{S}^1-\text{intro}-=$), \MLSpeqcomp ($\mathbb{S}^1-\text{comp}-=$): the $\mathbb{S}^1$ circle type;
- \MLIform ($I-$form), \MLIintroa ($I-\text{intro}_1$), \MLIintrob ($I-\text{intro}_2$), \MLIelim ($I-$elim), \MLIcompa ($I-\text{comp}_1$), \MLIcompb ($I-\text{comp}_2$), \MLIuniq ($I-$uniq), \MLIpeqintro ($I-\text{intro}-=$), \MLIpeqcomp ($I-\text{comp}-=$): the interval type;
- \MLsigmaintroa ($\Sigma-\text{intro}_1$), \MLsigmaintrob ($\Sigma-\text{intro}_2$), \MLsigmacompa ($\Sigma-\text{comp}_1$), \MLsigmacompb ($\Sigma-\text{comp}_2$), \MLsigmapeqintro ($\Sigma-\text{intro}-=$), \MLsigmapeqcomp ($\Sigma-\text{comp}-=$): suspensions;
- \MLPOform ($\sqcup-$form), \MLPOintroa ($\sqcup-\text{intro}_1$), \MLPOintrob ($\sqcup-\text{intro}_2$), \MLPOelim ($\sqcup-$elim), \MLPOcompa ($\sqcup-\text{comp}_1$), \MLPOcompb ($\sqcup-\text{comp}_2$), \MLPOuniq ($\sqcup-$uniq), \MLPOpeqintro ($\sqcup-\text{intro}-=$), \MLPOpeqcomp ($\sqcup-\text{comp}-=$): pushouts;
- \MLTform ($||\cdot||-$form), \MLTintro ($||\cdot||-$intro), \MLTelim ($||\cdot||-$elim), \MLTcomp ($||\cdot||-$comp), \MLTuniq ($||\cdot||-$uniq), \MLTpeqintro ($||\cdot||-\text{intro}-=$), \MLTpeqcomp ($||\cdot||-\text{comp}-=$): truncations;
- \MLtorusform (T^2-form), \MLtorusintro (T^2-intro), \MLtoruselim (T^2-elim), \MLtoruscomp (T^2-comp), \MLtoruspeqintroa ($T^2-\text{intro}-=_\text{p}$), \MLtoruspeqintrob ($T^2-\text{intro}-=_\text{q}$), \MLtoruspeqintroc ($T^2-\text{intro}-=_\text{t}$), \MLtoruspeqcompa ($T^2-\text{comp}-=_\text{p}$), \MLtoruspeqcompb ($T^2-\text{comp}-=_\text{q}$), \MLtoruspeqcompc ($T^2-\text{comp}-=_\text{t}$): the torus type.

5.6. **Defining new inference rules.** Of course, the reader is encouraged to develop her own abbreviations starting from the provided ones. To this aim two

commands are provided. They share the same syntax: `\prfMakeInferenceRule` and `\prfMakeInferenceRuleRef` take two arguments, the first one is the name of the command associated to the inference rule, and the second one is used to write the rule name. For example,

```
\prfMakeInferenceRule{NDANDI}{\mathord{\wedge}\textup{I}}
```

is how the conjunction introduction rule is defined, and

```
\prfMakeInferenceRuleRef{NDOREL}{\mathord{\vee}\textup{E}}
```

is how the disjunction elimination rule is defined. The rules generated by the `Ref` variant use their first argument as the reference to the assumption(s) they discharge.

5.7. **Stacking proofs and assumptions.** Sometimes, a proof is too large to fit into the text width. Although some strategies could be implemented to compress it, see the next section, they fail in extreme cases. For example, the elimination rule for the circle in Homotopy type theories is:

$$\frac{\Gamma, x : \mathbb{S}^1 \vdash C : \mathcal{U}_i \quad \Gamma \vdash b : C[\mathsf{base}/x] \quad \Gamma \vdash \ell : b =^{C}_{\mathsf{loop}} b \quad \Gamma \vdash p : \mathbb{S}^1}{\Gamma \vdash \mathsf{ind}_{\mathbb{S}^1}(x.\, C, b, \ell, \mathsf{base}) : C[p/x]}\, \mathbb{S}^1\text{-}\mathsf{comp}$$

typeset by

```
\MLScomp
{\Gamma, x \type \mathbb{S}^1 \vdash C \type \universe_i}
{\Gamma \vdash b \type C[\basepoint/x]}
{\Gamma \vdash \ell \type b =_{\loopcons}^{C} b}
{\Gamma \vdash p \type \mathbb{S}^1}
{\Gamma \vdash \ind_{\mathbb{S}^1}(x.\, C, b, \ell, \basepoint)
  \type C[p/x]}
```

It is clear that on an A5 paper, there is not enough space to write it down. In these cases, the package provides a way to *stack* the premises of a rule, obtaining

$$\frac{\begin{array}{ll}\Gamma, x : \mathbb{S}^1 \vdash C : \mathcal{U}_i & \Gamma \vdash \ell : b =^{C}_{\mathsf{loop}} b \\ \Gamma \vdash b : C[\mathsf{base}/x] & \Gamma \vdash p : \mathbb{S}^1\end{array}}{\Gamma \vdash \mathsf{ind}_{\mathbb{S}^1}(x.\, C, b, \ell, \mathsf{base}) : C[p/x]}\, \mathbb{S}^1\text{-}\mathsf{comp}$$

The corresponding LaTeX code is

```
\MLScomp
{\prfStackPremises
  {\Gamma, x \type \mathbb{S}^1 \vdash C \type \universe_i}
  {\Gamma \vdash b \type C[\basepoint/x]}
}
{\prfStackPremises
  {\Gamma \vdash \ell \type b =_{\loopcons}^{C} b}
  {\Gamma \vdash p \type \mathbb{S}^1}
}
{\Gamma \vdash
  \ind_{\mathbb{S}^1}(x.\, C, b, \ell, \basepoint) \type C[p/x]}
```

The command `\prfStackPremises{`a_1`}{...}{`a_n`}` takes the arguments $a_1, \ldots, a_n$ and typeset them as a proof tree with no lines with a_1 on the top.

Actually, stacking proofs is possible:

$$\dfrac{\begin{array}{c}\Gamma\ \mathsf{ctx}\\ \vdots\\ \Gamma, x : \mathbb{S}^1 \vdash \mathbb{S}^1 : \mathcal{U}_i\end{array}\quad \begin{array}{c}\Gamma\ \mathsf{ctx}\\ \vdots\\ \Gamma \vdash \ell : \mathsf{base} = \mathsf{base}\end{array}}{\dfrac{\begin{array}{c}\Gamma\ \mathsf{ctx}\\ \vdots\\ \Gamma \vdash \mathsf{base} : \mathbb{S}^1\end{array}\qquad \begin{array}{c}\Gamma\ \mathsf{ctx}\\ \vdots\\ \Gamma \vdash p : \mathbb{S}^1\end{array}}{\Gamma \vdash \mathsf{ind}_{\mathbb{S}^1}(x.\, C, b, \ell, \mathsf{base}) : C[p/x]}\, \mathbb{S}^1\text{-}\mathsf{comp}}$$

has been typeset by

```
\MLScomp
{\prfStackPremises
  {\prfsummary{\Gamma\;\context}
    {\Gamma, x \type \mathbb{S}^1 \vdash \mathbb{S}^1 \type
      \universe_i}}
  {\prfsummary{\Gamma\;\context}
    {\Gamma \vdash \basepoint \type \mathbb{S}^1}}
}
{\prfStackPremises
  {\prfsummary{\Gamma\;\context}
    {\Gamma \vdash \ell \type \basepoint = \basepoint}}
  {\prfsummary{\Gamma\;\context}
    {\Gamma \vdash p \type \mathbb{S}^1}}
}
{\Gamma \vdash
   \ind_{\mathbb{S}^1}(x.\, C, b, \ell, \basepoint) \type C[p/x]}
```

Since a stack is a proof tree, the parameters could be locally changed to control its appearance. For example

$$\dfrac{\begin{array}{c}\Gamma, x : \mathbb{S}^1 \vdash C : \mathcal{U}_i\\ \\ \Gamma \vdash b : C[\mathsf{base}/x]\end{array}\quad \begin{array}{c}\\ \Gamma \vdash \ell : b =^{C}_{\mathsf{loop}} b\\ \\ \Gamma \vdash p : \mathbb{S}^1\end{array}}{\Gamma \vdash \mathsf{ind}_{\mathbb{S}^1}(x.\, C, b, \ell, \mathsf{base}) : C[p/x]}\, \mathbb{S}^1\text{-}\mathsf{comp}$$

makes the lines in the left stack far apart.

```
\MLScomp
{\prfemptylinethickness20\prflinethickness
  \prfStackPremises
  {\Gamma, x \type \mathbb{S}^1 \vdash C \type \universe_i}
  {\Gamma \vdash b \type C[\basepoint/x]} }
{\prfStackPremises
  {\Gamma \vdash \ell \type b =_{\loopcons}^{C} b}
  {\Gamma \vdash p \type \mathbb{S}^1} }
{\Gamma \vdash \ind_{\mathbb{S}^1}(x.\, C, b, \ell, \basepoint)
  \type C[p/x]}
```

Spacing in stacks of proofs is normally difficult to control: if really sophisticated formatting is needed, it is better to consider the following option:

$$\dfrac{\begin{array}{c}\Gamma\ \mathsf{ctx}\\ \vdots\\ \Gamma, x:\mathbb{S}^1 \vdash \mathbb{S}^1 : \mathcal{U}_i\end{array}\quad \begin{array}{c}\Gamma\ \mathsf{ctx}\\ \vdots\\ \Gamma \vdash \ell : \mathsf{base} = \mathsf{base}\end{array}\qquad \begin{array}{cc}\Gamma \vdash \mathsf{base} : \mathbb{S}^1 & \Gamma \vdash p : \mathbb{S}^1\end{array}}{\Gamma \vdash \mathsf{ind}_{\mathbb{S}^1}(x.\, C, b, \ell, \mathsf{base}) : C[p/x]}\ \mathbb{S}^1\mathsf{-comp}$$

which uses the `array` environment

```
\MLScomp
{\prfassumption{
    \begin{array}{@{}c@{\quad}c@{}}
      {\prfsummary{\Gamma\;\context}
        {\Gamma, x \type \mathbb{S}^1 \vdash \mathbb{S}^1 \type
          \universe_i}} &
      {\Gamma \vdash \ell \type \basepoint = \basepoint} \\
      {\prfsummary{\Gamma\;\context}
        {\Gamma \vdash \basepoint \type \mathbb{S}^1}} &
      {\Gamma \vdash p \type \mathbb{S}^1}
    \end{array}}}
{\Gamma \vdash \ind_{\mathbb{S}^1}(x.\, C, b, \ell, \basepoint)
  \type C[p/x]}
```

or similar ones, using the multitude of packages to format tables. By the way, the obvious solution using stacks is

$$\dfrac{\begin{array}{c}\Gamma\ \mathsf{ctx}\\ \vdots\\ \Gamma, x:\mathbb{S}^1 \vdash \mathbb{S}^1 : \mathcal{U}_i\end{array}\quad \begin{array}{c}\Gamma\ \mathsf{ctx}\\ \vdots\\ \Gamma \vdash \ell : \mathsf{base} = \mathsf{base}\end{array}\qquad \begin{array}{cc}\Gamma \vdash \mathsf{base} : \mathbb{S}^1 & \Gamma \vdash p : \mathbb{S}^1\end{array}}{\Gamma \vdash \mathsf{ind}_{\mathbb{S}^1}(x.\, C, b, \ell, \mathsf{base}) : C[p/x]}\ \mathbb{S}^1\mathsf{-comp}$$

```
\MLScomp
{\prfStackPremises{\prfsummary{\Gamma\;\context}
    {\Gamma, x \type \mathbb{S}^1 \vdash \mathbb{S}^1 \type
      \universe_i}}
  {\prfsummary{\Gamma\;\context}
    {\Gamma \vdash \basepoint \type \mathbb{S}^1}} }
{\prfStackPremises{\prfassumption
    {\Gamma \vdash \ell \type \basepoint = \basepoint}}
  {\prfassumption
    {\Gamma \vdash p \type \mathbb{S}^1}} }
{\Gamma \vdash \ind_{\mathbb{S}^1}(x.\, C, b, \ell, \basepoint)
  \type C[p/x]}
```

6. Hints and Tricks

This section shows a few hints and tricks to use the package at its best.

Consider the proof:

$$\dfrac{\dfrac{}{A \vee \neg A}\,\text{lem} \quad \dfrac{\dfrac{[A]^1}{\neg\neg A \supset A}\,{\to}\text{I} \qquad \dfrac{\dfrac{\dfrac{[\neg\neg A]^2 \quad [\neg A]^1}{\bot}\,{\to}\text{E}}{A}\,\bot\text{E}}{\neg\neg A \supset A}\,{\to}\text{I}^2}{\neg\neg A \supset A}\,\vee\text{E}^1$$

the space between the axiom and the sub-proof of the positive case is visually much less than the space between the positive and the negative cases. Looking at boxes, the space is exactly the same, but the perception is that spacing is wrong.

We can correct this perception in two distinct ways: by adding space between the axiom and the positive case; or, conversely, by moving the negative case closer to the positive one.

The first strategy yields:

$$\dfrac{\dfrac{}{A \vee \neg A}\,\text{lem} \quad \dfrac{\dfrac{[A]^1}{\neg\neg A \supset A}\,{\to}\text{I} \qquad \dfrac{\dfrac{\dfrac{[\neg\neg A]^2 \quad [\neg A]^1}{\bot}\,{\to}\text{E}}{A}\,\bot\text{E}}{\neg\neg A \supset A}\,{\to}\text{I}^2}{\neg\neg A \supset A}\,\vee\text{E}^1$$

and this effect is given by adding an appropriate `\hspace` after the axiom, as in

```
\NDOREL{a:notA}{\NDLEM{A \vee \neg A}\hspace{.4em}}
{\NDIMPI{\NDDL{[l]a:notA}{A}}{\neg\neg A \supset A}}
{\NDIMPIL{a:notnotA}
  {\NDFE{\NDIMPE{\NDDL{[l]a:notnotA}{\neg\neg A}}
      {\NDDL{[l]a:notA}{\neg A}}{\bot}}{A}}
  {\neg\neg A \supset A}}
{\neg\neg A \supset A}
```

Adding the same space in front of the positive case is equivalent.

The second strategy yields:

$$\dfrac{\dfrac{}{A \vee \neg A}\,\text{lem} \quad \dfrac{\dfrac{[A]^1}{\neg\neg A \supset A}\,{\to}\text{I} \qquad \dfrac{\dfrac{\dfrac{[\neg\neg A]^2 \quad [\neg A]^1}{\bot}\,{\to}\text{E}}{A}\,\bot\text{E}}{\neg\neg A \supset A}\,{\to}\text{I}^2}{\neg\neg A \supset A}\,\vee\text{E}^1$$

Again, this is obtained by adding a negative `hspace` after the positive case, or, equivalently, before the negative one:

```
\NDOREL{a:notA}{\NDLEM{A \vee \neg A}}
{\NDIMPI{\NDDL{[l]a:notA}{A}}{\neg\neg A \supset A}}
{\hspace{-.8em}\NDIMPIL{a:notnotA}
  {\NDFE{\NDIMPE{\NDDL{[l]a:notnotA}{\neg\neg A}}
      {\NDDL{[l]a:notA}{\neg A}}{\bot}}{A}}
```

```
  {\neg\neg A \supset A}}
{\neg\neg A \supset A}
```

In general, to make a wide proof *compact*, one can appropriately add negative spaces in front of sub-proofs so to make them closer and letting them to overlap as boxes, but not visually, thus *tiling* the space.

Since proof trees are boxes, it is easy to align them on need. For example the following proof tree, with the bounding box put in evidence

$$\boxed{\dfrac{A \quad B}{A \wedge B}\,\wedge}$$

can be used wherever a box may appear. In the flow of text, it will look like $\boxed{\dfrac{A \quad B}{A \wedge B}\,\wedge}$, so that the conclusion is aligned with the baseline. This makes easier to align proof trees, as in

$$\boxed{\dfrac{f \quad g}{f \wedge g}\,\wedge} \qquad \boxed{\dfrac{\dfrac{}{A \vee \neg A}\,\text{lem} \quad \dfrac{[A]^1}{\neg\neg A \supset A}\,{\supset}\text{I} \quad \dfrac{\dfrac{[\neg\neg A]^2 \quad [\neg A]^1}{\bot}\,{\to}\text{E}}{A}\,\bot\text{E}}{\dfrac{}{\neg\neg A \supset A}\,{\supset}\text{I}^2}\,\vee\text{E}^1}$$

since this is the natural way to put proofs side by side:

```
\fbox{\prfsummarystyle=1
  \prfsummary{f}{g}{f \wedge g}}\qquad
\fbox{$
\NDOREL{a:notA}{\NDLEM{A \vee \neg A}}
{\NDIMPI{\NDDL{[l]a:notA}{A}}{\neg\neg A \supset A}}
{\hspace{-.4em}\NDIMPIL{a:notnotA}
  {\NDFE{\NDIMPE{\NDDL{[l]a:notnotA}{\neg\neg A}}
      {\NDDL{[l]a:notA}{\neg A}}{\bot}}{A}}
  {\neg\neg A \supset A}}
{\neg\neg A \supset A}$}
```

But, if really one has to include a proof tree in the flow of text, it is slightly better to vertically centre the box, as in $\boxed{\dfrac{A \quad B}{\vdots}\,\wedge}{\scriptstyle A \wedge B}$. This is obtained by

```
$\vcenter{\prfsummary{A}{B}{A \wedge B}}$
```

Of course, the result is not pleasant, because rows are far apart, which is unavoidable because of the height of the proof tree.

The same principle applies also to arrays of proof trees:

some text $\dfrac{A \quad B}{A \wedge B}\,\wedge_{\text{i}}$ $\dfrac{A \quad B}{A \wedge B}\,\wedge_{\text{II}}$ $\dfrac{A \quad B}{A \wedge B}\,\wedge_{\ddagger}$ $\dfrac{A \quad B}{A \wedge B}\,\wedge_{\text{d}}$

which has been typeset by

```
\begin{array}{lcccc}
  \text{some text} &
  {\prfsummarystyle=1
  \prfsummary<[l]proof:b1>{A}{B}{A \wedge B}} &
  {\prfsummarystyle=1
  \prfsummary<[l]proof:b2>{A}{B}{A \wedge B}} &
  {\prfsummarystyle=1
  \prfsummary<[l]proof:b3>{A}{B}{A \wedge B}} &
  {\prfsummarystyle=1
  \prfsummary<[l]proof:b4>{A}{B}{A \wedge B}}
\end{array}
```

vertically aligns the cells to their baselines.

On the contrary

$$\text{some text}\quad \overset{A\quad B}{\underset{A\wedge B}{\prod_{\mathrm{i}}}}\quad \overset{A\quad B}{\underset{A\wedge B}{\prod_{\mathrm{II}}}}\quad \overset{A\quad B}{\underset{A\wedge B}{\prod_{\ddagger}}}\quad \overset{A\quad B}{\underset{A\wedge B}{\prod_{\mathrm{d}}}}$$

is much better, and it is obtained by

```
\begin{array}{lcccc}
  \text{some text} &
  \vcenter{\prfsummarystyle=1
  \prfsummary<[l]proof:b1>{A}{B}{A \wedge B}} &
  \vcenter{\prfsummarystyle=1
  \prfsummary<[l]proof:b2>{A}{B}{A \wedge B}} &
  \vcenter{\prfsummarystyle=1
  \prfsummary<[l]proof:b3>{A}{B}{A \wedge B}} &
  \vcenter{\prfsummarystyle=1
  \prfsummary<[l]proof:b4>{A}{B}{A \wedge B}}
\end{array}
```

The labelling of proof summaries is useful when a proof is very large and there is the need to split it. The strategy is to select some sub-proofs and to show them as summaries: instead of writing

$$\dfrac{\dfrac{}{A\vee\neg A}\,\mathrm{lem}\quad \dfrac{[A]^1}{\neg\neg A\supset A}\,{\to}\mathrm{I}\quad \dfrac{\dfrac{\dfrac{[\neg\neg A]^2\quad [\neg A]^1}{\bot}\,{\to}\mathrm{E}}{A}\,\bot\mathrm{E}}{\neg\neg A\supset A}\,{\to}\mathrm{I}^2}{\neg\neg A\supset A}\,\vee\mathrm{E}^1$$

we may consider to define

$$\text{Let}\ \begin{pmatrix}[\neg\neg A]^* & \neg A^\dagger\\ \vdots & \\ \neg\neg A & \end{pmatrix}^* \supset A \equiv \begin{pmatrix}\dfrac{\dfrac{[\neg\neg A]^* \quad \neg A^\dagger}{\bot}\,{\to}\mathrm{E}}{A}\,\bot\mathrm{E}\\ \dfrac{}{\neg\neg A\supset A}\,{\to}\mathrm{I}^*\end{pmatrix}$$

allowing to abbreviate the whole proof as

$$\dfrac{\dfrac{}{A \vee \neg A}\,\text{lem} \quad \dfrac{\dfrac{[A]^{\dagger}}{\neg\neg A \supset A}\,{\supset}\text{I} \quad \begin{array}{c}[\neg\neg A]^{*}\\ \vdots\\ \neg\neg A \supset A\end{array}}{\neg\neg A \supset A}\,{\vee}\text{E}^{\dagger}}{}\,{}^{*}$$

The corresponding LaTeX code is

```
\setcounter{prfsummarycounter}{0}
\setcounter{prfassumptioncounter}{0}
\mbox{Let }
\left(\vcenter{\prfsummary<[f]s:abbrev>
    {\NDDL{s:notnotA}{\neg\neg A}}
    {\NDAL{s:notA}{\neg A}}
    {\neg\neg A \supset A}}\right)
\equiv
\left(\vcenter{\NDIMPIL{s:notnotA}
    {\NDFE{\NDIMPE{\NDDL{[l]s:notnotA}{\neg\neg A}}
        {\NDAL{[l]s:notA}{\neg A}}{\bot}}{A}}
    {\neg\neg A \supset A}}\right)
```

for the definition of the proof summary, and

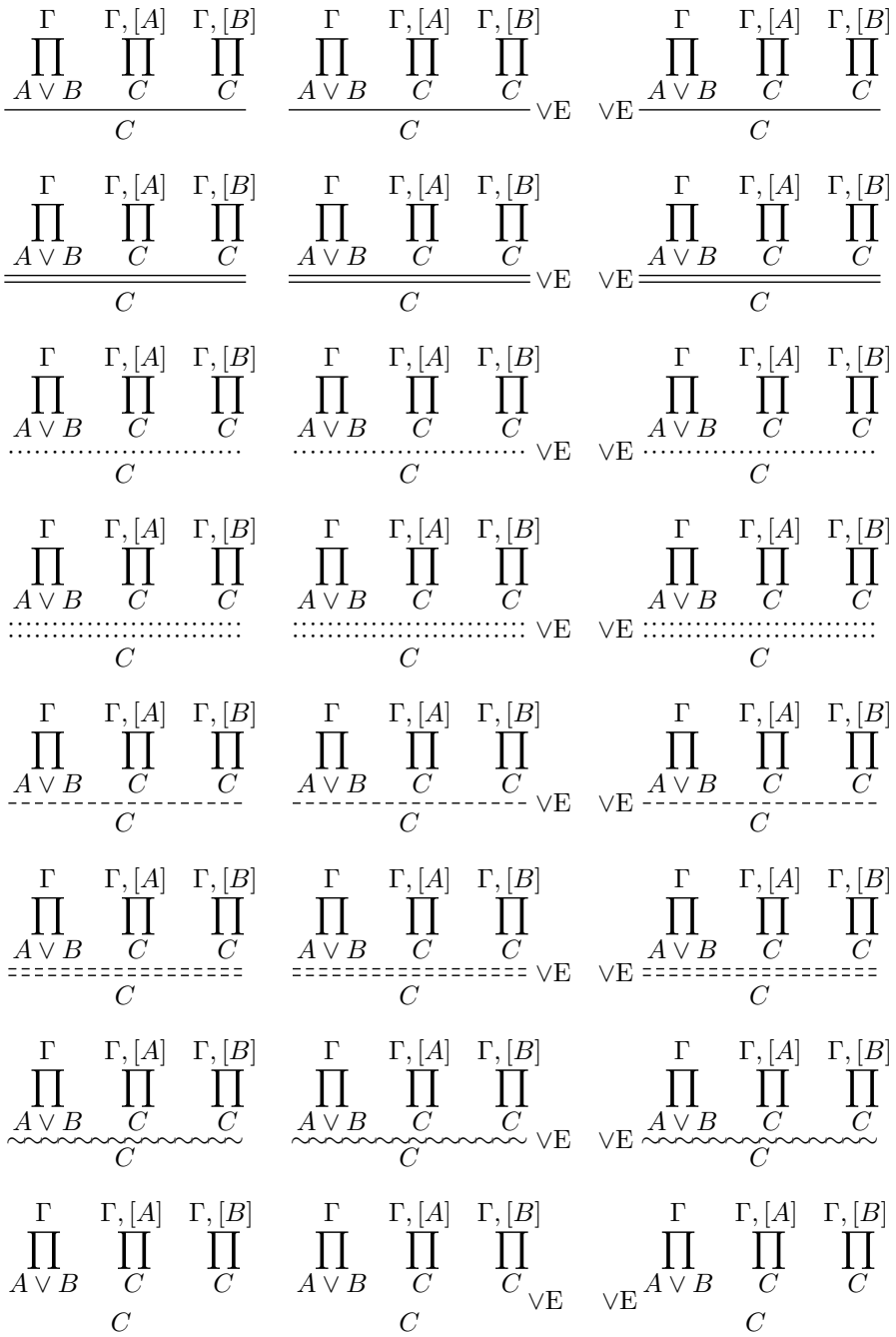
```
\NDOREL{s:notA}{\NDLEM{A \vee \neg A}}
{\NDIMPI{\NDDL{[l]s:notA}{A}}{\neg\neg A \supset A}}
{\hspace{-1em}\prfsummary<s:abbrev>
  {\NDDL{[l]s:notnotA}{\neg\neg A}}
  {\NDDL{[l]s:notA}{\neg A}}
    {\neg\neg A \supset A}}
{\neg\neg A \supset A}
```

for its use.

7. More Examples

This section shows a number of examples illustrating the package. See the previous sections for the description of the features.

The disjunction elimination rule, with various line options:



Proof that the Law of Excluded middle implies $\neg\neg A \supset A$:

$$\dfrac{\dfrac{}{A \vee \neg A}\,\text{lem} \qquad \dfrac{\dfrac{[A]}{\neg\neg A \supset A}\,{\to}\text{I} \qquad \dfrac{\dfrac{\dfrac{[\neg\neg A] \quad [\neg A]}{\bot}\,{\to}\text{E}}{A}\,\bot\text{E}}{\neg\neg A \supset A}\,{\to}\text{I}}{\neg\neg A \supset A}\,\vee\text{E}$$

Proof that the Law of Excluded middle implies $\neg\neg A \supset A$ with labels instead of rule names, except on axioms:

$$\dfrac{\dfrac{}{A \vee \neg A}\,\text{lem} \qquad \supset\text{I}\,\dfrac{[A]}{\neg\neg A \supset A} \qquad \supset\text{I}\,\dfrac{\bot\text{E}\,\dfrac{\supset\text{E}\,\dfrac{[\neg\neg A] \quad [\neg A]}{\bot}}{A}}{\neg\neg A \supset A}}{\neg\neg A \supset A}\,\vee\text{E}$$

Another simple proof in natural deduction:

$$\dfrac{\dfrac{\dfrac{\dfrac{[A \to (B \to C)] \quad [A]}{B \to C} \qquad \dfrac{[A \to B] \quad [A]}{B}}{C}}{A \to C}}{(A \to B) \to (A \to C)}}{(A \to (B \to C)) \to ((A \to B) \to (A \to C))}$$

The same proof, under the proposition-as-types interpretation:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{u\colon A \to (B \to C) \quad w\colon A}{uw\colon B \to C} \qquad \dfrac{v\colon A \to B \quad w\colon A}{vw\colon B}}{uw(vw)\colon C}}{\lambda w.\, uw(vw)\colon A \to C}}{\lambda vw.\, uw(vw)\colon (A \to B) \to (A \to C)}}{\lambda uvw.\, uw(vw)\colon (A \to (B \to C)) \to ((A \to B) \to (A \to C))}$$

A deduction in a sequent calculus:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{A \Rightarrow A} \qquad \dfrac{\dfrac{}{B \Rightarrow B} \quad \dfrac{}{C \Rightarrow C}}{B, B \to C \Rightarrow C}}{A, A \to B, B \to C \Rightarrow C}}{A, A \to B, A \to (B \to C) \Rightarrow C}}{A \to B, A \to (B \to C) \Rightarrow A \to C}}{A \to (B \to C) \Rightarrow (A \to B) \to (A \to C)}}{\Rightarrow (A \to (B \to C)) \to ((A \to B) \to (A \to C))}$$

Proof trees can be coloured, as kindly pointed out by Dominic Hughes:

$$\dfrac{\dfrac{\dfrac{[\neg(A\vee B)]^{1}\quad\dfrac{[A]^{2}}{A\vee B}\vee\mathrm{I}_1}{\bot}\neg\mathrm{E}}{\neg A}\neg\mathrm{I}^{2}\qquad\dfrac{\dfrac{[\neg(A\vee B)]^{1}\quad\dfrac{[B]^{3}}{A\vee B}\vee\mathrm{I}_2}{\bot}\neg\mathrm{E}}{\neg B}\neg\mathrm{I}^{3}}{\neg A\wedge\neg B}\wedge\mathrm{I}}{\neg(A\vee B)\supset\neg A\wedge\neg B}\to\mathrm{I}^{1}$$

Also. all the standard box manipulation commands can be freely applied. The following examples are not significant for doing mathematics, but the mechanics behind can be occasionally useful, for example, to shrink a large proof to fit the page length:

$$\dfrac{\dfrac{\dfrac{[\neg(A\vee B)]^{1}\quad\dfrac{[A]^{2}}{A\vee B}\vee\mathrm{I}_1}{\bot}\neg\mathrm{E}}{\neg A}\neg\mathrm{I}^{2}\qquad\dfrac{\dfrac{[\neg(A\vee B)]^{1}\quad\dfrac{[B]^{3}}{A\vee B}\vee\mathrm{I}_2}{\bot}\neg\mathrm{E}}{\neg B}\neg\mathrm{I}^{3}}{\neg A\wedge\neg B}\wedge\mathrm{I}}{\neg(A\vee B)\supset\neg A\wedge\neg B}\to\mathrm{I}^{1}$$

$$\dfrac{\dfrac{\dfrac{[\neg(A\vee B)]^{1}\quad\dfrac{[B]^{3}}{A\vee B}\vee\mathrm{I}_2}{\bot}\neg\mathrm{E}}{\neg B}\neg\mathrm{I}^{3}}{\neg A\wedge\neg B}\wedge\mathrm{I}}{\neg(A\vee B)\supset\neg A\wedge\neg B}\to\mathrm{I}^{1}$$

8. Fonts

The package works with any font. It uses the current math fonts for typesetting proofs, while it uses the current text font to typeset labels and rule names.

Care has been taken to ensure that the various dimensions and parameters in Section 3 are relative to the current font, that is, technically, they are expressed with units `ex` for vertical lengths, and `em` for horizontal lengths. Dashes are TEX rules with thickness `\prflinethickness`.

For unknown reasons, the `fontenc` package modifies slightly the values for `ex` and `em`, thus the graphical appearance of proof trees may vary when comparing the results obtained by compiling with and without this package.

In most cases, the graphical appearance of proofs is acceptable, even changing font and size. But using fonts whose body is particularly heavy, may result in proof lines which are too thin. In this case, the user of the package should increment the value of `\prflinethickness`.

The package, up to version 1.5, was designed to work with the Computer Modern family of fonts. It worked also with other fonts, without any major problem, but, as kindly signalled by Démi Nollet at ENS Lyon and université Paris-Diderot, dashed and dotted lines do not behave correctly, as dashes overlap. Please, update to the latest version of the package if you plan to use fonts different from Computer Modern.

9. Internals

A proof tree is typeset as a TeX box in horizontal mode. This means that wherever a character can stay, so does a proof: in principle, there is no need to put the proof in a math environment. Also, the width of a proof is exactly the width of the box; the height of the proof is the height of the conclusion plus the total height of all the matter above it; the depth of the proof is the depth of the conclusion. The proof is aligned so that the current baseline is the baseline of the conclusion.

For example, the proof of $g \supset \neg\neg g$ in natural deduction is:

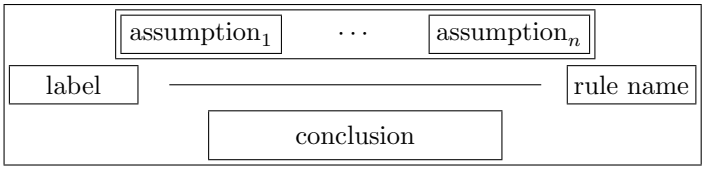
$$\text{proof} \equiv \boxed{\dfrac{\dfrac{\dfrac{[g] \quad [\neg g]}{\bot}\supset\text{E}}{\neg\neg g}\supset\text{I}}{g \supset \neg\neg g}\supset\text{I}}$$

The proof has been surrounded by a framebox to make evident its bounds. Also, since the letter g has a depth, the example shows how depth in the conclusion influences the alignment of the proof with respect to the preceding text.

Actually, the fundamental command in the package is `\prftree`: the commands to construct assumptions (`\prfassumption` and `\prfboundedassumption`), those to generate axioms (`\prfaxiom` and `\prfbyaxiom`), and `\prfsummary` are just appropriate instances.

The `\prftree` command is composed by a parser, which takes care of reading the various options and parameters, and by a graphical engine, `\prf@draw`, which calculates and draw the box containing the proof tree.

It may be useful to understand how the graphical engine works. In the first place, each proof tree is a box with a structure:



The conclusion, the proof line, and the *assumption line* are centred. The assumption line is the line whose first element is the conclusion of the first assumption, and whose last element is the conclusion of the last assumption, properly spaced so that all the assumptions fit in between. The width of the proof line is calculated as the maximum of the width of the assumption line and the conclusion, with the rule name and the label, if present, hanging on the right and the left, respectively.

To calculate the assumption line, the engine keeps track of the position of the conclusion within a proof tree, which reduces to remember how far is the conclusion from the left margin (`Lassum`), and how far it is from the right margin (`Rassum`). So, the assumption line starts from the value of `Lassum` of the first assumption, and finishes at `Rassum` of the last assumption.

Thus, with these values it is not difficult to figure out the mathematics to place the various boxes around, so to combine them into a proof tree. This is exactly what the graphical engine does.

Unfortunately, when one writes assumptions as simple formulae, without the `\prfassumption` command, the corresponding `Lassum` and `Rassum` are not set to 0, which is the right value. In fact, the recursive expansion of the `\prf@draw` macro follows the *natural* order in the construction of the proof box, which is extremely useful because it allows to locally modify parameters in sub-proofs; but this order conflicts with proper rendering of assumptions which are not proof trees.

Also, the hints on how to put space between assumptions, see Section 6, may have strange effects: if space is added in front of the first assumption or behind the last one, this space makes invalid the values of `Lassum` and `Rassum`, respectively, yielding hard to predict results.

It is worth remarking that the mathematics of the graphical engine is sound, which means that zero or negative values for the various dimensions specified as parameters, or using *bizarre* boxes in the fancy commands, yields the expected results, as far as boxes do not have parts which extends beyond the bounds.

The implementation of references mimics the implementation of `\label` and `\ref` in LaTeX. Whenever a reference is defined, through a command with the ⟨label⟩ as the first argument, the reference value is created according to the options, and it gets stored in the `.aux` file, by writing `\prfauxvalue`{label}{value} in the file. Then, when the source code will be recompiled, and the `.aux` file read, this command will be executed before any occurrence of a reference, which can be resolved.

Most difficulties in the implementation of references lie in the way to construct the boxes to be used in the proof tree. But, the tricky part is the interaction with the LaTeX and TeX kernel for error reporting. A small hack has been introduced to force recompilation when the references in a proof change.

10. Future Features and Bugs

Essentially, all the features of Buss's package have been implemented but one: alignment of proofs according to the $\vdash$ (or equivalent) sign. While this feature is occasionally useful in the writing of sequent proofs, it requires some trickery in the graphical engine, so it has been postponed for the moment.

Moreover, automatic compact proofs have been analysed, but not implemented. A compact proof minimises the amount of space between subsequent assumptions, eventually making the upper trees to overlap as boxes, but not as typed text.

The algorithm to obtain this result is not immediate: one should keep track of the left and right *skylines* of a proof. Comparing the left skyline of an assumption with the right skyline of the next one, one can calculate what is the distance between the boxes so that the distance between the closest points in the skylines is exactly `\prfinterspace`.

It is not simple to code such an algorithm in TeX, but the real difficulty is how to represent skylines and how to store them, since TeX provides no abstract data structures. Hence, the implementation of this feature has been postponed to a remote future, or to the will of a real TeX magician.

The abbreviated commands reflect their use by the author. It is quite possible that you want to define your own commands for inference rules of your interest. If you think they could be of general interest, send them by email to the author (see below) who will include them in a future release of the package, acknowledging your contribution.

Although the package has been tested for a long time by now, it is possible that a few bugs are still present. To signal a bug, please, write an email to the author (see below), possibly attaching a sample document which exhibit the misbehaviour, to help tracking and fixing.

Dipartimento di Scienza e Alta Tecnologia, Università degli Studi dell'Insubria, via Valleggio 11, I-22100 Como, Italy

Email address: `marco.benini@uninsubria.it`

URL: `http://marcobenini.wordpress.com`