# MATH formulas in PARragraph mode

# Typesetting Inference Rules

Didier Rémy

(Version 1.3.0, last modified 21/09/2015)

#### 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 interformula 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 par 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:

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.

### 2 The inferrule macro

The inferrule macro is designed to typeset inference rules. It should only<sup>1</sup> 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

$$\frac{one \ two \ three \ or \ more \ premises}{and \ any \ number \ of \ conclusions \ as \ well}$$

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

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

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 type setting 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 \\\\

<sup>&</sup>lt;sup>1</sup>Even 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

## 2.1 Single rules

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

$$\frac{aa \quad bb}{ee} \qquad \qquad \frac{bb \quad ee}{ee}$$

If the premise or the conclusion is empty, then the fraction bar is not typeset and the premise or the conclusion is centered:

\inferrule {}{aa} + \inferrule {aa \\\\ aa}{} 
$$aa + aa$$

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

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

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:

#### 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,

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

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

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

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

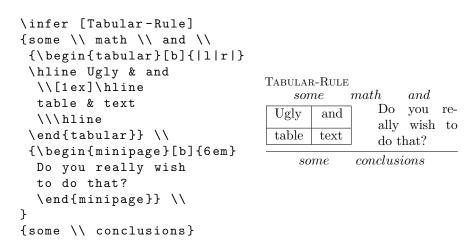
Customizing the horizontal skip between premises (default value is 2em).

\$\$\mprset {sep=6em} \inferrule {a \\ bbb} {cc}\$\$ 
$$\frac{a \qquad bbb}{cc}$$

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).

#### 2.4 Tabulars in inference rules

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



#### 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:

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:

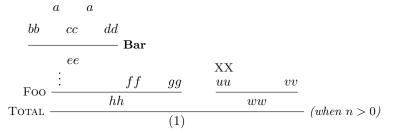
key	arg	Effect
before	tex	Execute <i>tex</i> before typesetting the rule. Useful for instance to change the maximal width of the rule.
width	d	Set the width of the rule to $d$
narrower	d	Set the width of the rule to $d$ times \hsize.
lab	$\ell$	Put label $\ell$ on the top of the rule as with the non-start version.
Lab	$\ell$	same as lab
left	$\ell$	Put label $\ell$ on the left of the rule
Left	$\ell$	Idem, but as if label $\ell$ had zero width.
Right	$\ell$	As Left, but on the right of the rule.
right	$\ell$	As left, but on the right of the rule.
leftskip	d	Cheat by (skip negative space) $d$ on the left side.
rightskip	d	Cheat by $d$ on the right side of the rule.
vdots	d	Raise the rule by $d$ 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	d	Set the separation between premises and conclusions to $s$ .
flushleft	_	flush premises to the left hand side
center	_	center premises on each line.
rewrite	d	
myfraction	tex	set fraction to tex command
fraction	lmr	set fraction pattern to $lmmr$ with leaders.
vskip	d	Set the vertical skip between premises and conclusions to $h$ .
vcenter		Make the rule centered around the fraction line as the non-star version

Here is an example of a complex derivation:



and its code

## 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 uncapitablized 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 \DefTirName, \LabTirName, \LeftTirName, and \RightTirName, each of which receives the label to be typeset as argument.

Finally, the vertical skip

key	arg	Effect
style	tex	set the default style for labels to $tex$
leftstyle	tex	idem for labels
rightstyle	tex	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 start version by passing the extra argument **vcenter** as illustrated below (the base line is aligned with the dotted line):

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

Pos 
$$\frac{aa \quad aa}{cc} \quad \text{(if } n > 0\text{)} \qquad \qquad \frac{aa \quad aa}{cc} \quad \text{(if } n < 0\text{)}$$

Or differently,

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

#### 2.8 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 type-setting 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 must be processes in reverse order and the vertical list is simply built upside down.

## 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.

# $5~~{ m H_EV_EA}$ 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,

would appear as follows with the compatible  $H_{\!\!E}V_{\!\!E}A$  mode:

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

Actually, it would be typeset and follows with the compatible  $H_{\hbox{\ensuremath{\overline{E}}}\mbox{\ensuremath{V}}\mbox{\ensuremath{E}}\mbox{\ensuremath{A}}\mbox{\ensuremath{E}}\mbox{\ensuremath{A}}\mbox{\ensuremath{E}}\mbox{\ensuremath{E}}\mbox{\ensuremath{A}}\mbox{\ensuremath{E}}\mbox{\ensuremath{A}}\mbox{\ensuremath{E}}\mbox{\ensuremath{A}}\mbox{\ensuremath{E}}\mbox{\ensuremath{A}}\mbox{\ensuremath{E}}\mbox{\ensuremath{B}}\mbox{\ensuremath{A}}\mbox{\ensuremath{E}}\mbox{\ensuremath{B}}\mbox{\ensuremath{B}}\mbox{\ensuremath{E}}\mbox{\ensuremath{B}}\mbox{\ensuremath{E}}\mbox{\ensuremath{B}}\mbox{\ensuremath{B}}\mbox{\ensuremath{E}}\mbox{\ensuremath{B}}\mbox{$