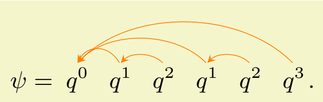
The pstring package

William Blum

April 29, 2008

Abstract

This is a short documentation for the pstring package for Latex. This package allows you to typeset justified sequences, also called pointing strings. The latest version of the package can be downloaded from `http://william.famille-blum.org/software/latex/index.html`.


$\psi = q^0 \quad q^1 \quad q^2 \quad q^1 \quad q^2 \quad q^3 .$

```
\Pstr[0.65cm]{ \psi = (n){q^0} \ (n4-n,60){q^1} \ (n5-n4){q^2}
\ (n8-n){q^1} \ (n9-n8){q^2} \ (n10-n){q^3} }.
```

1 Use and syntax

From version 0.3, pstring can use either PSTricks or PGF as a rendering engine. To specify which one you want, use the option `pstricks` or `pgf` when you load the pstring package. For instance to use PGF you must type:

```
\usepackage{pstring}[pgf]
```

If no option is specified then by default `pgf` is selected if you are running pdflatex and `pstricks` is selected if you are running latex.

You can specifies other options with the following commands:

`\pstrSetLabelStyle`{style}

Change the style used to typeset labels on arrows.

`\pstrSetArrowColor`{color}

Change the color used to typeset arrows.

`\pstrSetArrowLineWidth`{size}

Change the arrows line width.

`\pstrSetArrowLineStyle`{style}

Change the arrows line style. Valid styles are: `none`, `dashed`, `dotted` and `solid`.

The default options are as follows:

```
\pstrSetLabelStyle{\color{blue} \tiny}
\pstrSetArrowColor{orange}
\pstrSetArrowLineWidth{0.3pt}
\pstrSetArrowLineStyle{solid}
```

1.1 The succinct syntax

The syntax for creating a pointing string is as follows:

`\Pstr`[raiseamount][nodesep]{stringspecification}

- `raiseamount` (optional) is the size by which you want to raise the box containing the sequence. In PSTricks mode, the bounding box is not computed automatically therefore it is necessary to specify this value by hand. Using the PGF mode however, this value is ignored since the bounding box is computed automatically,
- `nodesep` (optional) is the vertical distance between the nodes and the links.
- `stringspecification` contains commands describing the nodes of the string. This parameter can contain normal text and latex commands (with some restriction, see section 1.2). Nodes that have link associated to them, or which have a link pointing to them must be specified with the node specification syntax described hereunder.

The syntax for creating nodes is as follows. If the node has no link associated to it then you just use the syntax `(nodename){content of the node}` where `nodename` is a name that you give to the node and `content` is the text you want to write inside the node.

If there is a link attached to the node then you use the syntax

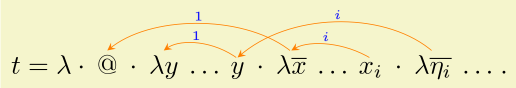
`(targetnode-nodename,angle:label){content}`

where

- `nodename` is a name that you give to the node,
- `targetnode` is the name of the node where the link must point to,
- `angle` (optional) specifies the angle between the horizontal line and the line tangent to the arrow representing the link at the top of the node,
- `label` (optional) is a text to be printed out on top of the link,
- `content` is the content of the node.

The arguments `angle` and `label` are optional. When the angle is not specified it is set to a default value of 45 (degrees).

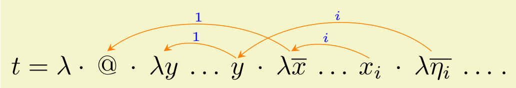
Here is an example:


1
1
i
i
$t = \lambda \cdot @ \cdot \lambda y \ldots y \cdot \lambda \overline{x} \ldots x_i \cdot \lambda \overline{\eta_i} \ldots .$

```
\Pstr[20pt]{
t = \lambda\cdot (app){@} \cdot (ly){\lambda y}
\ldots (y-ly,35:1){y} \cdot
(lx-app,38:1){\lambda \overline{x}}
\ldots (x-lx,30:i){x_i} \cdot
(leta-y,40:i){\lambda \overline{\eta_i}} {\ldots}
}.
```

1.2 Alternative syntax

You can use an alternative syntax to specify sequences. For instance the previous example can also be defined as follows:


1
1
i
i
$t = \lambda \cdot @ \cdot \lambda y \ldots y \cdot \lambda \overline{x} \ldots x_i \cdot \lambda \overline{\eta_i} \ldots .$

```
\pstr[0.7cm]{
\nd t = \lambda \cdot (app){@}
\nd \cdot (ly){\lambda y}
\nd \ldots (y-ly,35:1) {y}
\nd \cdot (lx-app,38:1){\lambda \overline{x}}
\nd \ldots (x-lx,30:i){x_i}
\nd \cdot (letay-y,40:i){\lambda \overline{\eta_i}}
\txt{\ldots} }.
```

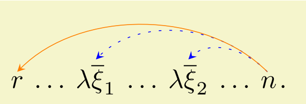
Although this syntax is more cumbersome than the succinct one, it permits to typeset sequences that are not definable with the succinct syntax. The following example is such an example:

[Sorry, you need to compile the documentation with latex in order to view this example]

Important note: The previous example can only be done using the pstricks engine and using the long syntax. This is due to the use of the latex command `\overbrace` which embraces several nodes inside the sequence. This cannot be handled in the PGF rendering mode due to the impossibility to create nodes outside of a PGF environment. It may be possible to fix this problem by using the overlay feature of PGF/Tikz, however overlays seem to work only within Tikz environment and not in standard PGF ones. Also overlays have the disadvantage to require two latex compilations in order to resolve node name references, and moreover they work only with pdflatex (not with latex).

1.2.1 Adding more links

It is possible to create additional links starting from a same node:


$r \ldots \lambda\overline{\xi}_1 \ldots \lambda\overline{\xi}_2 \ldots n.$

```
\pstr[0.6cm]{\ldots  \nd(r){r}
  \nd\ldots (l1){\lambda \overline{\xi}_1}
  \nd\ldots (l2){\lambda \overline{\xi}_2 }
  \nd\ldots (n-r,45){n}
  \arrow{n}{l1}{45}{}{blue}{dotted}
  \arrow{n}{l2}{45}{}{blue}{dotted}
 }.
```

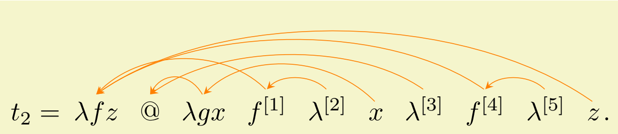
The syntax for arrow creation is:

`\arrow`{source}{target}{angle}{label}{linecolor}{linestyle}

- `source` is the name of the source node,
- `target` is the name of the target node,
- `angle` is the arrow angle,
- `label` is a text to be printed out on top of the link,
- `linecolor` is the color to be used to draw the link,
- `linestyle` is a line style used to draw the arrow. Valid styles are: `none`, `dashed`, `dotted` and `solid`.

2 Examples

2.1 A sequence with more links


$t_2 = \lambda f z \quad @ \quad \lambda g x \quad f^{[1]} \quad \lambda^{[2]} \quad x \quad \lambda^{[3]} \quad f^{[4]} \quad \lambda^{[5]} \quad z.$

```
\Pstr[1cm][2pt]{
t_2 = (n){\lambda f z}      \ (n2){@}
\ (n3-n2,60){\lambda g x}   \ (n4-n){f^{[1]}}
\ (n5-n4){\lambda^{[2]}}    \ (n6-n3){x}
\ (n7-n2,35){\lambda^{[3]}} \ (n8-n,35){f^{[4]}}
\ (n9-n8,45){\lambda^{[5]}} \ (n10-n,35){z} }.
```

2.2 Elevation of the links

A second optional parameter can be passed to `\pstr` in order to elevate the links in the sequence as follows:

First line = there should be enough space between the bottom of this line and the arc of the next line.

$s = s \cdot m \cdot \ldots \cdot \lambda\overline{\xi}$ the arc is elevated using the raise option of pstr.

```
$\begin{array}{rclrcl}
\mbox{First line} &=& \parbox[t]{8cm}{there should be enough space between the
                        bottom of this line and the arc of the next line.} \\
                s &=& \Pstr[28pt][10pt]{ s \cdot (m){m} \cdot \ldots \cdot (lmd-m,40)
                         {\lambda\overline{\xi}} }
    \quad \mbox{ the arc is elevated using the raise option of pstr.}
\end{array}$
```

2.3 Two sequences on the same line

$t = \lambda f z \quad @ \quad \lambda g x \quad f^{[1]} \quad \lambda^{[2]} \quad x \quad \lambda^{[3]} \quad f^{[4]} \quad \lambda^{[5]} \quad z \qquad t \upharpoonright r = \lambda f z \quad f^{[1]} \quad \lambda^{[2]} \quad f^{[4]} \quad \lambda^{[5]} \quad z$

```
\Pstr[0.8cm]{
t = (n){\lambda f z}\ (n2){@} \ (n3-n2,60){\lambda g x} \ (n4-n,45){f^{[1]}}
\ (n5-n4,45){\lambda^{[2]}} \ (n6-n3,45){x} \ (n7-n2,35){\lambda^{[3]}}
\ (n8-n,35){f^{[4]}} \ (n9-n8,45){\lambda^{[5]}} \ (n10-n,35){z}
}
\qquad
\Pstr[0.9cm]{
t\upharpoonright r = (n){\lambda f z} \ (n4-n,50){f}^{[1]}
\ (n5-n4,60){\lambda}^{[2]} \ (n8-n,45){f}^{[4]}
\ (n9-n8,60){\lambda}^{[5]} \ (n10-n,40){z}}
```

2.4 Sequences containing parenthesis

If you insert text containing parenthesis '(' and ')' between two nodes then you must surround the text with curly braces otherwise `pstr` will interpret it as a node specification:

$\psi_M(t \upharpoonright r) = q^0 \quad q^1 \quad q^2 \quad q^1 \quad q^2 \quad q^3 \in [\![M]\!]\ .$

```
\Pstr[0.8cm]{ {\psi_M(t\upharpoonright r) =\ }
(n){q^0}\ (n4-n,60){q^1}\ (n5-n4,60){q^2}
\ (n8-n,45){q^1}\ (n9-n8,60){q^2}
\ (n10-n,38){q^3} \in [\![ M ]\!] \ .}
```

2.5 Example taken from an article

(InputVar) If $t_1 \cdot x \cdot t_2$ is a traversal with $x \in N_{var}^{\upharpoonright r}$ and $?(t_1 \cdot x \cdot t_2) = ?(t_1) \cdot x$ then so is $t_1 \cdot x \cdot t_2 \cdot n$ for all λ-node n whose parent occurs in $\llcorner t_1 \cdot x \lrcorner$, n pointing to some occurrence of its parent node in $\llcorner t_1 \cdot x \lrcorner$.

(Copycat$^{@}$) If $t \cdot @ \cdot \lambda\overline{z} \ldots v_{\lambda\overline{z}}$ is a traversal then so is $t \cdot @ \cdot \lambda\overline{z} \ldots v_{\lambda\overline{z}} \cdot v_{@}$.

(Copycat$^{\lambda}$) If $t \cdot \lambda\overline{\xi} \cdot x \ldots v_{x}$ is a traversal then so is $t \cdot \lambda\overline{\xi} \cdot x \ldots v_{x} \cdot v_{\lambda\overline{\xi}}$.

(Copycat^{var}) If $\begin{tikzcd}[column sep=tiny] t \cdot y \cdot & \lambda\overline{\xi} & \ldots & v_{\lambda\overline{\xi}} \arrow[ll, bend right, "v"'] \end{tikzcd}$ is a traversal for some variable y not in $N^{\upharpoonright r}_{var}$ then so is $\begin{tikzcd}[column sep=tiny] t \cdot & y & \cdot & \lambda\overline{\xi} & \ldots & v_{\lambda\overline{\xi}} \arrow[ll, bend right, "v"'] & \cdot & v_{y} \arrow[llllll, bend right, "v"'] \end{tikzcd}$.