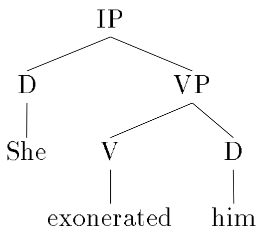
ps-trees.sty Documentation

Wolfgang Sternefeld
Version 2 — May 1998

May 5, 1999

Experimenting with a number of tree-macros I found that complex real-life examples often lead to problems with controlling the positioning of nodes in a tree. I therefore decided to use tree-dvips.sty (by Emma Pease), where the user has full control over the position of nodes by putting them into a table. The present macro inputs tree-dvips.sty and presupposes knowledge of its node-commands.

One of the shortcomings of tree-dvips.sty is that it does not control the correct use of the names for nodes. This means that in case of a user error, no warning will appear and no output will be printed. The basic idea of the present macro is to regain control over the use of names by using them as an encoding of the structure of a tree. More specifically, for a node named x to be connected with a node named y (i.e. x and y are the first arguments of the command `\node`) y must have the form cx, where x and y are strings of characters and c is exactly one character. For ordinary branching trees, this makes the specification of `\nodeconnect{#1}{#2}` superfluous. Consider the following example and compare the printed file with the .tex-source of this file:




```
\begin{treetab}{*4c}
&\Node{}{IP}\\[3ex]
\Node{l}{D}&&\Node{r}{VP}\\[3ex]
\Node{ml}{She}&\Node{lr}{V}&&\Node{rr}{D}\\[3ex]
&\Node{mlr}{exonerated}&&\Node{mrr}{him}
\end{treetab}
```

`\begin{treetab}` is defined in terms of `\begin{tabular}`. The actual execution of the node connections (based on the node names as specified by the first argument of `\Node`) is performed as a part of `\end{treetab}`. In case a node cannot be interpreted as a node in a tree, a warning will be typed out.

The remaining problem is to gain control over the width of columns in a table. This is often tedious, but the effort seems worthwhile as evidenced by the poor results of macros that calculate the position of nodes automatically. Being essentially an aesthetic problem, the present macro offers only limited support towards a uniform solution. The following, however, might be helpful.

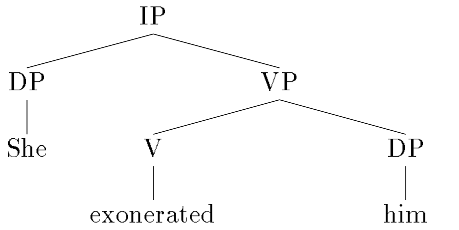
First of all, the effects of `\tabcolsep` and `\nodemargin` on a precise positioning of nodes are disturbing, so both values are set to zero by `\begin{treetab}`. In order to achieve a minimum of symmetry of branchings, a first idea would be that all rows should have the same width, implying that the width of each node should be determined by the widest leave. Since individual nodes that come later in a table might determine the size of previous nodes, it might be a good idea to put all relevant nodes (by default: all leaves) into boxes before starting the table. The minimal size of all nodes (or columns) can then be determined by the command `\NodeWidthNo{n}`, where n is the number of the box, whose width should become the minimal width of all other columns. `\NodeWidthNo` is defined as follows:

```
\newcommand{\NodeWidthNo}[1]{\setlength{\MinNodeWidth}{\the\wd#1}}
```

The length `\MinNodeWidth` will subsequently be used to determine the minimal width of each node. (In order to give an individual node the width of a particular prespecified box n, the node command can take an optional argument `\Node[n]{...}{...}`).

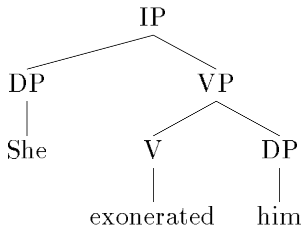
For example, if we let the word *exonerate* determine the size of all nodes, the result is the following:

```
\setbox1=\hbox{She}
\setbox2=\hbox{exonerated}
\setbox3=\hbox{him}
\NodeWidthNo2
%
\begin{treetab}{*4c}
&\Node{}{IP}\\[3ex]
\Node{l} {DP}&&\Node{r}{VP}\\[3ex]
\NodeNo1{ml}&\Node{lr}{V}&&\Node{rr} {DP}\\[3ex]
&\NodeNo2{mlr}&&\NodeNo3{mrr}
\end{treetab}
```

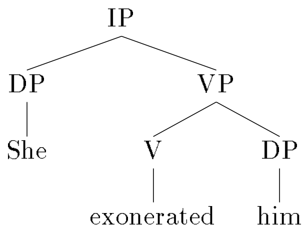


But now the space between box2 and box3 might strike one as too wide, being seperated by the box of the VP. One possibility would be to specify VPs node as `\Node[0]{r}{VP}` which puts VP into a box whose width is `\wd0`, which is zero because box0 has not yet been used. A more general way would be to place the VP node right into the middle between two columns. This saves us one column in the table; the command `\NodeZ` makes a box of width zero and puts it at the right edge of a column (i.e. “between” two columns):

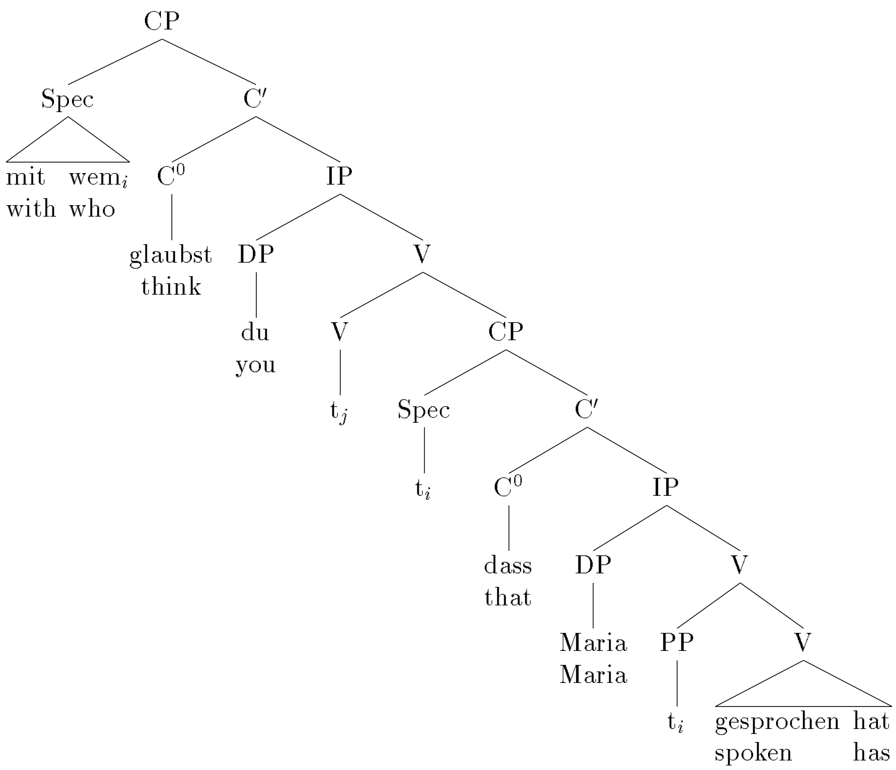
```
\begin{treetab}{ccc}
&\Node{}{IP}\\[3ex]
\Node{l} {DP}&\NodeZ{r}{VP}\\[3ex]
\NodeNo1{ml}&\Node{lr}{V}&\Node{rr} {DP}\\[3ex]
&\NodeNo2{mlr}&\NodeNo3{mrr}
\end{treetab}
```



But now we got an asymmetry of the branches at the top level, which can be fixed by shifting the IP to the left. This can be done by replacing `\Node{}{IP}` by `\NodeZ{}{IP}\hspace*{0.75\wd2}`:



For larger trees these asymmetries have to be levelled out consecutively, as shown in the next tree (an example from real life):



We first define a number of shortcuts which will keep the table managable:

```
\let\N=\Node
\let\NZ=\NodeZ
\let\NNo=\NodeNo
\let\NTNo=\NodeTNo
\let\MNW=\MinNodeWidth
\newcommand{\adjustA}{\hspace*{.75\MNW}}
\newcommand{\adjustB}{\hspace*{.625\MNW}}
\newcommand{\adjustC}{\hspace*{.5625\MNW}}
\newcommand{\adjustD}{\hspace*{.53125\MNW}}
\newcommand{\adjustE}{\hspace*{.515625\MNW}}
\newcommand{\End}{\\[4ex]}
\newcommand{\trivtab}[2]{\hbox{\begin{tabular}[t]{@{}c@{}}#1\\
               #2\end{tabular}}}
\newcommand{\twotab}[2]{\hbox{\begin{tabular}[t]{@{}ll@{}}#1\\
               #2\end{tabular}}}
```

Next we put all leaves into boxes:

```
\tabcolsep2pt
\setbox0=\twotab{mit & wem$_i$}{with & who}
\setbox1=\trivtab{glaubst}{think}
\setbox2=\trivtab{du}{you}
\setbox3=\hbox{t$_j$}
\setbox4=\hbox{t$_i$}
\setbox5=\trivtab{dass}{that}
\setbox6=\trivtab{Maria}{Maria}
\setbox7=\hbox{t$_i$}
\setbox8=\twotab{gesprochen & hat}{spoken & has}
```

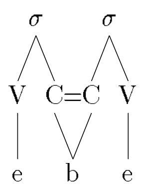
And finially we specify the minimal node width and construct the table:

```
\NodeWidthNo1
\begin{treetab}{*9{c}}
&\N{}{CP}\End
\N{l}{Spec}&&\N{r}{C$'$}\End
\NTNo0{ml}&\N{lr}{C$^0$}&&\N{rr}{IP}\End
&\NNo1[6]{mlr}&\N{lrr}{DP}&&\NZ{rrr}{V}\adjustE\End
&&\NNo2{mlrr}&\N{lrrr}{V}&&\NZ{rrrr}{CP}\adjustD\End
&&&\NNo3{mlrrr}&\N{lrrrr}{Spec}&&\NZ{rrrrr}{C$'$}\adjustC\End
&&&&\NNo4{mlrrrr}&\N{lrrrrr}{C$^0$}&&\NZ{rrrrrr}{IP}\adjustB\End
&&&&&\NNo5{mlrrrrr}&\N{lrrrrrr}{DP}&&\NZ{rrrrrrr}{V}\adjustA\End
&&&&&&\NNo6{mlrrrrrr}&\N{lrrrrrrr}{PP}&\hspace{\wd1}\NZ{rrrrrrrr}{V}\End
&&&&&&&\NNo7{mlrrrrrrr}&\NodeTZNo8{mrrrrrrrr}
\end{treetab}
```

Note that box0 is wider than box1, therefore the root should not be in the middle of box1 but should appear exactly in the middle between box0 and box2. We therefore have to replace the first line of the table by

```
\rlap{\hspace*{.25\wd0}\adjustA\makebox[0pt]{\N{}{CP}}}\End
```

Here is another real life example from phonology:



The main point to be observed is that the first line cannot contain `\Node`'s (but must contain `\node`'s), otherwise the macro will complain that it cannot find the root of a tree:

```
\setbox1=\hbox{mn}
\NodeWidthNo1
\begin{treetab}[t]{*4c}
\nodeZ{l}{$\sigma$}&&\nodeZ{r}{$\sigma$}\End
\Node{ll}{V}&\Node{rl}{C}\nodeZ{xx}{=}&\Node{gr}{C}&\Node{lr}{V}\End
\Node{mll}{e}&\NodeZ{mrl}{b}&&\Node{mlr}{e}
\end{treetab}
\nodeconnect{gr}{mrl}
```

One final warning: It seems to me that TEX internally uses certain boxes to construct its tables. This means that the width of a box might have changed by TEX internally to zero before you come to the point where you want to use the lenght prespecified by `\setbox`. It might therefore be necessary to store this lenght as soon as possible; e.g. by saying `\NodeWidthNo{n}` immediately after having defined `\setboxn`.