`tree-dvips` Tree Macros

Emma Pease

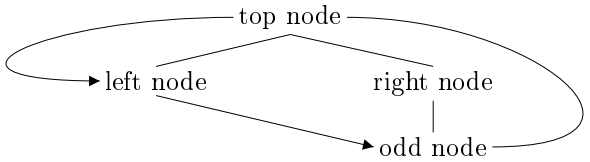
September 25, 2001

[This is Emma Pease's original documentation, slightly edited and adapted for the Web by Doug Arnold

This documentation is available as a PostScript File

The `tree-dvips` package is available at the usual places (follow this link for some suggestions.]

The tree macros package allows one to integrate TEX and Postscript. For example, one can use TEX to layout a tree and have Postscript draw the lines.



These macros work by defining locations on a page and then manipulating them in a variety of ways. The commands that created the above tree are as follows:

```
\begin{tabular}{ccc}
                    &\node{a}{top node}\\[3ex]
\node{b}{left node} &     & \node{c}{right node}\\[3ex]
                    &     & \node{d}{odd node}
\end{tabular}
\nodeconnect{a}{b}
\nodeconnect{a}{c}
\nodeconnect{c}{d}
\nodecurve[r]{a}[r]{d}{1in}
\anodeconnect[b]{b}[l]{d}
\anodecurve[l]{a}[l]{b}{1in}
```

You will notice that four nodes are defined, a, b, c, and d, using the `\node` command. These nodes are then connected using the `\nodeconnect` and `\nodecurve` commands.

1 Locating Commands

Location commands are those that deal with defining a location on a page. The basic command

```
\node{nodename}{object}
```

Each node has its name, height, width, and the location of the lower left hand corner point passed down to postscript where it will remain until needed. Note that the object will be printed by TEX but the lines drawn by Postscript. A variant of this command is

```
\nodepoint{nodename}[horizontal displace][vertical displace]
```

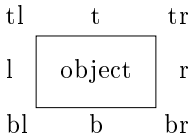
The node's height and width are 0pts, but the location can be displaced.

2 Connecting Commands

These commands connect two or more nodes. The two basic ones are

```
\nodeconnect[fromloc]{fromnodename}[toloc]{tonodename}
\nodecurve[fromloc]{fromnodename}[toloc]{tonodename}{depth}
```

fromnodename and tonodename must be the names of two existing nodes. Imagine the node as a box, fromloc and toloc are the locations on that box to draw the connecting lines.

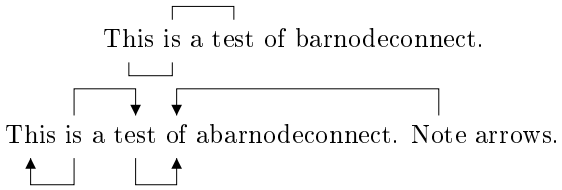


The present choices are t [top], b [bottom], l [left], r [right], tl [topleft], tr [topright], bl [bottomleft], and br [bottomright]. These could be expanded.

Other connecting commands are

```
\barnodeconnect[depth]{fromnodename}{tonodename}
\anodeconnect[fromloc]{fromnodename}[toloc]{tonodename}
\anodecurve[fromloc]{fromnodename}[toloc]{tonodename}{depth}
```

For example,



```
\node{c}{\strut This} \node{a}{\strut is} a \node{b}{\strut test}
of barnodeconnect.
\barnodeconnect{a}{b}
\barnodeconnect[-5pt]{a}{c}
\bigskip\bigskip

\node{c}{\strut This} \node{a}{\strut is} a \node{b}{\strut test}
\node{d}{\strut of} abarnodeconnect.  \node{e}{\strut Note} arrows.
\abarnodeconnect[10pt]{a}{b}
\abarnodeconnect[-10pt]{a}{c}
\abarnodeconnect[-10pt]{b}{d}
\abarnodeconnect[10pt]{e}{d}
```

A negative depth places the bar below the line; a positive depth (or the default, which is 5pt) places the bar above the line.

A few odd commands

```
\nodetriangle{fromnodename}{tonodename}
```

This creates a triangle whose apex is the bottom of `fromnodename` and whose base is the top of `tonodename`.

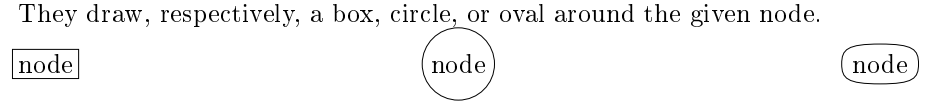
`\anodeconnect` and `\anodecurve` are the same as `\nodeconnect` and `\nodecurve` except that the connecting line has a arrowhead on it pointing to the second node.

3 Single Node commands

These commands adjust something around a single node rather than connecting nodes. The basic commands are

```
\nodebox{nodename}
\nodecircle[depth]{nodename}
\nodeoval{nodename}
```

They draw, respectively, a box, circle, or oval around the given node.



You will probably wish to call these commands after you have called all the connecting commands you will be using in a particular diagram.

4 Parameters

At the moment there are three parameters that can be changed. They are

- `\nodemargin` - A node's height and width are defined as the height plus depth and width of an hbox enclosing the object plus the nodemargin on each side. The default is 2pt.

- `\treelinewidth` - The width of the lines. The default is .3pt.
- `\dashlength` - The length of the dash, if you are using dashed lines. The default is 0pt (solid line).[1]
- `\arrowwidth` - the width of the arrowhead in the `\anodeconnect` and `\anodecurve` commands. Default is 4 pt.
- `\arrowlength` - the length of the arrowhead. Default is 4pt.

5 How to Run

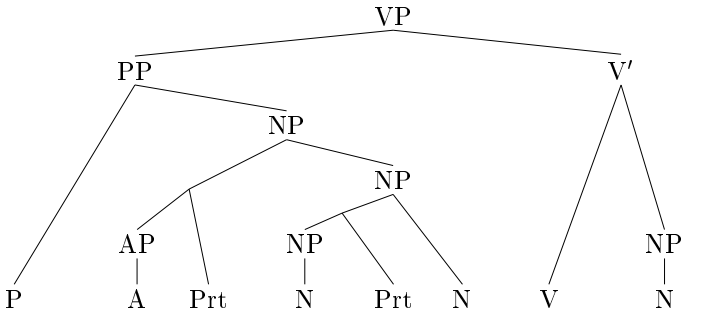
In the preamble, put:

```
\usepackage{tree-dvips}
```

Run through LaTeX and send to a postscript printer using dvips

6 Examples

A series of examples follow.



‘shoot at the red head of the bird’

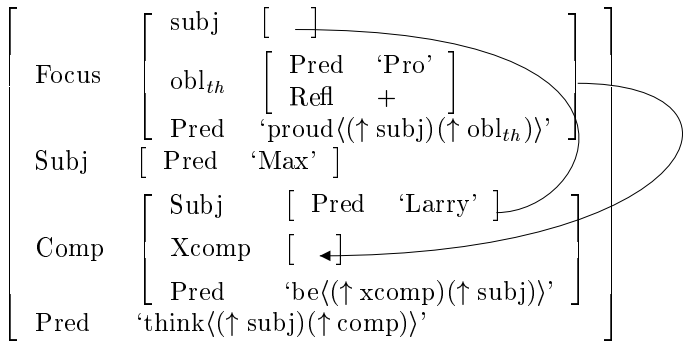
```
\let\mc=\multicolumn
\begin{tabular}[t]{@{}lllllllll@{}}
    &    &    &      &\node{a}{VP} \\[2ex]
    &\node{b}{PP}
         &    &      &    &    &\mc{2}{c}{\node{c}{V$'$}}\\[2ex]
    &    &    &\node{d}{NP} \\[2ex]
    &    &\nodepoint{e}
         &      &\node{f}{NP} \\
```

[1] The length of the dash and the length between the dashes are the same. An exercise for someone who knows postscript and tex is to allow the dash and the blank to vary in size.

```
    &    &    &\hfill\nodepoint{r}[0pt][3pt]   \\
    &\hfil\node{g}{AP}
         &    &\hfil\node{h}{NP}
                    &    &    &     &\hfil\node{i}{NP}\\[2ex]
\node{j}{P}
    &\hfil\node{k}{A}
         & \node{l}{Prt}
              &\hfil\node{m}{N}
                     &\node{n}{Prt}
                          &\node{o}{N}
                               &\node{p}{V}
                                     &\hfil\node{q}{N}\\[2ex]
‘zaw&‘oN &‘geq&?njiaw&‘geq&‘dou& khe &tshjaN \\
toward &red &    & bird &    &head& open&gun \\[1ex]
\mc{9}{@{}l}{‘shoot at the red head of the bird’} \\
\end{tabular}
\nodeconnect{a}{b}     \nodeconnect{e}{l}
\nodeconnect{a}{c}     \nodeconnect{d}{f}
\nodeconnect{b}{d}     \nodeconnect{g}{k}
\nodeconnect{b}{j}     \nodeconnect{f}{r}
\nodeconnect{c}{p}     \nodeconnect{r}{h}
\nodeconnect{c}{i}     \nodeconnect{r}{n}
\nodeconnect{d}{e}     \nodeconnect{f}{o}
\nodeconnect{e}{g}     \nodeconnect{h}{m}
\nodeconnect{i}{q}
```

The following two examples use `\outerfs` and `\modsmalltree`; these are both part of the lingmacros package. See `lingmacros.sty` for more information.

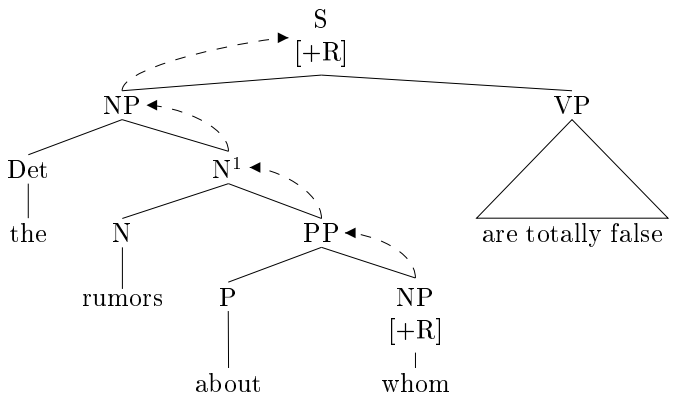
(100)




```
\enumsentence[(100)]{\evnup[2pt]
{\outerfs{
Focus & \outerfs{subj       & \outerfs{\ \nodepoint{a}\ }\\[1ex]
                obl$_{th}$ & \outerfs{Pred & ‘Pro’\\
                                           Refl & +}\\[1ex]
                Pred       & ‘proud$\langle(\uparrow {\rm subj})
```

```
                              (\uparrow {\rm obl}_{th})\rangle$’}%
\nodepoint{d}[-3pt][0pt]\\[2ex]
Subj & \outerfs{Pred & ‘Max’}\\[1ex]
Comp & \outerfs{Subj &\outerfs{Pred & ‘Larry’}%
\nodepoint{c}[-3pt][0pt]\\[1ex]
                Xcomp&\outerfs{\ \nodepoint{b}\ }\\[1ex]
                Pred & ‘be$\langle(\uparrow {\rm xcomp})
                       (\uparrow {\rm subj})\rangle$’}\\[1ex]
Pred & ‘think$\langle(\uparrow {\rm subj})
                     (\uparrow {\rm comp})\rangle$’}\\
}
\nodecurve[r]{a}[r]{c}{2in}[.5in]
\anodecurve[r]{d}[r]{b}{1in}[2in]
}
```

(1)




```
\enumsentence{\modsmalltree{6}{
     &       &   &\ns\node{a}{\begin{tabular}[t]{@{}c@{}}S\\
                          {}[+R]\end{tabular}}   \\
     &\node{b}{NP}
             &   &  && \node{c}{VP} \\
\node{d}{Det}
     &       &\node{e}{N$^1$} \\
\node{h}{the}
     &\node{i}{N}
             &   &\node{f}{PP}
                    &&\node{g}{are totally false}\\
     &\node{n}{rumors}
             &\node{j}{P}
                 &  &\ns\node{k}{\begin{tabular}[t]{@{}c@{}}NP\\
                                  {}[+R] \end{tabular}}\\
     &       &\node{l}{about}
                 &  &\node{m}{whom}}
\nodeconnect{a}{b}         \nodeconnect{e}{f}
```

```
\nodeconnect{a}{c}          \nodeconnect{i}{n}
\nodeconnect{b}{d}          \nodeconnect{f}{j}
\nodeconnect{b}{e}          \nodeconnect{f}{k}
\nodeconnect{d}{h}          \nodeconnect{j}{l}
\nodeconnect{e}{i}          \nodeconnect{k}{m}
\nodetriangle{c}{g}
{\makedash{4pt}
\anodecurve[t]{b}[l]{a}{10pt}  \anodecurve[t]{f}[r]{e}{10pt}
\anodecurve[t]{e}[r]{b}{10pt}  \anodecurve[t]{k}[r]{f}{10pt}
}}
```

7 Errors

A multitude of caveats.

- Any commands calling nodes must be read while TEX is still processing the page the nodes are defined on. In other words don't define the nodes on page 1 and connect them with commands that appear at the end of the paper.
- Nodes mentioned in node connecting commands must exist or else the job won't print.
- Make sure the dvips postscript output is sent to a postscript printer. It is possible to send the dvi, but not the postscript, output to another printer; the lines just won't appear, assuming the printer ignores specials it doesn't know about.