

MATH formulas in PARagraph mode

Typesetting Inference Rules

Didier Rémy

(Version 1.4.0, last modified 15/02/2020)

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 \and The-Last-One    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 TeX 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:

$$\begin{array}{ccc} one & two & three \\ four & & five \end{array}$$

This is to be compared with

$$\begin{array}{ccc} one & two & three \end{array}$$

which would overlap with the previous line. This behavior is automatic. Currently, it can be canceled with `\shortmathparfalse`, in case of a problem with the new behavior. However, there should be no reason to do so and this options may be removed in the future.

An alternative environment `mathparpagebreakable` allows for page breaks in the middle of the display. This cancels the previous optimisation 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

<i>one</i>	<i>two</i>	<i>three</i>	<i>or</i>	<i>more</i>	<i>premises</i>	
<i>and</i>	<i>any</i>	<i>number</i>	<i>of</i>	<i>conclusions</i>	<i>as</i>	<i>well</i>

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:

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

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 `\\\\`

$$\begin{array}{l}
 \text{\texttt{\backslash inferrule}} \\
 \text{\texttt{\{aa \\\\ bb\}}} \\
 \text{\texttt{\{dd \ \ ee \ \ ff\}}}
 \end{array}
 \qquad
 \begin{array}{c}
 aa \\
 bb \\
 \hline
 dd \quad ee \quad ff
 \end{array}$$

2.1 Single rules

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

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

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):

$$\begin{array}{l}
 \text{\texttt{\backslash inferrule \{\}\{aa\} +}} \\
 \text{\texttt{\backslash inferrule \{aa \\\\ aa\}\{\} +}} \\
 \text{\texttt{\backslash inferrule \{\}\{\}}}
 \end{array}
 \qquad
 \begin{array}{c}
 aa + \frac{aa}{aa} + \frac{?}{?}
 \end{array}$$

Use use `{ }` instead of `{}` to get an axiom for instance:

$$\begin{array}{l}
 \text{\texttt{\backslash inferrule { }\{aa\} +}} \\
 \text{\texttt{\backslash inferrule \{aa\}{ } }}
 \end{array}
 \qquad
 \begin{array}{c}
 \frac{\quad}{aa} + \frac{aa}{\quad}
 \end{array}$$

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

$$\begin{array}{c} \text{\texttt{\textbackslash inferrule [yop]}} \\ \text{\texttt{\{aa \textbackslash\ bb\}}} \\ \text{\texttt{\{cc\}}} \end{array} \qquad \begin{array}{c} \text{YOP} \\ \frac{aa \quad bb}{cc} \end{array}$$

See section 2.6 for changing typesetting of labels. A label can also be placed next to the rule directly, since the rule is centered:

$$\begin{array}{c} \text{\texttt{\textbackslash inferrule}} \\ \text{\texttt{\{aa \textbackslash\ bb\}}} \\ \text{\texttt{\{cc\}}} \\ \text{\texttt{\textbackslash quad (\textbackslash textsc {Yop})}} \end{array} \qquad \begin{array}{c} \frac{aa \quad bb}{cc} \quad (\text{YOP}) \end{array}$$

2.2 Customizing presentation

By default, lines are centered in inference rules. However, this can be changed by either `\mprset{flushleft}` or `\mprset{center}`. For instance,

$$\begin{array}{c} \text{\texttt{\$ \$ \mprset{flushleft}}} \\ \text{\texttt{\textbackslash inferrule}} \\ \text{\texttt{\{a \textbackslash\ bbb \textbackslash\ \textbackslash\ ccc \textbackslash\ dd\}}} \\ \text{\texttt{\{dd \textbackslash\ ee \textbackslash\ ff\} \$ \$}} \end{array} \qquad \begin{array}{c} \frac{a \quad bb}{ccc \quad dddd} \\ \frac{gg}{e \quad ff} \end{array}$$

Note that lines are aligned independently in the premise and the conclusion, which are both themselves centered. In particular, left alignment will not affect a single-line premise or conclusion.

2.3 Customizing rules

One may wish to change use rules for rewriting rule or implications, etc. There is a generic way of definition new rules by providing three parts: a tail, a body, and a head. The rule will then be built by joining all three components in this order and filling the body with leaders to extend as much as necessary. Here are examples

$$\begin{array}{c} \text{\texttt{\$ \$ \mprset{fraction={==}}}} \\ \text{\texttt{\textbackslash inferrule \{a \textbackslash\ bbb\} \{cc\} \$ \$}} \end{array} \qquad \frac{a \quad bbb}{cc}$$

$$\begin{array}{c} \text{\texttt{\$ \$ \mprset}} \\ \text{\texttt{\{fraction={\models=\textbackslash Rightarrow\}}}} \\ \text{\texttt{\textbackslash inferrule \{a \textbackslash\ bbb\} \{cc\} \$ \$}} \end{array} \qquad \frac{a \quad bbb}{cc}$$

The height and depth of the *body* are used to adjust vertical space. One, may “smash” the body to reduce the vertical space

$$\begin{array}{cc} a & bbb \\ \cdots & \\ & cc \end{array}$$

```

 $\frac{a}{cc}$ 
 $\frac{a}{cc}$ 

```

$$\frac{a}{cc} \quad \frac{a}{cc}$$
$$\begin{array}{c} \text{\texttt{\$}\texttt{\$}\backslash mprset \{sep=6em\}} \\ \text{\texttt{\$}\texttt{\$}\backslash inferrule \{a \ \backslash\ \ bbb\} \{cc\}\texttt{\$}\texttt{\$}} \end{array}$$

$$\begin{array}{ccc} a & & bbb \\ \hline & & cc \end{array}$$

$$\frac{\inferrule{\text{aa}\backslash\text{aa}\backslash\backslash\backslash\text{bbb}\backslash\text{bbb}}{\text{cc}}}{\hspace{3em}}$$

<i>aa</i>	<i>aa</i>	<i>aa</i>	<i>aa</i>	<i>aa</i>	<i>aa</i>	<i>aa</i>	<i>aa</i>
<i>bbb</i>	<i>bbb</i>	<i>bbb</i>	<i>bbb</i>	<i>bbb</i>	<i>bbb</i>	<i>bbb</i>	<i>bbb</i>
<hr/>		<hr/>		<hr/>		<hr/>	
<i>cc</i>		<i>cc</i>		<i>cc</i>		<i>cc</i>	

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
<hr/>		do that?	
<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}
```

<i>aa</i>	<i>bb</i>
<hr/>	
<i>cc</i>	<i>dd</i>
<hr/>	
<i>ee</i>	

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

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

Here is an example of a complex derivation:

$$\begin{array}{c}
 \begin{array}{ccccccc}
 & a & & a & & & \\
 & bb & & cc & & dd & \\
 \hline
 & & & & & \text{Bar} & \\
 & ee & & \vdots & & & \\
 \text{Foo} & \hline & ff & & gg & & \\
 & & & & & \text{XX} & \\
 & & & & & uu & \\
 & & & & & & vv \\
 \hline
 \text{TOTAL} & \hline & hh & & & & ww & \\
 & & & & & & & (when\ n > 0)
 \end{array}
 \end{array}
 \quad (1)$$

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 loader, one can automatically attach hypertarget 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
style	<i>tex</i>	set the default style for labels to <i>tex</i>
leftstyle	<i>tex</i>	idem for labels
rightstyle	<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):

$$\begin{array}{ccc}
 & & aaaa \\
 & aa & bb \\
 & \hline
 & cc & cc \\
 \dots\dots\dots & dd & \dots\dots\dots
 \end{array}
 \qquad
 \begin{array}{ccc}
 & & aaaa \\
 & aa & bb \\
 & \hline
 & cc & cc \\
 \dots\dots\dots & dd & \dots\dots\dots
 \end{array}$$

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

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

Or differently,

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

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} \\ \text{ppppp} \\ \hline \text{pppp} \quad \text{pppp} \\ \hline \text{mmm} \\ \hline \text{mmm} \quad \text{mm} \\ \hline \text{cccc} \quad \text{cccc} \end{array} $	$ \begin{array}{c} \text{RULE} \\ \text{ppppp} \\ \hline \text{pppp} \quad \text{pppp} \\ \hline \text{mmm} \quad \text{mmm} \\ \hline \text{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 behaviour can be recovered by calling `\MprRecoverOlderVerPremise`.

For example,

```
\newcount \clab
\newcommand {\lab}[1]
  {\global\advance \clab by 1\relax (\the\clab)}
\infer [New] {\lab A \\\lab B} {\lab C \\\lab D}
```

produces:

NEW BEHAVIOR			OLD BEHAVIOR		
(1)			(3)		
(2)	(3)		(2)	(1)	
(4)	(5)	(6)	(4)	(5)	(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 $\text{H}_\text{E}\text{V}_\text{E}\text{A}$ compatibility

The package also redefines `\hva` to do nothing in `mathpar` environment and nor in inference rules.

In `HeVeA`, `\and` will always produce a vertical break in `mathpar` environment; to obtain a horizontal break, use `\hva \and` instead. Conversely, `\` will always produce a horizontal break in type inference rules; to obtain a vertical break, use `\hva \` instead.

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 \\\ vv}{ww}
\end{mathpar}

```

which typesets in T_EX as follows,

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

would appear as follows with the compatible H_EV_EA mode:

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

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 \\\ vv}{ww}
\end{mathpar}

```

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

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