

MATH formulas in PARagraph mode

Typesetting Inference Rules

Didier Rémy

(Version 1.6.0, last modified 17/04/2026)

Abstract

This package provides macros for displaying lists of formulas that are typeset in mixed horizontal and vertical modes. The package is two-folded.

The first part is an environment `mathpar` that generalizes the math display mode to allow several formulas on the same line, and several lines in the same display. The arrangement of the sequence of formulas into lines is automatic depending on the line width and on a minimum inter-formula space and line width alike words in a paragraphs (in centerline mode). A typical application is displaying a set of type inference rules.

The second part is a macro `inferrule` to typeset inference rules themselves. Here again, both premises and conclusions are presented as list of formulas that should be displayed in almost the same way, except that the width is not fixed in advance; and the inference rule should use no more width than necessary so that other inference rules are given a chance to appear on the same line.

Although `mathpar` and `inferrule` look similar in their specification, and are often used in combination, they are in fact completely different in their implementations.

1 The `mathpar` environment

The `mathpar` environment is a “paragraph mode for formulas”. It allows to typeset long list of formulas putting as many as possible on the same line:

```
\begin{mathpar}
A-Formula \and           A-Formula   Longer-Formula
Longer-Formula \and     And           The - Last - One
And \and The-Last-One
\end{mathpar}
```

Formulas are separated by `\and` (or equivalently by a blank line). To enforce a vertical break it suffices to replace `\and` by `\\`.

The implementation of `mathpar` entirely relies on the paragraph mode for text. It starts a new paragraph, and a math formula within a paragraph, after adjusting the spacing and penalties for breaks. Then, it simply binds `\and` to something like `\goodbreak`.

Since **version 1.5.0**, the environment `mathpar` reduces the vertical skip using T_EX variable `\abovedisplayshortskip` instead of `\abovedisplayskip` when the first line of the display is short enough not to overlap with the last line before the display. For instance, when this is activated, you may compare this short line:

one *two*

preceding the display, with this much longer last line:

one *two*

This behavior can be set automatically with the package option `shortdisplay`, which is currently not set by default. It can be locally activated (resp. deactivated) with boolean `\shortmathparfalse` (resp. `\shortmathpartrue`). When the option is deactivated a short display following a short line would be displayed as:

one *two*

with a full `\displayskip` instead of a `\shortdisplayskip` amount.

An alternative environment `mathparpagebreakable` may also be used to allow for page breaks in the middle of the display. This however cancels the previous optimization for short displays.

2 The inferrule macro

The `inferrule` macro is designed to typeset inference rules. It should only¹ be used in math mode (or display math mode).

The basic use of the rule is

```
\inferrule
{one \\ two \\ three \\ or \\ more \\ premisses}
{and \\ any \\ number \\ of \\ conclusions \\ as \\ well}
```

This is the rendering on a large page

$\frac{\textit{one} \quad \textit{two} \quad \textit{three} \quad \textit{or} \quad \textit{more} \quad \textit{premisses}}{\textit{and} \quad \textit{any} \quad \textit{number} \quad \textit{of} \quad \textit{conclusions} \quad \textit{as} \quad \textit{well}}$
--

¹Even though the basic version may work in text mode, we discourage its use in text mode; the star-version cannot be used in text-mode

However, the same formula on a narrower page will automatically be typeset like that:

$$\begin{array}{c}
 \textit{one} \\
 \textit{two} \quad \textit{three} \quad \textit{or} \\
 \textit{more} \quad \textit{premises} \\
 \hline
 \textit{as} \quad \textit{well} \\
 \textit{of} \quad \textit{conclusions} \\
 \textit{and} \quad \textit{any} \quad \textit{number}
 \end{array}$$

An inference rule is mainly composed of a premise and a conclusion. The premise and the conclusions are both list of formulas where the elements are separated by `\`.

Note the asymmetry between typesetting of the premises and of conclusions where lines closer to the center are fit first.

A newline can be forced by adding an empty line `\\`

$$\begin{array}{c}
 \backslash\textit{inferrule} \\
 \{aa \\ \\ bb\} \\
 \{dd \\ \\ ee \\ \\ ff\}
 \end{array}
 \quad
 \begin{array}{c}
 aa \\
 bb \\
 \hline
 dd \quad ee \quad ff
 \end{array}$$

2.1 Single rules

Single rules are the default mode. Rules are aligned on their fraction bar, as illustrated below:

$$\begin{array}{c}
 aa \quad bb \\
 \hline
 ee
 \end{array}
 \quad
 \begin{array}{c}
 aa \\
 bb \quad ee \\
 \hline
 ee
 \end{array}$$

If the premise or the conclusion is empty, then the fraction bar is not typeset and the premise or the conclusion is centered (when both of them are empty, which does not make sense, we substitute them by question marks with a warning):

$$\begin{array}{c}
 \backslash\textit{inferrule} \{\}{aa} + \\
 \backslash\textit{inferrule} \{aa \\ \\ aa\}\{\} + \\
 \backslash\textit{inferrule} \{\}\{\}
 \end{array}
 \quad
 \begin{array}{c}
 aa + \frac{aa}{aa} + \frac{?}{?}
 \end{array}$$

Use `{ }` instead of `{}` to typeset an axiom. For instance:

$$\begin{array}{c}
 \backslash\textit{inferrule} \{\}{aa} + \\
 \backslash\textit{inferrule} \{aa\}\{\}
 \end{array}
 \quad
 \begin{array}{c}
 \frac{aa}{aa} + \frac{aa}{aa}
 \end{array}$$

The macro `\inferrule` accepts a label as optional argument, which will be typeset on the top left corner of the rule:

$$\begin{array}{c}
 \backslash\textit{inferrule} [Yop] \\
 \{aa \\ \\ bbc \\ \\ cc\} \\
 \{dd\}
 \end{array}
 \quad
 \begin{array}{c}
 YOP \\
 aa \quad bbc \quad cc \\
 \hline
 dd
 \end{array}$$


```


$$\frac{a \quad bbb}{cc}$$


```

`$$\frac{a \quad bbb}{cc}$$`

```


$$\frac{a \quad bbb}{cc}$$


```

`$$\frac{a \quad bbb}{cc}$$`

Since vertical skip does not take header and footer into account, which is usually better but sometimes odd, this can be adjusted explicitly:

```


$$\frac{a \quad bbb}{cc}$$


```

`$$\frac{a \quad bbb}{cc}$$`

Finally, it is also possible to provide its own definition of fraction by

```


$$\frac{a \quad bbb}{cc}$$


```

`$$\frac{a \quad bbb}{cc}$$`

Customizing the horizontal skip between premises (default value is 2em).

```


$$\frac{a \quad bbb}{cc}$$


```

`$$\frac{a \quad bbb}{cc}$$`

$$\frac{a \quad bbb}{cc}$$

Customizing the vertical space between premises (default value is empty).

Notice that leaving it empty and setting `vskip` to 0em is not quite equivalent as show below between the third and fourth rules (because the typesetting cannot use the primitive typesetting of fractions).

```


$$\frac{a \quad bbb}{cc}$$


```

`$$\frac{a \quad bbb}{cc}$$`

$$\begin{array}{cc}
aa & aa \\
bbb & bbb \\
\hline
cc &
\end{array}
\qquad
\begin{array}{cc}
aa & aa \\
bbb & bbb \\
\hline
cc &
\end{array}
\qquad
\begin{array}{cc}
aa & aa \\
bbb & bbb \\
\hline
cc &
\end{array}$$

$$\begin{array}{cc}
aa & aa \\
bbb & bbb \\
\hline
cc &
\end{array}$$

2.4 Tabulars in inference rules

Although you probably do not want to do that, you may still use tabular, array or minipage environments inside inference rules, but between braces, as follows:

```

\infer [Tabular-Rule]
{some \math and \
  {\begin{tabular}[b]{|l|r|}
  \hline Ugly & and
  \[1ex]\hline
  table & text
  \\\hline
  \end{tabular}} \
  {\begin{minipage}[b]{6em}
  Do you really wish
  to do that?
  \end{minipage}} \
}
{some \conclusions}

```

TABULAR-RULE

<i>some</i>	<i>math</i>	<i>and</i>
Ugly	and	Do you re-
table	text	ally wish to
<i>some</i>		<i>conclusions</i>

2.5 Derivation trees

To help writing cascades of rules forming a derivation tree, inference rules can also be aligned on their bottom line. For this, we use the star-version:

```

\inferrule*
  {\inferrule* {aa \ bb}{cc}
  \ dd}
  {ee}

```

$$\begin{array}{ccc}
aa & bb & \\
\hline
cc & & dd \\
\hline
ee & &
\end{array}$$

The star version can also take an optional argument, but with a different semantics. The optional argument is parsed by the `keyval` package, so as to offer

a set of record-like options:

key	arg	Effect
<code>before</code>	<i>tex</i>	Execute <i>tex</i> before typesetting the rule. Useful for instance to change the maximal width of the rule.
<code>width</code>	<i>d</i>	Set the width of the rule to <i>d</i>
<code>narrower</code>	<i>d</i>	Set the width of the rule to <i>d</i> times <code>\hsize</code> .
<code>lab</code>	<i>ℓ</i>	Put label <i>ℓ</i> on the top of the rule as with the non-start version.
<code>Lab</code>	<i>ℓ</i>	same as <code>lab</code>
<code>left</code>	<i>ℓ</i>	Put label <i>ℓ</i> on the left of the rule
<code>Left</code>	<i>ℓ</i>	Idem, but as if label <i>ℓ</i> had zero width.
<code>Right</code>	<i>ℓ</i>	As <code>Left</code> , but on the right of the rule.
<code>right</code>	<i>ℓ</i>	As <code>left</code> , but on the right of the rule.
<code>leftskip</code>	<i>d</i>	Cheat by (skip negative space) <i>d</i> on the left side.
<code>rightskip</code>	<i>d</i>	Cheat by <i>d</i> on the right side of the rule.
<code>vdots</code>	<i>d</i>	Raise the rule by <i>d</i> and insert vertical dots.

We remind at the end the global options that we've seen above that can also be set locally in derivation trees:

<code>sep</code>	<i>d</i>	Set the separation between premises and conclusions to <i>s</i> .
<code>flushleft</code>	–	flush premises to the left hand side
<code>center</code>	–	center premises on each line.
<code>rewrite</code>	<i>d</i>	
<code>myfraction</code>	<i>tex</i>	set fraction to <i>tex</i> command
<code>fraction</code>	<i>lmr</i>	set fraction pattern to <i>lm...mr</i> with leaders.
<code>vskip</code>	<i>d</i>	Set the vertical skip between premises and conclusions to <i>h</i> .
<code>vcenter</code>		Make the rule centered around the fraction line as the non-star version

Here is an example of a complex derivation:

$$\begin{array}{r}
 \begin{array}{ccc}
 a & & a \\
 bb & cc & dd \\
 \hline
 & \vdots & \\
 ee & & ff \quad gg \\
 \hline
 \text{FOO} & hh & \quad \quad \quad \frac{XX}{uu} \quad \quad \quad vv \\
 \hline
 \text{TOTAL} & & ww
 \end{array} \\
 \hline
 (1) \qquad \qquad \qquad (when\ n > 0)
 \end{array}$$

and its code

```

\inferrule*
  [left=Total, rightstyle=\em, right={{when $n > 0$}}]
  {\inferrule* [Left=Foo]
    {\inferrule*
      [Right=Bar, rightstyle=\bf,
        leftskip=2em, rightskip=vdots=1.5em]
      {a \ a \ \ \ bb \ cc \ dd}
      {ee}
      \ \ ff \ \ gg}
    {hh}
  \ \
  \inferrule* [lab=XX]{uu \ \ vv}{ww}}
  {(1)}

```

2.6 Label styles

The package uses `\DefTirNameStyle`, `\LabTirNameStyle`, `\LeftTirNameStyle`, and `\RightTirNameStyle` to typeset labels introduced with the default option, `Lab-`, `Left-`, or `Right-`, respectively (or their uncapitalized variants). This can safely be redefined by the user. `\DefTirName` is normally used for defining occurrences (*i.e.* in rule `\inferrule`) while the three other forms are used for referencing names (*i.e.* in the star-version). The styles can also be redefined using labeled-arguments of the star-version of `\inferrule` as described in table below.

Instead of just changing the style, the whole typesetting of labels may be changed by redefining the commands `\DefTirName`, `\LabTirName`, `\LeftTirName`, and `\RightTirName`, each of which receives the label to be typeset as argument.

Notice, that if the package `hyperref` is loaded, one can automatically attach hypertargets to rule definitions names of rules defined with the non-star version of `\inferrule` and refers to them by `\RefTirName` defined as follows:

```

\renewcommand{\DefTirName}[1]{\hypertarget{#1}{\TirName {#1}}}
\newcommand{\RefTirName}[1]{\hyperlink{#1}{\TirName {#1}}}

```

key	arg	Effect
<code>style</code>	<i>tex</i>	set the default style for labels to <i>tex</i>
<code>leftstyle</code>	<i>tex</i>	idem for labels
<code>rightstyle</code>	<i>tex</i>	idem for right labels

2.7 Star *v.s.* non-star version

The package also defines `\infer` as a shortcut for `\inferrule` but only if it is not previously defined.

There are two differences between the plain and star versions of `\inferrule`. The plain version centers the rule on the fraction line, while the star one centers the rule on the last conclusion, so as to be used in derivation trees.

Another difference is that the optional argument of the plain version is a label to always be placed on top of the rule, while the ***-version takes a record of arguments. Hence, it can be parameterized in many more ways.

One may recover the plain version from the star version by passing the extra argument `vcenter` as illustrated below (the base line is aligned with the dotted line):

$$\begin{array}{ccc}
 & & \text{aaaa} \\
 & & \frac{aa \quad bb}{cc \quad cc} \\
 \dots\dots\dots & dd & \dots\dots\dots \frac{\text{aaaa}}{aa \quad bb} \dots\dots\dots \\
 & & \frac{cc \quad cc}{dd}
 \end{array}$$

This is convenient, for instance to typeset rules with side conditions and keep them attached to the rule:

$$\begin{array}{cc}
 \text{Pos} & \text{NEG} \\
 \frac{aa \quad aa}{cc} \text{ (if } n > 0) & \frac{aa \quad aa}{cc} \text{ (if } n < 0)
 \end{array}$$

Or differently,

$$\begin{array}{cc}
 \text{Pos} \frac{\text{(if } n > 0) \quad aaa \quad aaa}{cc} & \text{NEG} \frac{\text{(if } n < 0) \quad aaa \quad aaa}{cc}
 \end{array}$$

2.8 Triple rules

There is also a version `\triplerule` originally design to print Hoare triples as rules, which will be rendered as follows, horizontally aligning rules at their

bottom horizontal rule line.

$ \begin{array}{c} \text{HOARE} \\ ppppp \\ \hline pppp \quad pppp \\ \hline mmm \\ \hline mmm \quad mm \\ \hline cccc \quad cccc \end{array} $	$ \begin{array}{c} \text{RULE} \\ ppppp \\ \hline pppp \quad pppp \\ \hline mmm \quad mmm \\ \hline cccc \quad \left(\begin{array}{c} \text{top} \\ \text{bot} \end{array} \right) \end{array} $
--	--

The first rule typeset as follows (in an environment where `\hsize` is 8em to follow its typesetting on several lines).

```

\triplerule[Hoare]
  {ppppp \\\ pppp \\\ pppp}
  {mmm \\\ mmm \\\ mm}
  {cccc \\\ cccc}

```

Here is the code for the second rule

```

\triplerule[Rule]
  {ppppp \\\ pppp \\\ pppp}
  {mmm \\\ mmm}
  {cccc \\\
    {\left(\begin{array}{c}
      top \\
      bot
    \end{array}\right)}
  }
}

```

Notice that the array environment is surrounded by braces, as explained in section §2.4.

2.9 Implementation

The main macro in the implementation of inference rules is the one that either premises and conclusions. The macros uses two box-registers one `hbox` for typesetting each line and one `vbox` for collecting lines. The premise appears as a list with `\\` as separator. Each element is considered in turn typeset in a `hbox` in display math mode. Its width is compare to the space left on the current line. If the box would not fit, the current horizontal line is transferred to the vertical box and emptied. Then, the current formula can safely be added to the horizontal line (if it does not fit, nothing can be done). When moved to the vertical list, lines are aligned on their center (as if their left-part was a left overlapped). At the end the `vbox` is readjusted on the right.

This description works for conclusions. For premises, the elements used to be processes in reverse order and the vertical list is simply built upside down, which was annoying when using counters in the premises. This has recently

been fixed, with a quite different implementation. In case of problem, the old behavior can be recovered by calling `\MprRecoverOlderVerPremise`.

For example,

```
\newcounter \clab
\newcommand {\lab}[1]
  {\stepcounter{\clab}#1~(\theclab)}{}}
\infer [New]
  {\lab A \\\ \lab B \ \ \lab C}
  {\lab D\ \ \lab E \ \ \lab F}
```

produces:

NEW BEHAVIOR	OLD BEHAVIOR
$\frac{\begin{array}{ccc} & A (1) & \\ B (2) & & C (3) \end{array}}{D (4) \quad E (5) \quad F (6)}$	$\frac{\begin{array}{ccc} & A (3) & \\ B (2) & & C (1) \end{array}}{D (4) \quad E (5) \quad F (6)}$

3 Other options for the mathpar environment

The vertical space in `mathpar` is adjusted by `\MathparLineskip`. To restore the normal paragraph parameters in `mathpar` mode (for instance for some inner paragraph), use the command `\MathparNormalpar`. The environment uses `\MathparBindings` to rebind `\`, `\and`, and `\par`. You can redefine thus command to change the default bindings or add your own.

4 Examples

See the source of this documentation —the file `mathpartir.tex`— for full examples.

5 H_EV_EA compatibility

The package also redefines `\hva` to do nothing in \LaTeX mode in both `mathpar` environment nor in inference rules

In $\text{H}_E\text{V}_E\text{A}$ mode and `mathpar` environment, `\and` will always produce a vertical break in `mathpar` environment; to obtain a horizontal break, use `\hva \and` instead. In $\text{H}_E\text{V}_E\text{A}$ mode inference rules, `\` and `\\` will behaves as in `tex`. In addition you may write `\hva\\` to produce the behavior `\` and `\\` to produce the behavior of `\\`.

For instance, by default the following code,

```
\begin{mathpar}
\inferrule* [Left=Foo]{}{}

\inferrule* [Left=Foo]
```

```

{\inferrule* [Right=Bar,width=8em,
             leftskip=2em,rightskip=2em,vdots=1.5em]
  {a \ a \ bb \ cc \ dd}
  {ee}
  \ ff \ gg}
{hh}
\and
\inferrule* [lab=XX]{uu \ v}{ww}
\end{mathpar}

```

which typesets in T_EX as follows,

$$\begin{array}{c}
 \begin{array}{c}
 a \quad a \\
 \hline
 bb \quad cc \quad dd \quad \text{BAR} \\
 \vdots \\
 ee \quad ff \quad gg \\
 \hline
 hh
 \end{array}
 \quad
 \begin{array}{c}
 \text{XX} \\
 \hline
 uu \quad vv \\
 \hline
 ww
 \end{array}
 \end{array}$$

would appear as follows with the compatible H_EV_EA mode:

$$\begin{array}{c}
 \begin{array}{c}
 a \quad a \quad bb \quad cc \quad dd \\
 \hline
 ee \quad \text{BAR} \quad ff \quad gg \\
 \hline
 hh
 \end{array}
 \quad
 \begin{array}{c}
 \text{XX} \\
 \hline
 uu \quad vv \\
 \hline
 ww
 \end{array}
 \end{array}$$

To obtain (almost) the same rendering as in T_EX, it could be typed as

```

\begin{mathpar}
  \inferrule* [Left=Foo]
  {\inferrule* [Right=Bar,width=8em,
               leftskip=2em,rightskip=2em,vdots=1.5em]
    {a \ a \ \hva bb \ cc \ dd}
    {ee}
    \ ff \ gg}
  {hh}
  \hva \and
\inferrule* [lab=XX]{uu \ v}{ww}
\end{mathpar}

```

Actually, it would be typeset and follows with the compatible H_EV_EA mode:

$$\begin{array}{c}
 \begin{array}{c}
 a \quad a \\
 \hline
 bb \quad cc \quad dd \quad \text{BAR} \\
 ee \quad ff \quad gg \\
 \hline
 hh
 \end{array}
 \quad
 \begin{array}{c}
 \text{XX} \\
 \hline
 uu \quad vv \\
 \hline
 ww
 \end{array}
 \end{array}$$