The pgfmore package

Frédéric Boulanger
Frederic.Boulanger@supelec.fr

March 17, 2004

1 Introduction

pgfmore is a LaTeX package that adds support for distinct fill, stroke and text colors to the excellent pgf package. It also makes arguments between square brackets really optional and node definitions global (even inside a pgfscope environment).
pgfmore defines new commands to ease the use of nodes: you can draw bows between nodes or loops on a node, put labels on them and so on. Everything in pgfmore relies on pgf only (and on a standard LaTeX installation). This is not really a documentation (look at the comments in the source code for details), but a kind of "show room" for pgfmore.

2 Unit length commands

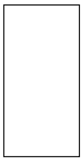
The three following commands allow to set the length unit for PGF pictures. `\pgfsetxunit` sets the unit for the horizontal axis, `\pgfsetyunit` sets the unit for the vertical axis, and `\pgfsetunit` sets both PGF units.

`\pgfsetunit{`*length*`}`
set both horizontal and vertical unit lengths to *length*

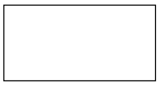
`\pgfsetxunit{`*length*`}`
set the horizontal unit length to *length*

`\pgfsetyunit{`*length*`}`
set the vertical unit length to *length*

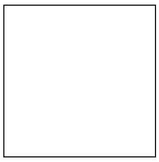
```
\pgfsetxunit{1cm}
\pgfsetyunit{2cm}
\pgfrect{\pgfxy(0,0)}{\pgfxy(1,1)}
```



```
\pgfsetxunit{2cm}
\pgfsetyunit{1cm}
\pgfrect{\pgfxy(0,0)}{\pgfxy(1,1)}
```



```
\pgfsetunit{1cm}
\pgfrect{\pgfxy(0,0)}{\pgfxy(2,2)}
```

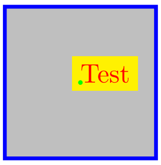


3 Colors

pgfmore defines four colors: the stroke color is used for the “pen” color when drawing paths, the fill color is used when filling figures, the text color is used for text strings, and the back colour is used for text background. `\pgfsetcolor` allows to set one of these colors. After the name of the color to set, `\pgfsetcolor` accepts any color specification that the color package accepts.

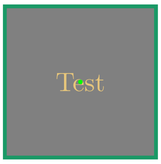
`\pgfsetcolor{`*type*`}`<*color spec*>
set pgfmore’s *type* color to <*color spec*>.

```
\pgfsetcolor{stroke}{blue}
\pgfsetcolor{fill}{lightgray}
\pgfsetcolor{text}{red}
\pgfsetcolor{back}{yellow}
\pgfsetlinewidth{0.5mm}
\pgfrect[fillstroke]{\pgfxy(0,0)}{\pgfxy(2,2)}
\pgfputat{\pgfxy(1,1)}{\pgfbox{Test}[opaque]}
```



where the green dot shows the postion of the $(1, 1)$ point.

```
\pgfsetcolor{stroke}[rgb]{0.1,0.6,0.4}
\pgfsetcolor{fill}[gray]{0.5}
\pgfsetcolor{text}[cmyk]{0.1,0.2,0.6,0.0}
\pgfsetlinewidth{0.5mm}
\pgfrect[fillstroke]{\pgfxy(0,0)}{\pgfxy(2,2)}
\pgfputat{\pgfxy(1,1)}{\pgfbox[center,center]{Test}}
```



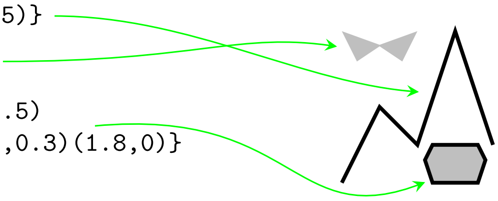
where the green dot shows the postion of the $(1, 1)$ point.

4 Polygons

`\pgfxypolyline` builds a path that connects points with straight lines. According to the first optional argument, the line is just added to the current path (virtual), stroked (stroke), filled (after closing the current path, fill), or stroked and then filled (fillstroke).

`\pgfxypolyline[`*mode*`]{`*coordinate list*`}`
build a broken line through all the points in *coordinate list* and stroke and or fill it according to *mode*.

```
\pgfsetcolor{stroke}{black}
\pgfsetcolor{fill}{lightgray}
\pgfsetlinewidth{0.5mm}
\pgfxypolyline{(0,0)(0.5,1)(1,0.5)(1.5,2)(2,0.5)}
\pgfxypolyline[fill]{(0,2)(0.2,1.6)(0.5,1.8)
                     (0.8,1.6)(1,2)(0.5,1.82)}
\pgfxypolyline[virtual]{(1.2,0)(1.1,0.3)(1.2,0.5)
                        (1.5,0.5)(1.8,0.5)(1.9,0.3)(1.8,0)}
\pgfclosepath\pgffillstroke
```



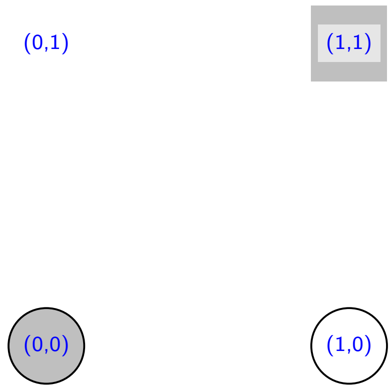
the green arrows show the result of the `\pgfxypolyline` commands.

5 Node commands

pgfmore redefines node commands so that their optional argument defaults to “stroke”. A new `\pgfnodetext` command associates text to a node. This text is drawn on a background rectangle (its color is the “back” PGF color) if the “opaque” parameter is present. The text is drawn in the “text” PGF color.

`\pgfnodetext{`*node name*`}{`*text*`}[`*opaque*`]`
associate *text* to node *node name*. Draw the text in an opaque box if `[`*opaque*`]` is present.

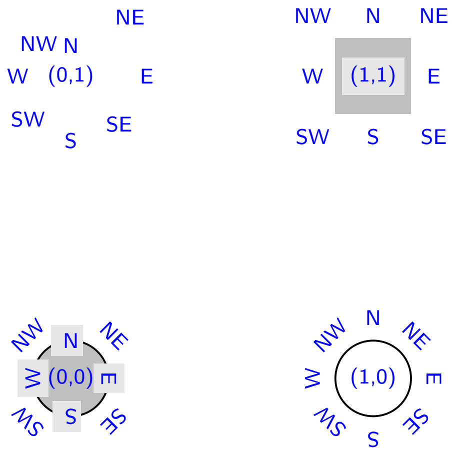
```
\footnotesize\sffamily
\pgfsetcolor{stroke}{black}
\pgfsetcolor{fill}{lightgray}
\pgfsetcolor{text}{blue}
\pgfsetcolor{back}[gray]{0.9}
\pgfsetlinewidth{0.25mm}
\pgfnodecircle{00}[fillstroke]{\pgfxy(0,0)}{5mm}
\pgfnodetext{00}{(0,0)}
\pgfnodecircle{01}[virtual]{\pgfxy(0,1)}{5mm}
\pgfnodetext{01}{(0,1)}
\pgfnodecircle{10}[stroke]{\pgfxy(1,0)}{5mm}
\pgfnodetext{10}{(1,0)}
\pgfnoderect{11}[fill]{\pgfxy(1,1)}{\pgfpoint{1cm}{1cm}}
\pgfnodetext{11}{(1,1)}[opaque]
```



`\pgfbordernodetext` puts text at the border of a node. The text position is specified as a geographical direction from the position of the node. The text is drawn on a rectangular box if the "options" parameter contains "o". The text is rotated to be perpendicular to the direction if the "options" parameter contains "r".

`\pgfbordernodetext[`*offset*`]{`*node name*`}{`*direction*`}[`*options*`]{`*text*`}`
draw *text* at the *direction* border of node *node name*. A positive *offset* puts the text farther from the center of the node. The text in drawn over an opaque box if *options* contains `o`, it is drawn perpendicularly to *direction* if *options* contains `r`.

```
[...]
\pgfbordernodetext{00}{north}[or]{N}
\pgfbordernodetext[3mm]{00}{nw}[r]{NW}
\pgfbordernodetext{00}{west}[or]{W}
\pgfbordernodetext[3mm]{00}{sw}[r]{SW}
[...]
\pgfbordernodetext[-1mm]{01}{north}{N}
\pgfbordernodetext[1mm]{01}{nw}{NW}
\pgfbordernodetext[2mm]{01}{west}{W}
\pgfbordernodetext[3mm]{01}{sw}{SW}
\pgfbordernodetext[3.5mm]{01}{south}{S}
\pgfbordernodetext[4mm]{01}{se}{SE}
\pgfbordernodetext[5mm]{01}{east}{E}
\pgfbordernodetext[6mm]{01}{ne}{NE}
%
\pgfbordernodetext[3mm]{10}{north}[r]{N}
[...]
\pgfbordernodetext[3mm]{10}{ne}[r]{NE}
%
\pgfbordernodetext[3mm]{11}{north}{N}
[...]
\pgfbordernodetext[3mm]{11}{ne}{NE}
```

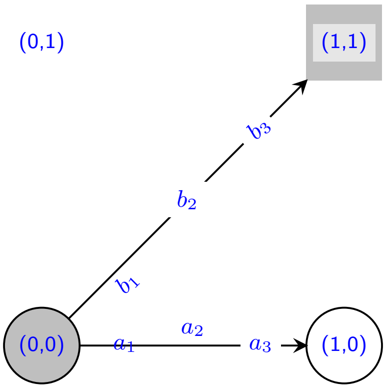


6 Connection labels

`\pgfedgelabel` puts a label on a straight connexion between two nodes. With "o" in the "options" parameters, the text is drawn on a rectangle filled with the back PGF color. With the "r" option, the text is rotated to be aligned with the edge between the nodes. The second optional parameter (between the node names) gives the fraction of the way between the two nodes at which the label will be put.

`\pgfedgelabel[`*options*`]{`*node 1*`}[`*fraction*`]{`*node 2*`}[`*offset*`]{`*text*`}`
draw *text* as the label of an edge between nodes *node 1* and *node 2*. The center of the text is put halfway between the nodes unless *fraction* specifies another place. The text is drawn over an opaque box if *options* contains `o`, it is drawn in the direction of the edge if *options* contains `r`.

```
[...]
\small
\pgfsetcolor{back}{white}
\pgfsetendarrow{\pgfarrowsingle}
\pgfnodeconnline{00}{10}
\pgfedgelabel{00}{10}[1.5ex]{\(a_{2}\)}
\pgfedgelabel{00}[0.2]{10}{\(a_{1}\)}
\pgfedgelabel[o]{00}[0.8]{10}{\(a_{3}\)}
\pgfnodeconnline{00}{11}
\pgfedgelabel[o]{00}{11}{\(b_{2}\)}
\pgfedgelabel[r]{00}[0.2]{11}[-1.5ex]{\(b_{1}\)}
\pgfedgelabel[or]{00}[0.8]{11}{\(b_{3}\)}
```

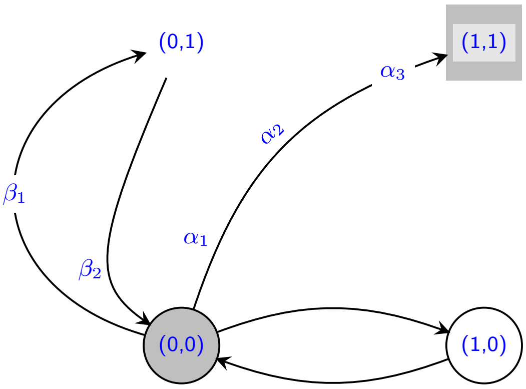


`\pgfnodebow` draws a bow between two node. The bow deviates according to the third argument: when the third argument is positive, the bow deviate to the left (when looking at the target node from the first node); when the third argument is negative, the bow deviates to the right. The absolute value of the third argument give the amount of the deviation with respect to the distance between the nodes.
`\pgfnodebowlabel` puts a label (opaque and/or rotated) on a bow between two nodes.

`\pgfnodebow[`*pair of fractions*`]{`*node 1*`}{`*node 2*`}{`*deviation*`}`
draw a bow from node *node 1* to *node 2*. Deviate as much as *deviation* times the distance between the nodes, to the left when *deviation* is positive. The control points of the Bezier curve of the bow are placed at 0.4 and 0.6 times the distance between the nodes. These fractions can be changed with the first optional argument.

`\pgfnodebow[`*pair of fractions*`]{`*node 1*`}{`*node 2*`}{`*deviation*`}[`*options*`]{`*text*`}`
draw *text* as the label of a bow drawn by `\pgfnodebow`. The text is drawn over an opaque box if *options* contains `o`, it is drawn in the direction of the edge if *options* contains `r`.

```
[...]
\small
\pgfsetcolor{back}{white}
\pgfsetendarrow{\pgfarrowsingle}
\pgfnodebow{00}{11}{0.2}
\pgfnodebowlabel{00}{11}{0.2}[r]{\(\alpha_{2}\)}
\pgfnodebowlabel[0.2,0.2]{00}{11}{0.15}{\(\alpha_{1}\)}
\pgfnodebowlabel[0.8,0.8]{00}{11}{0.1}[o]{\(\alpha_{3}\)}
\pgfnodebow[0.2,0.8]{00}{01}{0.7}
\pgfnodebowlabel[0.2,0.8]{00}{01}{0.55}[o]{\(\beta_{1}\)}
\pgfnodebow[0.7,0.8]{01}{00}{-0.3}
\pgfnodebowlabel[0.7,0.8]{01}{00}{-0.3}{\(\beta_{2}\)}
```



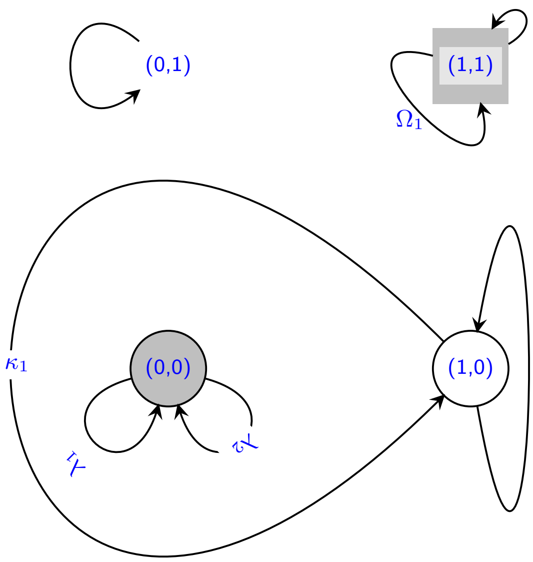
7 Node loops

`\pgfnodeconnloop` connects a node to itself with a loop. A points gives the direction of the loop relative to the center of the node. An angle specifies the aperture of the loop.
`\pgfnodelooplabel` puts a label on a node loop. The label is drawn on an opaque rectangle if "options" contains "o". It is rotated to be perpendicular to the direction of the loop unless the "options" contains "n".

`\pgfnodeconnloop{`*node name*`}{`*angle*`}(`x`,`y`)`
draw a loop on node *node name* in the (x, y) direction. The loop is opened by *angle* on each side.

`\pgfnodelooplabel{`*node name*`}[`*fraction*`](`x`,`y`)[`*options*`]{`*text*`}`
draw *text* as the label of a loop drawn by `\pgfnodeconnloop`. The text is drawn over an opaque box if *options* contains `o`, it is drawn horizontally only if *options* contains `n` (the default is `r`, a rotated label). *fraction* is the fraction of the distance between the center of the node and the relative (x, y) point at which to put the center of the label.

```
[...]
\small
\pgfsetcolor{back}{white}
\pgfsetendarrow{\pgfarrowsingle}
\pgfnodeconnloop{00}{30}(-0.3,-0.3)
\pgfnodelooplabel{00}(-0.3,-0.3){\(\lambda_{1}\)}
\pgfnodeconnloop{00}{-30}(0.3,-0.3)
\pgfnodelooplabel{00}[0.8](0.3,-0.3)[o]{\(\lambda_{2}\)}
\pgfnodeconnloop{11}{15}(0.2,0.2)
\pgfnodeconnloop{11}{60}(-0.2,-0.2)
\pgfnodelooplabel{11}(-0.2,-0.2)[n]{\(\Omega_{1}\)}
\pgfnodeconnloop{01}{40}(-0.4,0)
\pgfnodeconnloop{10}{45}(-2.0,0)
\pgfnodelooplabel{00}[0.25](-2.0,0)[on]{\(\kappa_{1}\)}
\pgfnodeconnloop{10}{80}(0.25,0)
```


(0,1)
(1,1)
Ω_1
κ_1
(0,0)
(1,0)
λ_1
λ_2