drv.mp
derivation trees with METAPOST*

almost a user guide†

Laurent M´
laurent.mehats@gmail.com

documented version: 0.92

$$\dfrac{\dfrac{\dfrac{\dfrac{}{A,\Gamma\vdash B}\,\gamma \quad \dfrac{}{\Delta\vdash C}\,\delta}{A,\Gamma,\Delta\vdash B\wedge C}\,\wedge_R \quad \overset{\theta}{\cdots\cdots} \quad B\wedge C,\Theta\vdash D}{A,\Gamma,\Delta,\Theta\vdash D}\,\text{cut} \quad \dfrac{\dfrac{}{\Gamma,\Delta,\Theta\vdash A\to D}\,\to_R \quad \dfrac{}{E,\Upsilon\vdash F}\,\upsilon}{\Gamma,\Delta,\Theta,(A\to D)\to E,\Upsilon\vdash F}\,\to_L}{\Gamma,\Delta,\Theta,\Pi,\Upsilon\vdash F}\,\text{cut}$$

Contents

*You don't need to know METAPOST to use this package.
†Feel free to improve! Last update: June 9, 2009.

1 Usage

1.1 Structure of a METAPOST file using drv.mp

Preamble

```
input drv;
verbatimtex %&latex
⟨LaTeX preamble⟩
\begin{document}
etex;
```

Figures

```
⟨optional drv tunings⟩
beginfig(⟨index⟩)
  ⟨judgment & inference declarations⟩
  draw drv_tree;
  ⟨optional extra METAPOST code⟩
endfig;
```

Postamble

```
end
```

For each "`beginfig(`⟨*index*⟩`)`, `endfig;`" pair in a file ⟨*jobname*⟩`.mp`, METAPOST will output an Encapsulated PostScript file ⟨*jobname*⟩`.`⟨*index*⟩.

1.2 Running METAPOST

You have to run

`mpost` ⟨*jobname*⟩`.mp`

at least twice (once more if you use sub-tree delimiters, see § 2.4). On the first run, METAPOST collects the LaTeX code generated by `drv.mp` declaration macros (see § 2) and writes it to the file ⟨*jobname*⟩`-delayed.mp`. On the second run, METAPOST preprocesses the LaTeX code in ⟨*jobname*⟩`-delayed.mp` and typesets the derivation trees.

If you get an error on the first run then it comes from the `drv.mp`/METAPOST code. If you get an error on the second run then it comes from the LaTeX code. In both cases, correct the error (see § A), delete ⟨*jobname*⟩`-delayed.mp` and run "`mpost` ⟨*jobname*⟩`.mp`" twice again (a `makefile` can do that for you).

1.3 LaTeX inclusion commands

Encapsulated PostScript files ⟨*jobname*⟩`.`⟨*index*⟩ generated by METAPOST can be included in LaTeX documents using the `\includegraphics{`⟨*jobname*⟩`.`⟨*index*⟩`}` command from the `graphics.sty` (or `graphicx.sty`) package.

However, I suggest using the following `\drv{`⟨*jobname*⟩`.`⟨*index*⟩`}` command, which is such that the baseline of the included picture coincides with the baseline of the inclusion point.

```
\usepackage{graphicx}
\makeatletter
\def\Gin@def@bp#1\relax#2#3{\gdef#2{#3}}
\newsavebox{\graphicsbox}
\newcommand*{\drv}[1]{%
\sbox{\graphicsbox}{\includegraphics{#1}}%
\raisebox{\Gin@lly bp}%
{\usebox{\graphicsbox}}}
\makeatother
```

The code for `\drv` was suggested by Josselin N on the `fr.comp.text.tex` Usenet group.

2 Judgment & inference declarations

2.1 `jgm` & `nfr`

`jgm` ⟨*numeric*⟩ ⟨*strings list*⟩

⟨*numeric*⟩	judgment index
⟨*strings list*⟩	sub-judgments *math-mode* LaTeX code

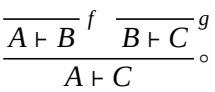
`nfr` ⟨*numeric 1*⟩ `(`⟨*numerics list*⟩`)` `(`⟨*string*⟩`,` ⟨*numeric 2*⟩`)`

⟨*numeric 1*⟩	inference index
⟨*numerics list*⟩	list of premise indices
⟨*string*⟩	inference label *math-mode* LaTeX code
⟨*numeric 2*⟩	inference line style (`0`, `1`, `2`, `3`, `4`, `5` or `6`)

"`jgm` ⟨*numeric*⟩" declares a judgment which index is ⟨*numeric*⟩; "`nfr` ⟨*numeric*⟩" declares an inference which conclusion is the index ⟨*numeric*⟩ judgment. A judgment can be declared before or after the corresponding inference, no matter.

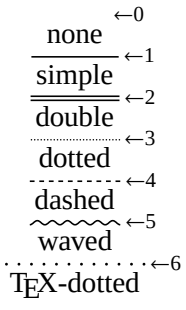
First example

```
beginfig(110)
jgm 0 "A\vdash B";
jgm 1 "B\vdash C";
jgm 2 "A\vdash C";
nfr 0 () ("f", 1);
nfr 1 () ("g", 1);
nfr 2 (0, 1) ("\circ", 1);
draw drv_tree;
endfig;
```



Inference line styles

```
beginfig(111)
jgm 10 "\text{none}";
jgm 11 "\text{simple}";
jgm 12 "\text{double}";
jgm 13 "\text{dotted}";
jgm 14 "\text{dashed}";
jgm 15 "\text{waved}";
jgm 16 "\text{\TeX-dotted}";
nfr 10 () ("\leftarrow 0", 0);
nfr 11 (10) ("\leftarrow 1", 1);
nfr 12 (11) ("\leftarrow 2", 2);
nfr 13 (12) ("\leftarrow 3", 3);
nfr 14 (13) ("\leftarrow 4", 4);
nfr 15 (14) ("\leftarrow 5", 5);
nfr 16 (15) ("\leftarrow 6", 6);
draw drv_tree;
endfig;
```



Declarations order Declarations can occur in any order.

```
beginfig(120)
% preorder declarations
jgm 0 "0";
  jgm 1 "00";
    jgm 2 "000";
    jgm 3 "001";
    jgm 4 "002";
  jgm 5 "01";
    jgm 6 "010";
    jgm 7 "011";
    jgm 8 "012";
  jgm 9 "02";
    jgm 10 "020";
    jgm 11 "021";
    jgm 12 "022";
nfr 0 (1, 5, 9) ("a", 1);
  nfr 1 (2, 3, 4) ("b", 1);
    nfr 2 () ("c", 1);
    nfr 3 () ("d", 1);
    nfr 4 () ("e", 1);
  nfr 5 (6, 7, 8) ("f", 1);
    nfr 6 () ("g", 1);
    nfr 7 () ("h", 1);
    nfr 8 () ("i", 1);
  nfr 9 (10, 11, 12) ("j", 1);
    nfr 10 () ("k", 1);
    nfr 11 () ("l", 1);
    nfr 12 () ("m", 1);
draw drv_tree;
endfig;
```

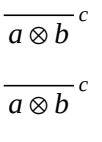
```
beginfig(121)
% postorder declarations
    jgm 0 "000";
    jgm 1 "001";
    jgm 2 "002";
  jgm 3 "00";
    jgm 4 "010";
    jgm 5 "011";
    jgm 6 "012";
  jgm 7 "01";
    jgm 8 "020";
    jgm 9 "021";
    jgm 10 "022";
  jgm 11 "02";
jgm 12 "0";
    nfr 0 () ("a", 1);
    nfr 1 () ("b", 1);
    nfr 2 () ("c", 1);
  nfr 3 (0, 1, 2) ("d", 1);
    nfr 4 () ("e", 1);
    nfr 5 () ("f", 1);
    nfr 6 () ("g", 1);
  nfr 7 (4, 5, 6) ("h", 1);
    nfr 8 () ("i", 1);
    nfr 9 () ("j", 1);
    nfr 10 () ("k", 1);
  nfr 11 (8, 9, 10) ("l", 1);
nfr 12 (3, 7, 11) ("m", 1);
draw drv_tree;
endfig;
```

$$\dfrac{\overline{000}^{\,c}\quad \dfrac{\overline{001}^{\,d}\quad \overline{002}^{\,e}}{00}{}^{\,b}\quad \dfrac{\overline{010}^{\,g}\quad \overline{011}^{\,h}\quad \overline{012}^{\,i}}{01}{}^{\,f}\quad \dfrac{\overline{020}^{\,k}\quad \overline{021}^{\,l}\quad \overline{022}^{\,m}}{02}{}^{\,j}}{0}{}^{\,a} \tag{120}$$

$$\dfrac{\overline{000}^{\,a}\quad \dfrac{\overline{001}^{\,b}\quad \overline{002}^{\,c}}{00}{}^{\,d}\quad \dfrac{\overline{010}^{\,e}\quad \overline{011}^{\,f}\quad \overline{012}^{\,g}}{01}{}^{\,h}\quad \dfrac{\overline{020}^{\,i}\quad \overline{021}^{\,j}\quad \overline{022}^{\,k}}{02}{}^{\,l}}{0}{}^{\,m} \tag{121}$$

Sub-judgments

```
beginfig(130)
jgm 0 "a\otimes b";
resp. jgm 0 "a", "\otimes", "b";
nfr 0 () ("c", 1);
draw drv_tree;
endfig;
```



The outputs induced by

“`jgm 0 "a\otimes b";`” and “`jgm 0 "a", "\otimes", "b";`”

are the same. Using the latter declaration, you can manipulate sub-judgments independently from each-other (see § 4.2 and § 4.3).

Title page derivation tree

```
beginfig(100)
jgm 0 "\Pi\vdash (A\to D)\to E";
jgm 1 "A, \Gamma\vdash B";
jgm 2 "\Delta\vdash C";
jgm 3 "A, \Gamma, \Delta\vdash B\wedge C";
jgm 4 "B\wedge C, \Theta\vdash D";
jgm 5 "A, \Gamma, \Delta, \Theta\vdash D";
jgm 6 "\Gamma, \Delta, \Theta\vdash A\to D";
jgm 7 "E, \Upsilon\vdash F";
jgm 8 "\Gamma, \Delta, \Theta, (A\to D)\to E, \Upsilon\vdash F";
jgm 9 "\Gamma, \Delta, \Theta, \Pi, \Upsilon\vdash F";
nfr 0 () ("\pi", 4);
nfr 1 () ("\gamma", 2);
nfr 2 () ("\delta", 1);
nfr 3 (1, 2) ("\wedge_R", 1);
nfr 4 () ("\theta", 3);
nfr 5 (3, 4) ("\text{cut}", 1);
nfr 6 (5) ("\to_R", 1);
nfr 7 () ("\upsilon", 2);
nfr 8 (6, 7) ("\to_L", 1);
nfr 9 (0, 8) ("\text{cut}", 1);
draw drv_tree;
endfig;
```

2.2 dcl

`dcl` enables the simultaneous declarations of a judgment and of the corresponding inference.

"`dcl` ⟨*numeric 1*⟩ `(`⟨*numerics list*⟩`)` `(`⟨*string*⟩`,` ⟨*numeric 2*⟩`)` ⟨*strings list*⟩`;`"
is a shorthand for:
"`jgm` ⟨*numeric 1*⟩ ⟨*strings list*⟩`;`
`nfr` ⟨*numeric 1*⟩ `(`⟨*numerics list*⟩`)` `(`⟨*string*⟩`,` ⟨*numeric 2*⟩`);`"

```
beginfig(140)
dcl 0 () ("f", 1) "A\vdash B";
dcl 1 () ("g", 1) "B\vdash C";
dcl 2 (0, 1) ("\circ", 1) "A\vdash C";
draw drv_tree;
endfig;
```

$$\dfrac{\dfrac{}{A \vdash B}\,f \quad \dfrac{}{B \vdash C}\,g}{A \vdash C}\,\circ$$

```
beginfig(141)
dcl 0 (1, 5, 9) ("a", 1)     "0";
  dcl 1 (2, 3, 4) ("b", 1)     "00";
    dcl 2 () ("c", 1)            "000";
    dcl 3 () ("d", 1)            "001";
    dcl 4 () ("e", 1)            "002";
  dcl 5 (6, 7, 8) ("f", 1)     "01";
    dcl 6 () ("g", 1)            "010";
    dcl 7 () ("h", 1)            "011";
    dcl 8 () ("i", 1)            "012";
  dcl 9 (10, 11, 12) ("j", 1)  "02";
    dcl 10 () ("k", 1)           "020";
    dcl 11 () ("l", 1)           "021";
    dcl 12 () ("m", 1)           "022";
draw drv_tree;
endfig;
```

$$\dfrac{\dfrac{\dfrac{}{000}\,c \quad \dfrac{}{001}\,d \quad \dfrac{}{002}\,e}{00}\,b \quad \dfrac{\dfrac{}{010}\,g \quad \dfrac{}{011}\,h \quad \dfrac{}{012}\,i}{01}\,f \quad \dfrac{\dfrac{}{020}\,k \quad \dfrac{}{021}\,l \quad \dfrac{}{022}\,m}{02}\,j}{0}\,a$$

2.3 mvd

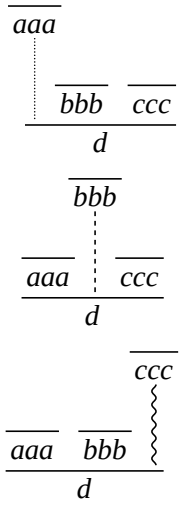
A premise index ⟨*numeric 1*⟩ in an inference declaration can be replaced with

"`mvd` ⟨*numeric 1*⟩ `(`⟨*numeric 2*⟩`,` ⟨*numeric 3*⟩`)`"

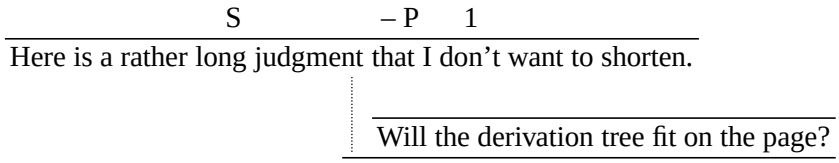
which declares ⟨*numeric 2*⟩ "phantom" inference steps starting from judgment ⟨*numeric 1*⟩. The "phantom" inference steps are intended to be drawn as a path using path-style ⟨*numeric 3*⟩.

`mvd` ⟨*numeric 1*⟩ `(`⟨*numeric 2*⟩`,` ⟨*numeric 3*⟩`)`
⟨*numeric 1*⟩ index of the origin judgment
⟨*numeric 2*⟩ number of steps
⟨*numeric 3*⟩ phantom steps style (`0`, `1`, `2`, `3`, `4`, `5` or `6`)

```
beginfig(150)
jgm 1 "aaa";
jgm 2 "bbb";
jgm 3 "ccc";
jgm 4 "d";
nfr 1 () ("", 1);
nfr 2 () ("", 1);
nfr 3 () ("", 1);
nfr 4 (mvd 1 (2, 3), 2, 3) ("", 1);
resp. nfr 4 (1, mvd 2 (2, 4), 3) ("", 1);
resp. nfr 4 (1, 2, mvd 3 (2, 5)) ("", 1);
draw drv_tree;
endfig;
```




```
beginfig(160)
jgm 0 "\textsc{Size matters -- Part 1}";
jgm 1 "\text{Here is a rather long"&   % string concatenation
      " judgment that I don't want to shorten.}";
jgm 2 "\text{Will the derivation tree fit on the page?}";
jgm 3 "\text{It does.}";
nfr 0 () ("", 0);
nfr 1 (0) ("", 1);
nfr 2 () ("", 1);
nfr 3 (mvd 1 (2, 3), 2) ("", 1);
draw drv_tree;
endfig;
```



2.4 Sub-tree delimiters & labels

- `Nfr` ⟨*numeric 1*⟩ (⟨*numerics list*⟩)
  (⟨*string 1*⟩, ⟨*string 2*⟩, ⟨*string 3*⟩, ⟨*numeric 2*⟩)

⟨*numeric 1*⟩	inference index
⟨*numerics list*⟩	list of premise indices
⟨*string 1*⟩	inference label *math-mode* LaTeX code
⟨*string 2*⟩	left delimiter label *math-mode* LaTeX code
⟨*string 3*⟩	right delimiter label *math-mode* LaTeX code
⟨*numeric 2*⟩	inference line style (`0`, `1`, `2`, `3`, `4`, `5` or `6`)

```
beginfig(170)
jgm 1 "a";
jgm 2 "b";
jgm 3 "c";
jgm 4 "d";
nfr 1 () ("", 1);
nfr 2 () ("", 1);
Nfr 3 (1, 2) ("", "E", "", 1);
Nfr 4 (3) ("", "", "F", 1);
draw drv_tree;
endfig;
```

$$E\left\{\dfrac{\dfrac{a}{c}\quad\dfrac{b}{d}}{}\right\}F$$

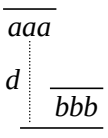
- "`Dcl` ⟨*numeric 1*⟩ (⟨*numerics list*⟩)
  (⟨*string 1*⟩, ⟨*string 2*⟩, ⟨*string 3*⟩, ⟨*numeric 2*⟩) ⟨*strings list*⟩;"
  is a shorthand for:
  "`jgm` ⟨*numeric 1*⟩ ⟨*strings list*⟩;
  `Nfr` ⟨*numeric 1*⟩ (⟨*numerics list*⟩)
  (⟨*string 1*⟩, ⟨*string 2*⟩, ⟨*string 3*⟩, ⟨*numeric 2*⟩);"

```
beginfig(180)
dcl 1 () ("", 1) "a";
Dcl 2 (1) ("", "", "B", 1) "c";
dcl 3 () ("", 1) "d";
Dcl 4 (2, 3) ("", "E", "", 1) "f";
draw drv_tree;
endfig;
```

$$E\left\{\dfrac{\left\{\dfrac{a}{c}\right\}B\quad\dfrac{}{d}}{f}\right.$$

- `Mvd` ⟨*numeric 1*⟩ (⟨*numeric 2*⟩, ⟨*string 1*⟩, ⟨*string 2*⟩, ⟨*numeric 3*⟩)
  - ⟨*numeric 1*⟩ index of the origin judgment
  - ⟨*numeric 2*⟩ number of inference steps
  - ⟨*string 1*⟩ left phantom steps label *math-mode* LaTeX code
  - ⟨*string 2*⟩ right phantom steps label *math-mode* LaTeX code
  - ⟨*numeric 3*⟩ phantom steps style (`0`, `1`, `2`, `3`, `4`, `5` or `6`)

```
beginfig(190)
jgm 1 "aaa";
jgm 2 "bbb";
jgm 3 "ccc";
nfr 1 () ("", 1);
nfr 2 () ("", 1);
nfr 3 (Mvd 1 (2, "d", "", 3), 2) ("", 1);
draw drv_tree;
endfig;
```



3 drv tunings

3.1 drv_font_size

`drv_font_size` ⟨*string*⟩
- ⟨*string*⟩ LaTeX font-size command
  - `"\tiny"`
  - `"\scriptsize"`
  - `"\footnotesize"`
  - `"\small"`
  - `"\normalsize"` ∗ *default* ∗
  - `"\large"`
  - etc.

`"\tiny"`

`"\small"`

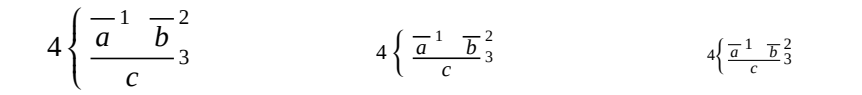
`"\Large"`

3.2 drv_math_style

`drv_math_style` (⟨*suffix*⟩, ⟨*string*⟩)
- ⟨*suffix*⟩ component type (`drv`, `jdg`, `ilb`, `dlb` or `plb`)
  - `drv` derivation tree ∗ *default style:* *"\displaystyle"* ∗
  - `jdg` judgment ∗ *default style:* *"\textstyle"* ∗
  - `ilb` inference label ∗ *default style:* *"\scriptstyle"* ∗
  - `dlb` delimiter label ∗ *default style:* *"\textstyle"* ∗
  - `plb` phantom steps label ∗ *default style:* *"\textstyle"* ∗
- ⟨*string*⟩ LaTeX math-style command

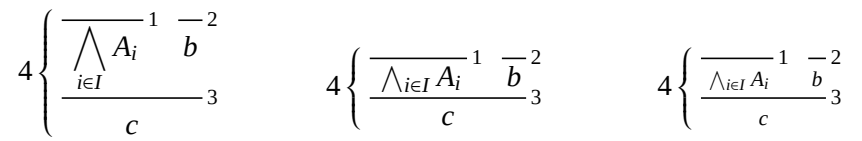
Examples “drv_math_style (drv, —);”

"\displaystyle" "\textstyle" "\scriptstyle"



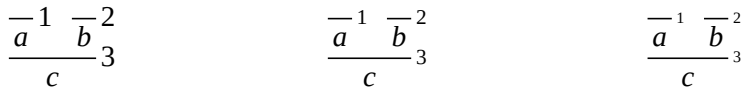
Examples “drv_math_style (jdg, —);”

"\displaystyle" "\textstyle" "\scriptstyle"



Examples “drv_math_style (ilb, —);”

"\textstyle" "\scriptstyle" "\scriptscriptstyle"



3.3 drv_scale

drv_scale (⟨*suffix*⟩, ⟨*numeric*⟩)

⟨*suffix*⟩ scale type (clr, prm, jdg or ilb)

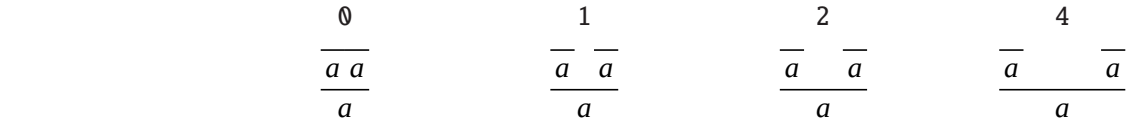
- clr (see examples) ∗ *default scale: 1* ∗
- prm (see examples) ∗ *default scale: 1* ∗
- jdg (see examples) ∗ *default scale: 1* ∗
- ilb (see examples) ∗ *default scale: 1* ∗

⟨*numeric*⟩ scale value

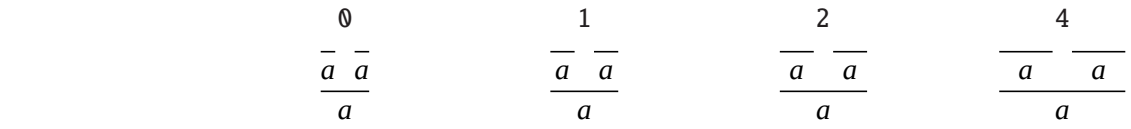
Examples “drv_scale (clr, —);”

0 1 2 4

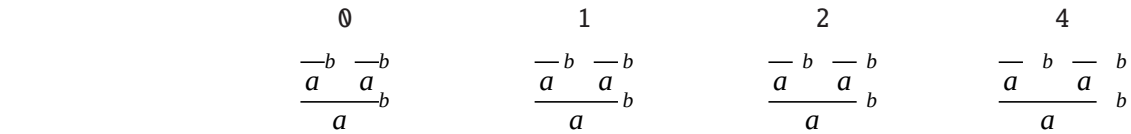
Examples “drv_scale (prm, —);”



Examples “drv_scale (jgm, —);”

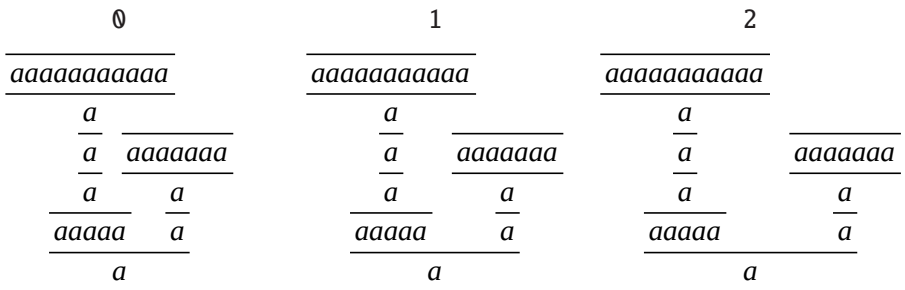


Examples “drv_scale (ilb, —);”



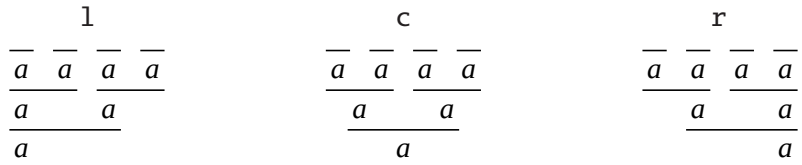
3.4 drv_junction_style

drv_junction_style ⟨*numeric*⟩
⟨*numeric*⟩ junction style (0, 1 or 2)
0 “fully-interlacing”
1 “semi-interlacing” ∗ *default* ∗
2 “non-interlacing”



3.5 drv_alignment_style

drv_alignment_style ⟨*suffix*⟩
⟨*suffix*⟩ alignment style (l, c or r)
l left
c centered ∗ *default* ∗
r right

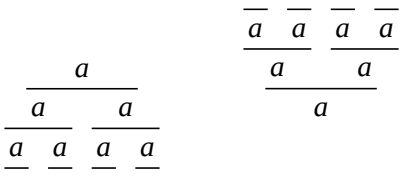


3.6 drv_labels_position

`drv_labels_position` ⟨*suffix*⟩
⟨*suffix*⟩ position (`l` or `r`)
`l` left
`r` right ∗ *default* ∗

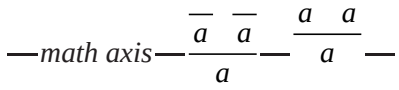
3.7 drv_roots_position

`drv_roots_position` ⟨*suffix*⟩
⟨*suffix*⟩ position (`t` or `b`)
`t` top
`b` bottom ∗ *default* ∗



3.8 drv_axis_reference

`drv_axis_reference` ⟨*suffix*⟩
⟨*suffix*⟩ reference type (`iln` or `jdg`)
`iln` root inference line ∗ *default* ∗
`jdg` root judgment



Notice that `drv_axis_reference` is irrelevant if you don't use the `\drv` inclusion command (see § 1.3).

3.9 drv_left_delimiter & drv_right_delimiter

`drv_left_delimiter` ⟨*string*⟩
⟨*string*⟩ left delimiter math-mode LaTeX code

`"("`	(
`"["`	[
`"\lbrace"`	{ ∗ *default* ∗
`"\langle"`	⟨
etc.	

`drv_right_delimiter` ⟨*string*⟩
⟨*string*⟩ right delimiter math-mode LaTeX code
`")"`)
`"]"`]
`"\rbrace"` } ∗ *default* ∗
`"\rangle"` ⟩
etc.

Examples "`drv_left_delimiter —;`"

`"("` $E\left(\dfrac{\overline{\dfrac{a}{c}}\,B\;\overline{d}}{f}\right.$ `"\lfloor"` $E\left\lfloor\dfrac{\overline{\dfrac{a}{c}}\,B\;\overline{d}}{f}\right.$ `"."` $E\,\dfrac{\overline{\dfrac{a}{c}}\,B\;\overline{d}}{f}$

Examples "`drv_right_delimiter —;`"

`"\rangle"` $E\left\{\dfrac{\overline{\dfrac{a}{c}}\rangle B\;\overline{d}}{f}\right.$ `"\uparrow"` $E\left\{\dfrac{\overline{\dfrac{a}{c}}\uparrow B\;\overline{d}}{f}\right.$ `"."` $E\left\{\dfrac{\overline{\dfrac{a}{c}}\;B\;\overline{d}}{f}\right.$

3.10 drv_fraction_mode

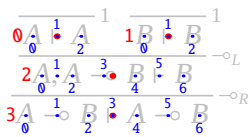
`drv_fraction_mode` ⟨*suffix*⟩
⟨*suffix*⟩ fraction-mode status (`on` or `off`)
`on` ∗ *default* ∗
`off`
When derivation trees are typeset in "fraction mode", heights of leaf-judgments above which there is no inference line are ignored. Warning: this mode may cause overlaps in conjunction with interlacing junction-styles (`0` and `1`).

`on` $\dfrac{\dfrac{\overbrace{A,\Gamma\vdash B}\quad\overline{B,\Delta\vdash C}}{A,\Gamma,\Delta\vdash C}\,\text{cut}}{\Gamma,\Delta\vdash A\to C}\,{\to_R}$ `off` $\dfrac{\dfrac{\overbrace{A,\Gamma\vdash B}\quad\overline{B,\Delta\vdash C}}{A,\Gamma,\Delta\vdash C}\,\text{cut}}{\Gamma,\Delta\vdash A\to C}\,{\to_R}$ typeset as $\dfrac{\dfrac{\overbrace{A,\Gamma\vdash B}\quad\overline{B,\Delta\vdash C}}{A,\Gamma,\Delta\vdash C}\,\text{cut}}{\Gamma,\Delta\vdash A\to C}\,{\to_R}$

3.11 drv_proof_mode

`drv_proof_mode` ⟨*suffix*⟩
⟨*suffix*⟩ proof-mode status (`on` or `off`)
`on`
`off` ∗ *default* ∗

`on`



`off`

$$\dfrac{\dfrac{}{A \vdash A}\,1 \quad \dfrac{}{B \vdash B}\,1}{\dfrac{A, A \multimap B \vdash B}{A \multimap B \vdash A \multimap B}\,\multimap_R}\,\multimap_L$$

Red numbers (resp. dots): judgment indices (resp. central points).
Blue numbers (resp. dots): sub-judgment indices (resp. central points).

4 Low level features

4.1 Low level inference declarations

- `NFR` ⟨*numeric 1*⟩ (⟨*numerics list*⟩)
  (⟨*string 1*⟩, ⟨*string 2*⟩, ⟨*string 3*⟩, ⟨*string 4*⟩)
  (⟨*numeric 2*⟩, ⟨*suffix*⟩, ⟨*numeric 3*⟩)

⟨*numeric 1*⟩	inference index
⟨*numerics list*⟩	list of premise indices
⟨*string 1*⟩	left inference label *math-mode* LaTeX code
⟨*string 2*⟩	right inference label *math-mode* LaTeX code
⟨*string 3*⟩	left delimiter label *math-mode* LaTeX code
⟨*string 4*⟩	right delimiter label *math-mode* LaTeX code
⟨*numeric 2*⟩	junction style (`0`, `1`, `2` or `3`)
	`0` fully-interlacing
	`1` semi-interlacing
	`2` non-interlacing
	`3` user specified (tricky)
⟨*suffix*⟩	alignment style (`l`, `c`, `r` or `u`)
	`l` left
	`c` centered
	`r` right
	`u` user specified (tricky)
⟨*numeric 3*⟩	inference line style (`0`, `1`, `2`, `3`, `4`, `5` or `6`)

```
beginfig(370)
jgm 0 "a";
NFR 0 () ("1", "2", "3", "4", 1, c, 1);
draw drv_tree;
endfig;
```

$$3\left\{\,{}^{1}\overline{a}^{\,2}\,\right\}4$$

- “DCL ⟨*numeric 1*⟩ (⟨*numerics list*⟩)
  (⟨*string 1*⟩, ⟨*string 2*⟩, ⟨*string 3*⟩, ⟨*string 4*⟩)
  (⟨*numeric 2*⟩, ⟨*suffix*⟩, ⟨*numeric 3*⟩) ⟨*strings list*⟩;”
  is a shorthand for:
  “jgm ⟨*numeric 1*⟩ ⟨*strings list*⟩;
  NFR ⟨*numeric 1*⟩ (⟨*numerics list*⟩)
  (⟨*string 1*⟩, ⟨*string 2*⟩, ⟨*string 3*⟩, ⟨*string 4*⟩)
  (⟨*numeric 2*⟩, ⟨*suffix*⟩, ⟨*numeric 3*⟩);”

```
beginfig(380)
DCL 0 ()     ("1", "", "", "", 1, c, 1) "a";
DCL 1 ()     ("", "2", "", "", 1, c, 1) "b";
DCL 2 (0, 1) ("", "", "3", "", 1, c, 1) "c";
DCL 3 (2)    ("", "", "", "4", 1, c, 1) "d";
draw drv_tree;
endfig;
```

$$3\left\{\begin{array}{c}{}^{1}\overline{a}\quad\overline{b}^{\,2}\\ \hline c\\ \hline d\end{array}\right\}4$$

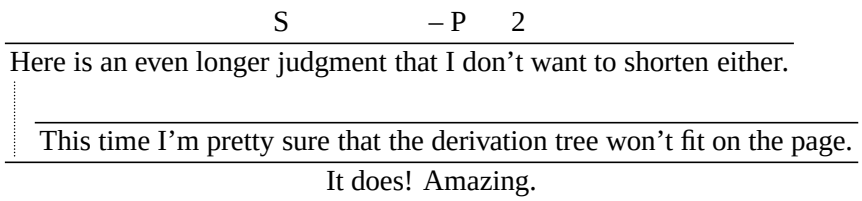
- MVD ⟨*numeric 1*⟩ (⟨*numeric 2*⟩, ⟨*string 1*⟩, ⟨*string 2*⟩)
  (⟨*suffix*⟩, ⟨*numeric 3*⟩)

⟨*numeric 1*⟩	index of the origin judgment
⟨*numeric 2*⟩	number of inference steps
⟨*string 1*⟩	left phantom steps label *math-mode* LaTeX code
⟨*string 2*⟩	right phantom steps label *math-mode* LaTeX code
⟨*suffix*⟩	alignment style (l, c, r or u)
⟨*numeric 3*⟩	phantom steps style (0, 1, 2, 3, 4, 5 or 6)

```
beginfig(390)
jgm 0 "aaaaa";
jgm 1 "aaa";
jgm 2 "a";
jgm 3 "aaaaaaaa";
jgm 4 "a";
nfr 0 () ("", 1);
nfr 1 () ("", 1);
nfr 2 (MVD 0 (2, "", "", l, 3), 1) ("", 1);
nfr 3 () ("", 1);
nfr 4 (2, MVD 3 (5, "", "", r, 4)) ("", 1);
draw drv_tree;
endfig;
```

$$\dfrac{\dfrac{\dfrac{\dfrac{}{aaaaaaaa}}{aaaaa}}{aaa}}{a}\qquad a$$

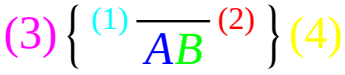
```
beginfig(391)
jgm 0 "\textsc{Size matters -- Part 2}";
jgm 1 "\text{Here is an even longer judgment"&
      " that I don't want to shorten either.}";
jgm 2 "\text{This time I'm pretty sure that the"&
      " derivation tree won't fit on the page.}";
jgm 3 "\text{It does! Amazing.}";
nfr 0 () ("", 0);
nfr 1 (0) ("", 1);
nfr 2 () ("", 1);
nfr 3 (MVD 1 (2, "", "", 1, 3), 2) ("", 1);
draw drv_tree;
endfig;
```



4.2 Inside derivation trees

Components and central points

```
beginfig(400) % components
jgm 7 "A", "B";
NFR 7 () ("(1)", "(2)", "(3)", "(4)", 0, c, 1);
drv_freeze; % usually called by drv_tree
draw sbj[7][0] withcolor (0, 0, 1); % sub-judgment           A
draw sbj[7][1] withcolor (0, 1, 0); % sub-judgment           B
draw l_ilb[7]  withcolor (0, 1, 1); % left inference label  (1)
draw r_ilb[7]  withcolor (1, 0, 0); % right inference label (2)
draw l_dlb[7]  withcolor (1, 0, 1); % left delimiter label  (3)
draw r_dlb[7]  withcolor (1, 1, 0); % right delimiter label (4)
draw l_dlm[7];                      % left delimiter
draw iln[7];                        % inference line
draw r_dlm[7];                      % right delimiter
endfig;
```

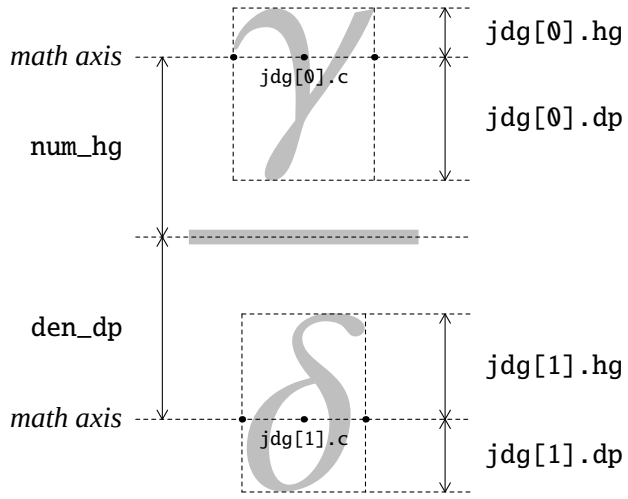


```
beginfig(401) % central points
jgm 7 "A", "B";
NFR 7 () ("(1)", "(2)", "(3)", "(4)", 0, c, 1);
draw drv_tree withcolor .75*background;
pickup pencircle scaled 2.5;
draw sbj[7][0].c withcolor (0, 0, 1);
draw sbj[7][1].c withcolor (0, 1, 0);
draw l_ilb[7].c  withcolor (0, 1, 1);
draw r_ilb[7].c  withcolor (1, 0, 0);
draw l_dlb[7].c  withcolor (1, 0, 1);
draw r_dlb[7].c  withcolor (1, 1, 0);
endfig;
```

Dimensions

```
beginfig(410)
dcl 0 ()  ("", 0) "\gamma";
dcl 1 (0) ("", 1) "\delta";
draw drv_tree;
endfig;
```

(The picture below may look weird if you don't use scalable fonts.)



Depths (`den_dp`, `jdg[0].dp` and `jdg[1].dp`) are negative; heights (`num_hg`, `jdg[0].hg` and `jdg[1].hg`) are positive.

drv.mp typesets derivation trees essentially according to the algorithm for typesetting fractions described in Appendix G of the TEXbook (see [2, 3]).

drv_axis drv_axis (⟨*suffix*⟩, ⟨*numeric*⟩)
⟨*suffix*⟩ reference type (iln, jdg or dlm)
iln inference line
jdg judgment
dlm delimiter
⟨*numeric*⟩ reference judgment index

```
beginfig(420)
dcl 1 () ("", 1) "a";
Dcl 2 (1) ("", "{}", "{}", 1) "b";
draw drv_tree;
drv_axis (iln, 1);
resp. drv_axis (jdg, 2);
resp. drv_axis (dlm, 2);
endfig;
```

$$\text{—}\textit{math axis}\text{—}\ \left\{\frac{\overline{}}{\overline{a}\ \overline{b}}\right\}\ \text{—}\ \left\{\frac{\overline{}}{\overline{a}\ \overline{b}}\right\}\ \text{—}\ \left\{\frac{\overline{}}{\overline{a}\ \overline{b}}\right\}\ \text{—}\textit{math axis}\text{—}$$

4.3 Basic METAPOST use

"User specified" junction style

```
beginfig(430)
jgm 0 "{\cdot}";
jgm 1 "{\cdot}";
jgm 2 "\text{You may check that the distance"&
      " between the two dots above is 5cm.}";
nfr 0 () ("", 1);
nfr 1 () ("", 1);
NFR 2 (0, 1) ("", "", "", "", 3, c, 1); % caution: 3
xpart jdg[1].c=xpart jdg[0].c+5cm;
draw drv_tree;
endfig;
```

$$\frac{\overline{}}{\cdot}\qquad\qquad\frac{\overline{}}{\cdot}$$

You may check that the distance between the two dots above is 5cm.

“User specified” alignment style

```
beginfig(440)                                          % "\vdash":
jgm 0 "B, A, \Gamma", "\vdash", "C";                   % sbj[0][1]
jgm 1 "A, \Gamma", "\vdash", "B\multimap C";           % sbj[1][1]
jgm 2 "\Gamma", "\vdash", "A\multimap(B\multimap C)"; % sbj[2][1]
nfr 0 ()  ("", 0);
NFR 1 (0) ("", "\multimap_{R}", "", "", 0, u, 1); % caution: u
NFR 2 (1) ("", "\multimap_{R}", "", "", 0, u, 1); % caution: u
xpart sbj[0][1].c=xpart sbj[1][1].c=xpart sbj[2][1].c;
draw drv_tree;
endfig;
```

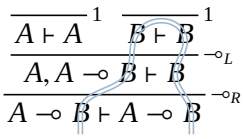
$$\dfrac{\dfrac{B, A, \Gamma \vdash C}{A, \Gamma \vdash B \multimap C}\,{\multimap}_R}{\Gamma \vdash A \multimap (B \multimap C)}\,{\multimap}_R$$

drv_styled

⟨*path*⟩ drv_styled ⟨*numeric*⟩

⟨*path*⟩	METAPOST path expression
⟨*numeric*⟩	path style (0, 1, 2, 3, 4, 5 or 6)

```
beginfig(450)
jgm 4 "A", "\vdash", "A";
jgm 5 "B", "\vdash", "B";
jgm 6 "A", ",", "A", "\multimap", "B", "\vdash", "B";
jgm 7 "A", "\multimap", "B", "\vdash", "A", "\multimap", "B";
nfr 4 () ("1", 1);
nfr 5 () ("1", 1);
nfr 6 (4, 5) ("\multimap_{L}", 1);
nfr 7 (6) ("\multimap_{R}", 1);
draw drv_tree;
draw (sbj[7][2].c shifted (0, -num_hg) ..
      sbj[7][2].c {up} ..
      sbj[6][4].c ..
      sbj[5][0].c .. tension 1.05 ..
      sbj[5][2].c ..
      sbj[6][6].c ..
      sbj[7][6].c {down} ..
      sbj[7][6].c shifted (0, -num_hg))
  drv_styled 2 withcolor (159, 182, 205)/255; % slategray3
endfig;
```



References

[1] John D. Hobby. *A User's Manual for METAPOST*, 2009. http://tug.org/metapost.html

[2] Bogusław Jackowski. *Appendix G illuminated*. *TUGboat*, 27(1):83–90, 2006. http://www.tug.org/TUGboat/Articles/tb27-1/

[3] Donald E. Knuth. *The TEXbook*. Addison-Wesley, 1984.

A Debugging

Recall that you have to run "`mpost` ⟨*jobname*⟩`.mp`" *at least twice* (once more if you use sub-tree delimiters). If you get an error on the first run then it comes from the `drv.mp`/METAPOST code. If you get an error on the second run then it comes from the LaTeX code.

Error on the first run METAPOST behaves essentially as TeX/LaTeX when it finds and error (see [1, Debugging]): it stops, "explains" the error in some way (look for the line starting with an exclamation mark), shows some lines of context, and asks you what to do next (answer `x` to terminate the run). If you're lucky, the error comes from an inconsistency that `drv.mp` can detect: in such a case the explanation should be quite understandable (see below).

```
beginfig(460)
jgm 0 "A\vdash B";
jgm 1 "B\vdash C";
jgm 2 "A\vdash C";
jgm 3 "C\vdash D";
jgm 4 "A\vdash D";
nfr 0 () ("f", 1);
nfr 1 () ("g", 1);
nfr 2 (0, 1) ("\circ", 1);
nfr 3 () ("h", 1);
nfr 4 (0, 3) ("\circ", 1);
draw drv_tree;
endfig;
```

METAPOST error message.

```
! drv (fig. 460): 0 has been used
already as a premise index
(inference 2).
⟨error context⟩
l.60 nfr 4 (0, 3) ("\circ", 1)
                                ;
?
```

Error on the second run METAPOST fails to preprocess the LaTeX code in ⟨*jobname*⟩`-delayed.mp` and suggests that you "`see mpxerr.log`", which is a regular LaTeX log-file. This file tells you which part of the LaTeX code is faulty but unfortunately not where to find it in ⟨*jobname*⟩`.mp`.

B Related packages

- `bussproofs.sty` (Samuel R. B);
- `proof.sty` (Makoto T);
- `prooftree.sty` (Paul T);
- `virginialake.sty` (Alessio G).

Some of these packages are described on Peter S 's LaTeX for Logicians web-page.