MATH formulas in **PAR**agraph mode

Typesetting **I**nference **R**ules

Didier Rémy

(Version 1.3.2, last modified 24/02/2016)

1 Introduction

The package mathpartir provides macros for displaying formulas and lists of formulas that are typeset in mixed horizontal and vertical modes.

The environment `mathpar` 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 alike words are displayed in a paragraphs (in centerline mode). A typical application is displaying a set of type inference rules.

The macro `inferrule` typesets inference rules. Both premises and conclusions are presented as lists of formulas and are typeset in paragraph mode and wrapped into multiple lines whenever necessary.

2 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
Longer-Formula \and
And \and The-Last-One
\end{mathpar}
```

$$A-Formula \qquad Longer-Formula \qquad And \qquad The-Last-One$$

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`.

3 The inferrule macro

The inferrule macro is designed to typeset inference rules. It should only[1] 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

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

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

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

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 `\\\\`

```
\inferrule
   {aa \\\\ bb}
   {dd \\ ee \\ ff}
```

$$\frac{\begin{array}{c} aa \\ bb \end{array}}{\begin{array}{ccc} dd & ee & ff \end{array}}$$

[1] 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

3.1 Single rules

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

$$\frac{aa \qquad bb}{ee} \qquad \frac{\begin{array}{c} aa \\ bb \qquad ee \end{array}}{ee}$$

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}{}
```

$$aa + \begin{array}{c} aa \\ aa \end{array}$$

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

```
\inferrule { }{aa} +
\inferrule {aa}{ }
```

$$\frac{}{aa} + \frac{aa}{}$$

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

```
\inferrule [yop]
   {aa \\ bb}
   {cc}
```

$$\textsc{Yop}\ \frac{aa \qquad bb}{cc}$$

See section 3.6 for changing typesetting of labels. A label can also be placed next to the rule directly, since the rule is centered:

```
\inferrule
   {aa \\ bb}
   {cc}
\quad (\textsc {Yop})
```

$$\frac{aa \qquad bb}{cc}\quad (\textsc{Yop})$$

3.2 Customizing presentation

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

```
$$\mprset{flushleft}
  \inferrule
    {a \\ bbb \\\\ ccc \\ dd}
    {dd \\ ee \\ ff}$$
```

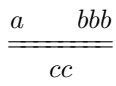
$$\frac{\begin{array}{ll} a & bb \\ ccc & dddd \end{array}}{\begin{array}{ll} e & ff \\ gg & \end{array}}$$

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.

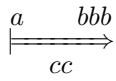
3.3 Customizing rules

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

```
$$\mprset{fraction={===}}
  \inferrule {a \\ bbb} {cc}$$
```

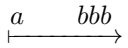



```
$$\mprset
  {fraction={\models=\Rightarrow}}
 \inferrule {a \\ bbb} {cc}$$
```

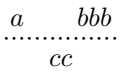


The height and depth of the *body* are used to adjust vertical space. One, may "smash" the body to reduce the vertical space

```
$$\mprset
  {fraction={%
      {\scriptstyle\vdash}%
      {\smash-}%
      {\rightarrow\!\!}%
      }}
 \inferrule {a \\ bbb} {cc}\,\,$$
```

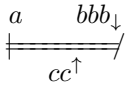



```
$$\mprset {fraction={\cdot\cdots\cdot}}
 \inferrule {a \\ bbb} {cc}$$
```



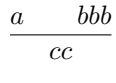
Since vertical skip does not take header and footer into account, which is usually better but sometimes odd, this can be adjusted explicitly:

```
$$\mprset
  {fraction={|=/},
   fractionaboveskip=0.6ex,
   fractionbelowskip=0.4ex}
 \inferrule
   {a \\ bbb_{\downarrow}}
   {cc^{\T\uparrow}}$$
```



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

```
\def \Over #1#2{\hbox{$#1 \over #2$}}
$$\mprset{myfraction=\Over}
  \inferrule {a \\ bbb} {cc}$$
```



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

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

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

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).

```
$$\def\R{\inferrule {aa\\aa\\\\bbb\\bbb} {cc}
  \hspace{3em}}
  \R  \mprset{vskip=0ex}\R  \mprset{vskip=1ex}\R$$
```

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

3.4 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:

```
\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

$$\frac{\begin{array}{ccc} some & math & and \\ \begin{array}{|l|r|}\hline Ugly & and \\ \hline table & text \\ \hline \end{array} & & \begin{array}{l}\text{Do you really wish to}\\ \text{do that?}\end{array} \end{array}}{\begin{array}{cc} some & conclusions \end{array}}$$

3.5 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}
```

$$\dfrac{\dfrac{aa \qquad bb}{cc} \qquad dd}{ee}$$

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 *tex* 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`	ℓ	Put label ℓ on the top of the rule as with the non-start version.
`Lab`	ℓ	same as lab
`left`	ℓ	Put label ℓ on the left of the rule
`Left`	ℓ	Idem, but as if label ℓ had zero width.
`Right`	ℓ	As `Left`, but on the right of the rule.
`right`	ℓ	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 $lm...mr$ 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:

$$\textsc{Total}\ \dfrac{\textsc{Foo}\ \dfrac{\dfrac{a \quad a}{bb \quad cc \quad dd}\ \textbf{Bar}}{\vdots}\ \dfrac{ee}{ff \quad gg}}{hh} \qquad \dfrac{\text{XX}\ \dfrac{}{uu \quad vv}}{ww}}{(1)}\ \textit{(when } n > 0\textit{)}$$

and its code

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

3.6 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 `\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

3.7 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):

$$\cdots\cdots\cdots\ \dfrac{\dfrac{aa \qquad bb}{cc \qquad cc}}{dd}\ \cdots\cdots\cdots\ \dfrac{aa \qquad \dfrac{aa \qquad bb}{cc \qquad cc}}{dd}\ \cdots\cdots\cdots$$

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

$$\textsc{Pos}\ \dfrac{aa \qquad aa}{cc}\ (\text{if } n > 0) \qquad\qquad \textsc{Neg}\ \dfrac{aa \qquad aa}{cc}\ (\text{if } n < 0)$$

Or differently,

$$\textsc{Pos}\ \frac{aaa \qquad aaa}{cc}\ (\text{if } n > 0) \qquad\qquad \textsc{Neg}\ \frac{aaa \qquad aaa}{cc}\ (\text{if } n < 0)$$

3.8 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 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 must be processes in reverse order and the vertical list is simply built upside down.

4 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.

5 Examples

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

6 HeVeA compatibility

The package also redefines `\hva` to do nothing in `mathpar` environment and 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* [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 TEX as follows,

$$\dfrac{\dfrac{\dfrac{}{bb}\quad\overset{a}{\overset{\vdots}{cc}}\quad\overset{a}{dd}}{ee}\,\textsc{Bar}}{\textsc{Foo}\;\dfrac{}{ff}\quad\dfrac{}{gg}}{hh} \qquad\qquad \dfrac{\overset{\textsc{XX}}{\dfrac{}{uu}\quad\dfrac{}{vv}}}{ww}$$

would appear as follows with the compatible HEVEA mode:

$$\dfrac{\dfrac{\dfrac{}{a}\quad\overset{a}{\overset{\vdots}{bb}}\quad\overset{a}{cc}\quad\overset{a}{dd}}{ee}\,\textsc{Bar}}{\textsc{Foo}\;\dfrac{}{ff}\quad\dfrac{}{gg}}{hh} \qquad\qquad \dfrac{\overset{\textsc{XX}}{\dfrac{}{uu}\quad\dfrac{}{vv}}}{ww}$$

To obtain (almost) the same rendering as in TEX, 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 HEVEA mode:

$$\textsc{Foo}\ \dfrac{\dfrac{a \quad a}{bb \quad cc \quad dd}\ \textsc{Bar} \qquad ff \quad gg}{ee \qquad\qquad hh} \qquad \dfrac{\text{XX}}{uu \quad vv \;\Big/\; ww}$$

7 Implementation

```
%% Identification
%% Preliminary declarations

\RequirePackage {keyval}

%% Options
%% More declarations

%% PART I: Typesetting maths in paragraphe mode

%% \newdimen \mpr@tmpdim
%% Dimens are a precious ressource. Uses seems to be local.
\let \mpr@tmpdim \@tempdima

% To ensure hevea \hva compatibility, \hva should expands to nothing
% in mathpar or in inferrule
\let \mpr@hva \empty

%% normal paragraph parametters, should rather be taken dynamically
\def \mpr@savepar {%
  \edef \MathparNormalpar
     {\noexpand \lineskiplimit \the\lineskiplimit
      \noexpand \lineskip \the\lineskip}%
  }

\def \mpr@rulelineskip {\lineskiplimit=0.3em\lineskip=0.2em plus 0.1em}
\def \mpr@lesslineskip {\lineskiplimit=0.6em\lineskip=0.5em plus 0.2em}
\def \mpr@lineskip  {\lineskiplimit=1.2em\lineskip=1.2em plus 0.2em}
\let \MathparLineskip \mpr@lineskip
\def \mpr@paroptions {\MathparLineskip}
\let \mpr@prebindings \relax

\newskip \mpr@andskip \mpr@andskip 2em plus 0.5fil minus 0.5em

\def \mpr@goodbreakand
   {\hskip -\mpr@andskip  \penalty -1000\hskip \mpr@andskip}
\def \mpr@and {\hskip \mpr@andskip}
\def \mpr@andcr {\penalty 50\mpr@and}
\def \mpr@cr {\penalty -10000\mpr@and}
```

```
%\def \mpr@cr {\penalty -10000\vadjust{\vbox{}}\mpr@and}
\def \mpr@eqno #1{\mpr@andcr #1\hskip 0em plus -1fil \penalty 10}

\def \mpr@bindings {%
  \let \and \mpr@andcr
  \let \par \mpr@andcr
  \let \\\mpr@cr
  \let \eqno \mpr@eqno
  \let \hva \mpr@hva
  }
\let \MathparBindings \mpr@bindings

% \@ifundefined {ignorespacesafterend}
%    {\def \ignorespacesafterend {\aftergroup \ignorespaces}

\newenvironment{mathpar}[1][]
  {$$\mpr@savepar \parskip 0em \hsize \linewidth \centering
     \vbox \bgroup \mpr@prebindings \mpr@paroptions #1\ifmmode $\else
     \noindent $\displaystyle\fi
     \MathparBindings}
  {\unskip \ifmmode $\fi\egroup $$\ignorespacesafterend}

\newenvironment{mathparpagebreakable}[1][]
  {\begingroup
   \par
   \mpr@savepar \parskip 0em \hsize \linewidth \centering
      \mpr@prebindings \mpr@paroptions #1%
      \vskip \abovedisplayskip \vskip -\lineskip%
     \ifmmode  \else  $\displaystyle\fi
     \MathparBindings
  }
  {\unskip
   \ifmmode $\fi \par\endgroup
   \vskip \belowdisplayskip
   \noindent
  \ignorespacesafterend}

% \def \math@mathpar #1{\setbox0 \hbox {$\displaystyle #1$}\ifnum
%     \wd0 < \hsize  $$\box0$$\else \bmathpar #1\emathpar \fi}

%%% HOV BOXES

\def \mathvbox@ #1{\hbox \bgroup \mpr@normallineskip
  \vbox \bgroup \tabskip 0em \let \\ \cr
  \halign \bgroup \hfil $##$\hfil\cr #1\crcr \egroup \egroup
  \egroup}

\def \mathhvbox@ #1{\setbox0 \hbox {\let \\\qquad $#1$}\ifnum \wd0 < \hsize
      \box0\else \mathvbox {#1}\fi}
```

```
%% Part II -- operations on lists

\newtoks \mpr@lista
\newtoks \mpr@listb

\long \def\mpr@cons #1\mpr@to#2{\mpr@lista {\\{#1}}\mpr@listb \expandafter
{#2}\edef #2{\the \mpr@lista \the \mpr@listb}}

\long \def\mpr@snoc #1\mpr@to#2{\mpr@lista {\\{#1}}\mpr@listb \expandafter
{#2}\edef #2{\the \mpr@listb\the\mpr@lista}}

\long \def \mpr@concat#1=#2\mpr@to#3{\mpr@lista \expandafter {#2}\mpr@listb
\expandafter {#3}\edef #1{\the \mpr@listb\the\mpr@lista}}

\def \mpr@head #1\mpr@to #2{\expandafter \mpr@head@ #1\mpr@head@ #1#2}
\long \def \mpr@head@ #1#2\mpr@head@ #3#4{\def #4{#1}\def#3{#2}}

\def \mpr@flatten #1\mpr@to #2{\expandafter \mpr@flatten@ #1\mpr@flatten@ #1#2}
\long \def \mpr@flatten@ \\#1\\#2\mpr@flatten@ #3#4{\def #4{#1}\def #3{\\#2}}

\def \mpr@makelist #1\mpr@to #2{\def \mpr@all {#1}%
   \mpr@lista {\\}\mpr@listb \expandafter {\mpr@all}\edef \mpr@all {\the
   \mpr@lista \the \mpr@listb \the \mpr@lista}\let #2\empty
   \def \mpr@stripof ##1##2\mpr@stripend{\def \mpr@stripped{##2}}\loop
     \mpr@flatten \mpr@all \mpr@to \mpr@one
     \expandafter \mpr@snoc \mpr@one \mpr@to #2\expandafter \mpr@stripof
     \mpr@all \mpr@stripend
     \ifx \mpr@stripped \empty \let \mpr@isempty 0\else \let \mpr@isempty 1\fi
     \ifx 1\mpr@isempty
   \repeat
}

\def \mpr@rev #1\mpr@to #2{\let \mpr@tmp \empty
   \def \\##1{\mpr@cons ##1\mpr@to \mpr@tmp}#1\let #2\mpr@tmp}

%% Part III -- Type inference rules

\newif \if@premisse
\newbox \mpr@hlist
\newbox \mpr@vlist
\newif \ifmpr@center \mpr@centertrue
\def \mpr@vskip {}
\def \mpr@htovlist {%
   \setbox \mpr@hlist
      \hbox {\strut
             \ifmpr@center \hskip -0.5\wd\mpr@hlist\fi
             \unhbox \mpr@hlist}%
   \setbox \mpr@vlist
      \vbox {\if@premisse
```

```
                \box \mpr@hlist
                \ifx \mpr@vskip \empty \else
                  \hrule height 0em depth \mpr@vskip width 0em
                \fi
                \unvbox \mpr@vlist
             \else
                \unvbox \mpr@vlist
                \ifx \mpr@vskip \empty \else
                  \hrule height 0em depth \mpr@vskip width 0em
                \fi
                \box \mpr@hlist
             \fi}%
}
% OLD version
% \def \mpr@htovlist {%
%    \setbox \mpr@hlist
%       \hbox {\strut \hskip -0.5\wd\mpr@hlist \unhbox \mpr@hlist}%
%    \setbox \mpr@vlist
%       \vbox {\if@premisse \box \mpr@hlist \unvbox \mpr@vlist
%              \else \unvbox \mpr@vlist \box \mpr@hlist
%              \fi}%
% }

\def \mpr@item #1{$\displaystyle #1$}
\def \mpr@sep{2em}
\def \mpr@blank { }
\def \mpr@hovbox #1#2{\hbox
  \bgroup
  \ifx #1T\@premissetrue
  \else \ifx #1B\@premissefalse
  \else
     \PackageError{mathpartir}
       {Premisse orientation should either be T or B}
       {Fatal error in Package}%
  \fi \fi
  \def \@test {#2}\ifx \@test \mpr@blank\else
  \setbox \mpr@hlist \hbox {}%
  \setbox \mpr@vlist \vbox {}%
  \if@premisse \let \snoc \mpr@cons \else \let \snoc \mpr@snoc \fi
  \let \@hvlist \empty \let \@rev \empty
  \mpr@tmpdim 0em
  \expandafter \mpr@makelist #2\mpr@to \mpr@flat
  \if@premisse \mpr@rev \mpr@flat \mpr@to \@rev \else \let \@rev \mpr@flat \fi
  \def \\##1{%
     \def \@test {##1}\ifx \@test \empty
        \mpr@htovlist
        \mpr@tmpdim 0em %%% last bug fix not extensively checked
     \else
      \setbox0 \hbox{\mpr@item {##1}}\relax
      \advance \mpr@tmpdim by \wd0
```

```
      %\mpr@tmpdim 1.02\mpr@tmpdim
      \ifnum \mpr@tmpdim < \hsize
         \ifnum \wd\mpr@hlist > 0
           \if@premisse
             \setbox \mpr@hlist
                \hbox {\unhbox0 \hskip \mpr@sep \unhbox \mpr@hlist}%
           \else
             \setbox \mpr@hlist
                \hbox {\unhbox \mpr@hlist  \hskip \mpr@sep \unhbox0}%
           \fi
         \else
         \setbox \mpr@hlist \hbox {\unhbox0}%
         \fi
      \else
         \ifnum \wd \mpr@hlist > 0
            \mpr@htovlist
            \mpr@tmpdim \wd0
         \fi
         \setbox \mpr@hlist \hbox {\unhbox0}%
      \fi
      \advance \mpr@tmpdim by \mpr@sep
   \fi
   }%
   \@rev
   \mpr@htovlist
   \ifmpr@center \hskip \wd\mpr@vlist\fi \box \mpr@vlist
   \fi
   \egroup
}

%%% INFERENCE RULES

\@ifundefined{@@over}{%
    \let\@@over\over % fallback if amsmath is not loaded
    \let\@@overwithdelims\overwithdelims
    \let\@@atop\atop \let\@@atopwithdelims\atopwithdelims
    \let\@@above\above \let\@@abovewithdelims\abovewithdelims
  }{}

%% The default

\def \mpr@@fraction #1#2{\hbox {\advance \hsize by -0.5em
    $\displaystyle {#1\mpr@over #2}$}}
\def \mpr@@nofraction #1#2{\hbox {\advance \hsize by -0.5em
    $\displaystyle {#1\@@atop #2}$}}

\let \mpr@fraction \mpr@@fraction

%% A generic solution to arrow

```

```
\def \mpr@@fractionaboveskip {0ex}
\def \mpr@@fractionbelowskip {0.22ex}

\def \mpr@make@fraction #1#2#3#4#5{\hbox {%
     \def \mpr@tail{#1}%
     \def \mpr@body{#2}%
     \def \mpr@head{#3}%
     \setbox1=\hbox{$#4$}\setbox2=\hbox{$#5$}%
     \setbox3=\hbox{$\mkern -3mu\mpr@body\mkern -3mu$}%
     \dimen0\ht3\advance\dimen0 by \dp3\relax
     \dimen0 0.5\dimen0\relax
     \advance \dimen0 by \mpr@@fractionaboveskip
     \setbox1=\hbox {\raise \dimen0 \box1}\relax
     \dimen0 -\dimen0\advance \dimen0 \mpr@@fractionaboveskip\dimen0 -\dimen0
     \advance \dimen0 by \mpr@@fractionbelowskip
     \setbox2=\hbox {\lower \dimen0 \box2}\relax
     \setbox0=\hbox {$\displaystyle {\box1 \atop \box2}$}%
     \dimen0=\wd0\box0
     \box0 \hskip -\dimen0\relax
     \hbox to \dimen0 {$%\color{blue}
       \mathrel{\mpr@tail}\joinrel
       \xleaders\hbox{\copy3}\hfil\joinrel\mathrel{\mpr@head}%
     $}}}

%% Old stuff should be removed in next version
\def \mpr@@nothing #1#2
    {$\lower 0.01pt \mpr@@nofraction {#1}{#2}$}
\def \mpr@@reduce #1#2{\hbox
    {$\lower 0.01pt \mpr@@fraction {#1}{#2}\mkern -15mu\rightarrow$}}
\def \mpr@@rewrite #1#2#3{\hbox
    {$\lower 0.01pt \mpr@@fraction {#2}{#3}\mkern -8mu#1$}}
\def \mpr@infercenter #1{\vcenter {\mpr@hovbox{T}{#1}}}

\def \mpr@empty {}
\def \mpr@inferrule
  {\bgroup
     \ifnum \linewidth<\hsize \hsize \linewidth\fi
     \mpr@rulelineskip
     \let \and \qquad
     \let \hva \mpr@hva
     \let \@rulename \mpr@empty
     \let \@rule@options \mpr@empty
     \let \mpr@over \@@over
     \mpr@inferrule@}
\newcommand {\mpr@inferrule@}[3][]
  {\everymath={\displaystyle}%
   \def \@test {#2}\ifx \empty \@test
      \setbox0 \hbox {$\vcenter {\mpr@hovbox{B}{#3}}$}%
   \else
   \def \@test {#3}\ifx \empty \@test
```

```
      \setbox0 \hbox {$\vcenter {\mpr@hovbox{T}{#2}}$}%
   \else
   \setbox0 \mpr@fraction {\mpr@hovbox{T}{#2}}{\mpr@hovbox{B}{#3}}%
   \fi \fi
   \def \@test {#1}\ifx \@test\empty \box0
   \else \vbox
%%% Suggestion de Francois pour les etiquettes longues
%%%   {\hbox to \wd0 {\RefTirName {#1}\hfil}\box0}\fi
      {\hbox {\DefTirName {#1}}\box0}\fi
   \egroup}

\def \mpr@vdotfil #1{\vbox to #1{\leaders \hbox{$\cdot$} \vfil}}

% They are two forms
% \inferrule [label]{[premisses}{conclusions}
% or
% \inferrule* [options]{[premisses}{conclusions}
%
% Premisses and conclusions are lists of elements separated by \\
% Each \\ produces a break, attempting horizontal breaks if possible,
% and  vertical breaks if needed.
%
% An empty element obtained by \\\\ produces a vertical break in all cases.
%
% The former rule is aligned on the fraction bar.
% The optional label appears on top of the rule
% The second form to be used in a derivation tree is aligned on the last
% line of its conclusion
%
% The second form can be parameterized, using the key=val interface. The
% folloiwng keys are recognized:
%
%  width                set the width of the rule to val
%  narrower             set the width of the rule to val\hsize
%  before               execute val at the beginning/left
%  lab                  put a label [Val] on top of the rule
%  lskip                add negative skip on the right
%  left                 put a left label [Val]
%  Left                 put a left label [Val],  ignoring its width
%  right                put a right label [Val]
%  Right                put a right label [Val], ignoring its width
%  leftskip             skip negative space on the left-hand side
%  rightskip            skip negative space on the right-hand side
%  vdots                lift the rule by val and fill vertical space with dots
%  after                execute val at the end/right
%
%  Note that most options must come in this order to avoid strange
%  typesetting (in particular  leftskip must preceed left and Left and
%  rightskip must follow Right or right; vdots must come last
%  or be only followed by rightskip.
```

```
%
%% Keys that make sence in all kinds of rules
\def \mprset #1{\setkeys{mprset}{#1}}
\define@key {mprset}{andskip}[]{\mpr@andskip=#1}
\define@key {mprset}{lineskip}[]{\lineskip=#1}
\define@key {mprset}{lessskip}[]{\lineskip=0.5\lineskip}
\define@key {mprset}{flushleft}[]{\mpr@centerfalse}
\define@key {mprset}{center}[]{\mpr@centertrue}
\define@key {mprset}{rewrite}[]{\let \mpr@fraction \mpr@@rewrite}
\define@key {mprset}{atop}[]{\let \mpr@fraction \mpr@@nofraction}
\define@key {mprset}{myfraction}[]{\let \mpr@fraction #1}
\define@key {mprset}{fraction}[]{\def \mpr@fraction {\mpr@make@fraction #1}}
% To be documented.
\define@key {mprset}{defaultfraction}[]{\let \mpr@fraction \mpr@@fraction}
\define@key {mprset}{sep}{\def\mpr@sep{#1}}
\define@key {mprset}{fractionaboveskip}{\def\mpr@@fractionaboveskip{#1}}
\define@key {mprset}{fractionbelowskip}{\def\mpr@@fractionbelowskip{#1}}
\define@key {mprset}{style}[1]{\def\TirNameStyle{#1}def}
\define@key {mprset}{rightstyle}[1]{\def\RightTirNameStyle{#1}}
\define@key {mprset}{leftstyle}[1]{\def\LeftTirNameStyle{#1}}
\define@key {mprset}{vskip}[1]{\def \mpr@vskip{#1}}

\newbox \mpr@right
\define@key {mpr}{flushleft}[]{\mpr@centerfalse}
\define@key {mpr}{center}[]{\mpr@centertrue}
\define@key {mpr}{rewrite}[]{\let \mpr@fraction \mpr@@rewrite}
\define@key {mpr}{myfraction}[]{\let \mpr@fraction #1}
\define@key {mpr}{fraction}[]{\def \mpr@fraction {\mpr@make@fraction #1}}
\define@key {mpr}{width}{\hsize #1}
\define@key {mpr}{sep}{\def\mpr@sep{#1}}
\define@key {mpr}{before}{#1}
\define@key {mpr}{lab}{\let \DefTirName \LabTirName \def \mpr@rulename {#1}}
\define@key {mpr}{Lab}{\let \DefTirName \LabTirName \def \mpr@rulename {#1}}
\define@key {mpr}{style}[1]{\def\TirNameStyle{#1}def}
\define@key {mpr}{rightstyle}[1]{\def\RightTirNameStyle{#1}}
\define@key {mpr}{leftstyle}[1]{\def\LeftTirNameStyle{#1}}
\define@key {mpr}{vskip}[1]{\def \mpr@vskip{#1}}
\define@key {mpr}{narrower}{\hsize #1\hsize}
\define@key {mpr}{leftskip}{\hskip -#1}
\define@key {mpr}{reduce}[]{\let \mpr@fraction \mpr@@reduce}
\define@key {mpr}{rightskip}
  {\setbox \mpr@right \hbox {\unhbox \mpr@right \hskip -#1}}
\define@key {mpr}{LEFT}{\setbox0 \hbox {$#1$}\relax
     \advance \hsize by -\wd0\box0}

\define@key {mpr}{left}{\setbox0 \hbox {$\LeftTirName {#1}\;$}\relax
     \advance \hsize by -\wd0\box0}
\define@key {mpr}{Left}{\llap{$\LeftTirName {#1}\;$}}
\define@key {mpr}{right}
```

```
  {\setbox0 \hbox {$\;\RightTirName {#1}$}\relax \advance \hsize by -\wd0
   \setbox \mpr@right \hbox {\unhbox \mpr@right \unhbox0}}
\define@key {mpr}{RIGHT}
  {\setbox0 \hbox {$#1$}\relax \advance \hsize by -\wd0
   \setbox \mpr@right \hbox {\unhbox \mpr@right \unhbox0}}
\define@key {mpr}{Right}
  {\setbox \mpr@right \hbox {\unhbox \mpr@right \rlap {$\;\RightTirName {#1}$}}}
\define@key {mpr}{vdots}{\def \mpr@vdots {\@@atop \mpr@vdotfil{#1}}}
\define@key {mpr}{after}{\edef \mpr@after {\mpr@after #1}}
\define@key {mpr}{vcenter}[]{\mpr@vcentertrue}

\newif \ifmpr@vcenter \mpr@vcenterfalse
\newcommand \mpr@inferstar@ [3][]{\begingroup
  \setbox0 \hbox
        {\let \mpr@rulename \mpr@empty \let \mpr@vdots \relax
         \setbox \mpr@right \hbox{}%
         \setkeys{mpr}{#1}%
         $\ifx \mpr@rulename \mpr@empty \mpr@inferrule {#2}{#3}\else
          \mpr@inferrule [{\mpr@rulename}]{#2}{#3}\fi
          \box \mpr@right \mpr@vdots$
          \ifmpr@vcenter \aftergroup \mpr@vcentertrue \fi}
  \ifmpr@vcenter
     \box0
  \else
     \setbox1 \hbox {\strut}
     \@tempdima \dp0 \advance \@tempdima by -\dp1
     \raise \@tempdima \box0
  \fi
  \endgroup}

\def \mpr@infer {\@ifnextchar *{\mpr@inferstar}{\mpr@inferrule}}
\newcommand \mpr@err@skipargs[3][]{}
\def \mpr@inferstar*{\ifmmode
    \let \@do \mpr@inferstar@
  \else
    \let \@do \mpr@err@skipargs
    \PackageError {mathpartir}
      {\string\inferrule* can only be used in math mode}{}%
  \fi \@do}


%%% Exports

% Envirnonment mathpar

\let \inferrule \mpr@infer

% make a short name \infer is not already defined
\@ifundefined {infer}{\let \infer \mpr@infer}{}

```

```
\def \TirNameStyle #1{\small \textsc{#1}}
\def \LeftTirNameStyle #1{\TirNameStyle {#1}}
\def \RightTirNameStyle #1{\TirNameStyle {#1}}

\def \lefttir@name #1{\hbox {\small \LeftTirNameStyle{#1}}}
\def \righttir@name #1{\hbox {\small \RightTirNameStyle{#1}}}
\let \TirName \lefttir@name
\let \LeftTirName \lefttir@name
\let \DefTirName \lefttir@name
\let \LabTirName \lefttir@name
\let \RightTirName \righttir@name

%%% Other Exports

% \let \listcons \mpr@cons
% \let \listsnoc \mpr@snoc
% \let \listhead \mpr@head
% \let \listmake \mpr@makelist
```