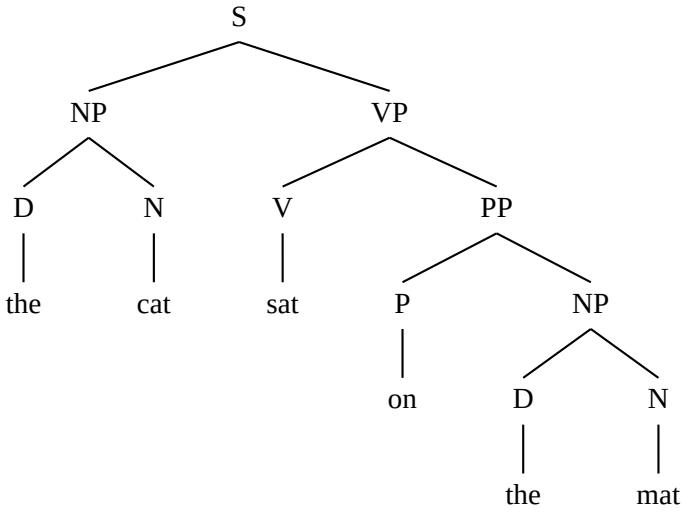
pst-qtree: a Qtree-like front end for pst-tree

David Chiang

April 9, 2007

Basic trees are specified exactly as in Qtree:

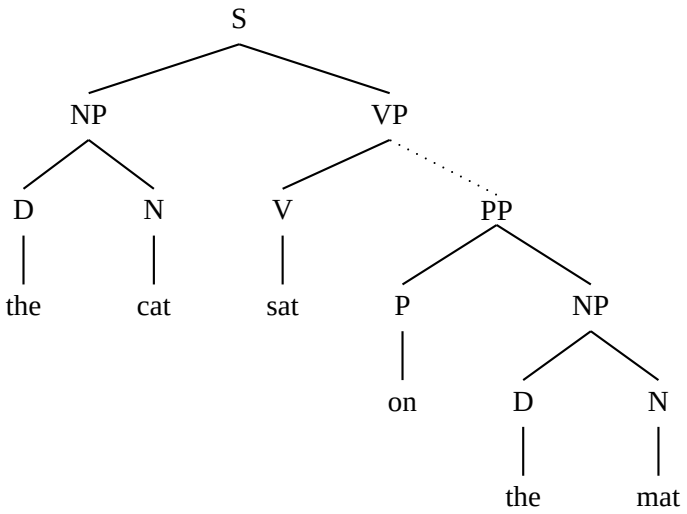
```
\Tree [.S [.NP [.D the ] [.N cat ] ]
          [.VP [.V sat ]
               [.PP [.P on ]
                    [.NP [.D the ] [.N mat ] ] ] ] ]
```



If a node label starts with `!`, then the node is not automatically wrapped inside a `\psnode` for you. You may use a different node macro, and can pass options to it:

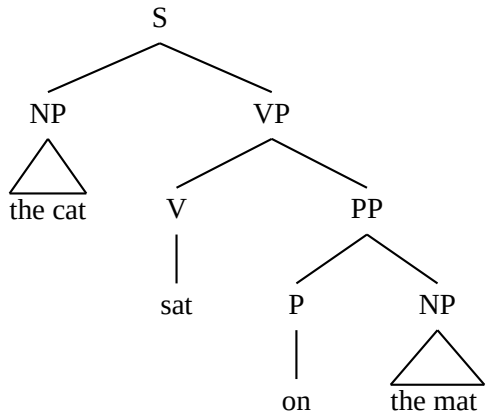
```
\newcommand{\dottededge}{\ncdiag[arm=0,angleA=-90,angleB=90,linestyle=dotted]}

\Tree [.S [.NP [.D the ] [.N cat ] ]
          [.VP [.V sat ]
               [.!\TR[edge=\dottededge]{PP} [.P on ]
                                            [.NP [.D the ] [.N mat ] ] ] ] ]
```



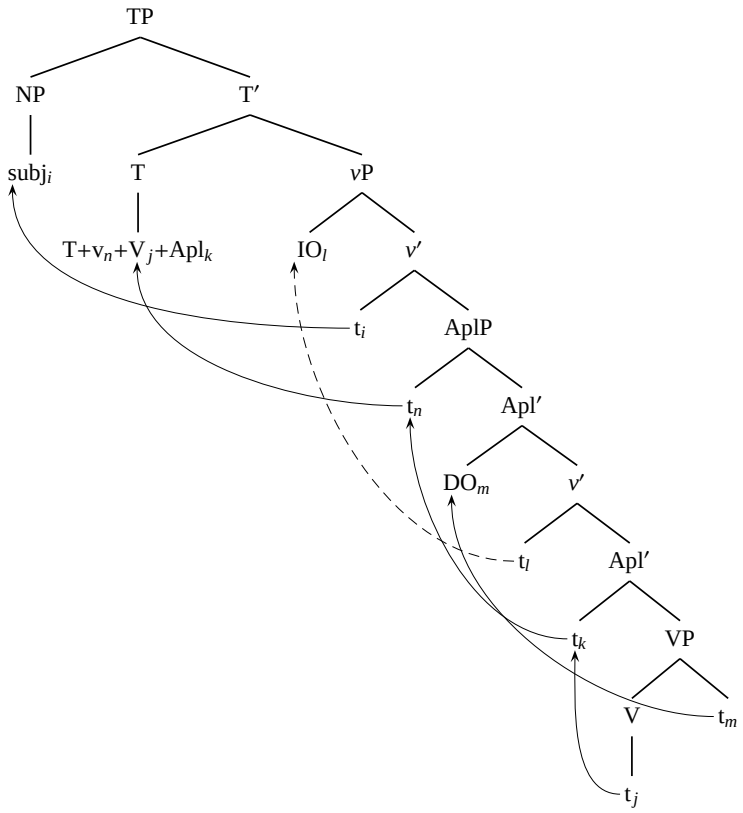
One special type of (large) node defined by `pst-qtree` is `\Troof`, which is similar to Qtree's `\qroof` but not identical in use.

```
\Tree [.S [.NP !\Troof{the cat} ]
          [.VP [.V sat ]
               [.PP [.P on ]
                    [.NP !\Troof{the mat } ] ] ] ]
```



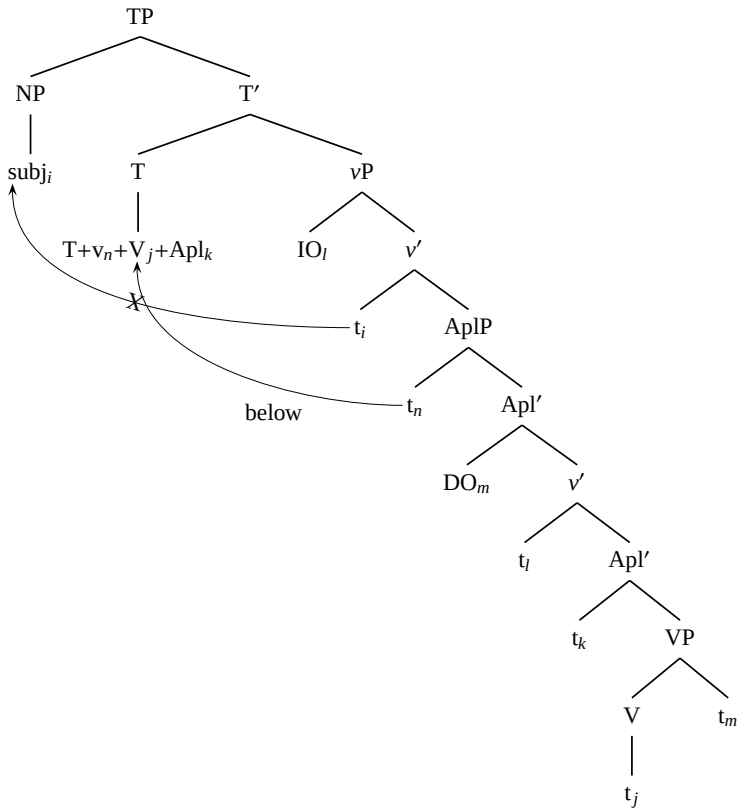
Arrows can be drawn between nodes using commands from `pstricks`: assign a label to each node using `\rnode`, then connect them using node connection commands such as `\nccurve` or `\ncangle`.

```
\Tree
[.TP [.NP \rnode{subj1}subj$_i$ ]
     [.T\1   [.T T+v$_n$+\rnode{V}V$_j$+Apl$_k$ ]
             [.{\it v}P \rnode{io}{ }IO$_l$
                 [.{\it v}\1 \rnode{subj2}t$_i$
                   [.AplP \rnode{v1}t$_n$
                     [.Apl\1 \rnode{do}DO$_m$
                       [.{\it v}\1  \rnode{io1}t$_l$
                         [.Apl\1 \rnode{apl1}t$_k$
                           [.VP [.V \rnode{V1}t$_j$ ]
                                \rnode{do1}t$_m$  ] ] ] ] ] ] ] ] ]
\psset{linewidth=0.3pt,arrowsize=4pt}
\psset{angleA=180,angleB=-90}
\nccurve{->}{subj2}{subj1}
\nccurve{->}{do1}{do}
\nccurve[linestyle=dashed]{->}{io1}{io}
\nccurve{->}{V1}{apl1}
\nccurve{->}{apl1}{v1}
\nccurve{->}{v1}{V}
```



Arrows can be labeled using `\lput{:U}` (on top of the arrow), or `\taput` `\tbput` `\tlput` or `\trput` (above, below, left of, or right of the arrow) after the `\nccurve` or `\ncangle` command.

```
\Tree
[.TP [.NP \rnode{subj1}subj$_i$ ]
     [.T\1   [.T T+v$_n$+\rnode{V}V$_j$+Apl$_k$ ]
             [.{\it v}P \rnode{io}{ }IO$_l$
                 [.{\it v}\1 \rnode{subj2}t$_i$
                   [.AplP \rnode{v1}t$_n$
                     [.Apl\1 \rnode{do}DO$_m$
                       [.{\it v}\1  \rnode{io1}t$_l$
                         [.Apl\1 \rnode{apl1}t$_k$
                           [.VP [.V \rnode{V1}t$_j$ ]
                                \rnode{do1}t$_m$  ] ] ] ] ] ] ] ] ]
\psset{linewidth=0.3pt,arrowsize=4pt}
\psset{angleA=180,angleB=-90}
\nccurve{->}{subj2}{subj1}\lput{:U}{X}
\nccurve{->}{v1}{V}\tbput{below}
```



Trees can be made to grow sideways, or upside down, using `\psset{treemode=R}` for right, L for left, or U for up. Other useful layout parameters inherited from `pstricks` include `levelsep`, `ltreesep`, `ltreefit` and `nodesep`.

```
\psset{treemode=R}
\Tree [.{VV}:xianzhu [.NN:chengshi NR:Zhongguo [.CD:shisi M:ge ]
                        NN:bianjing NN:kaifang ]
          [.NN:chengjiu NN:jingji NN:jianshe ] ]
```