

# tikz-qtrees: better trees with TikZ

David Chiang

Version 1.1 (Christmas 2009)

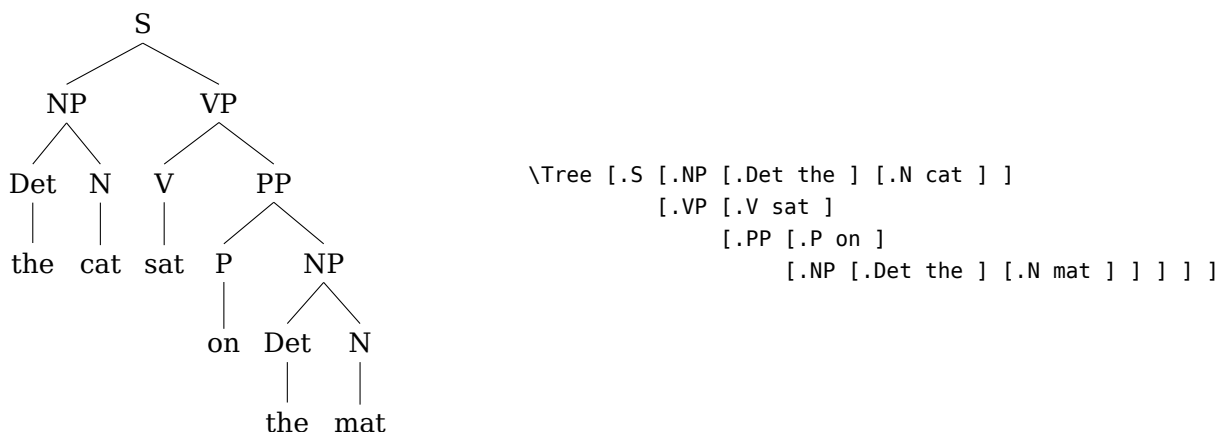
The `tikz-qtrees` package provides a macro for drawing trees with TikZ<sup>1</sup> using the easy syntax of Alexis Dimitriadis' `Qtree`<sup>2</sup>. It improves on TikZ's standard tree-drawing facility by laying out tree nodes without collisions; it improves on `Qtree` by adding lots of features from TikZ; and it improves on `pst-qtrees` in being usable with pdfTeX and XeTeX.

## 1 Basics

To load the package in L<sup>A</sup>T<sub>E</sub>X:

```
\usepackage{tikz}
\usepackage{tikz-qtrees}
```

The simplest usage is identical to `Qtree`:



Subtrees are delimited by square brackets. A subtree's root label is joined by a dot (.) to its opening bracket.<sup>3</sup> As in `Qtree`, spaces are required after every (internal or leaf) node label.

`\Tree` works inside or outside a `tikzpicture` environment, but many of the features described below require the explicit `tikzpicture` environment.

<sup>1</sup><http://sourceforge.net/projects/pgf/>

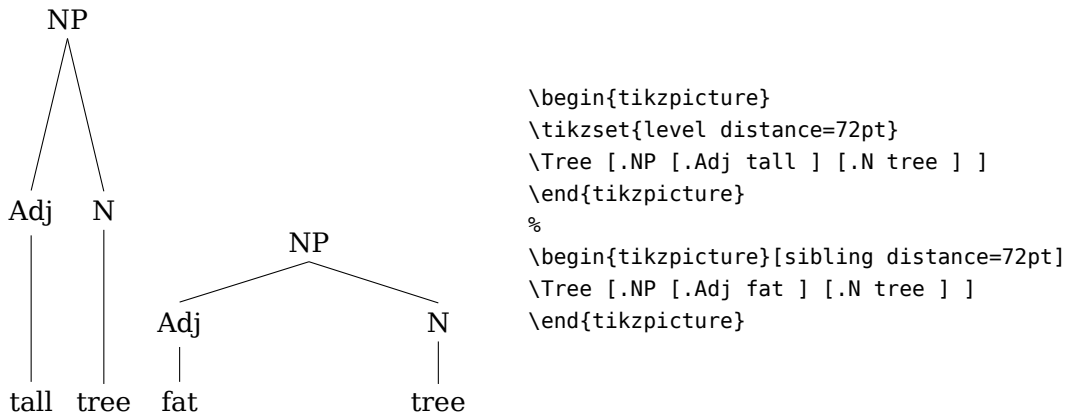
<sup>2</sup><http://www.ling.upenn.edu/advice/latex/qtrees/>

<sup>3</sup>You can also write the label after the closing bracket instead of the opening bracket, or both, or neither. Please see the `Qtree` documentation for details.

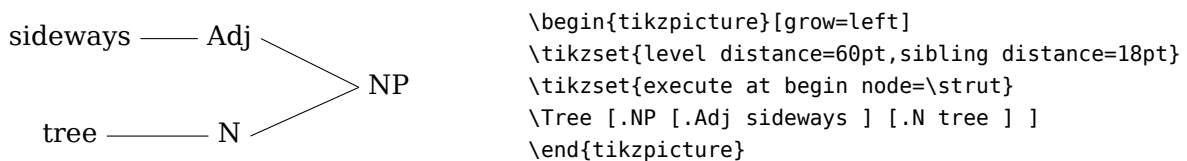
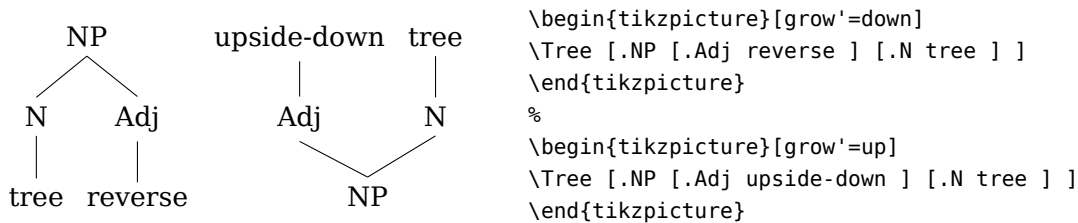
**Options** Some options for standard TikZ trees work for `\Tree` as well:

- `level distance`: vertical distance between the anchors of a parent and its children
- `sibling distance`: horizontal distance between the boundaries of sister subtrees (not the anchors of their roots, as in standard TikZ trees). Note that TikZ nodes already have some horizontal space around them (`inner xsep`, by default `0.3333em`), so even `sibling distance=0pt` leaves some room.

These are set either by writing `\tikzset{option=value}` or by writing `[option=value]` after a `\begin{tikzpicture}` or `\begin{scope}`.<sup>4</sup> For example:



The `grow=direction` and `grow'=direction` options control the orientation of trees just as for standard TikZ trees. However, *direction* must be one of up, down, left, or right. The difference between `grow` and `grow'` is that `grow` places children counterclockwise with respect to their parent and `grow'` places them clockwise:



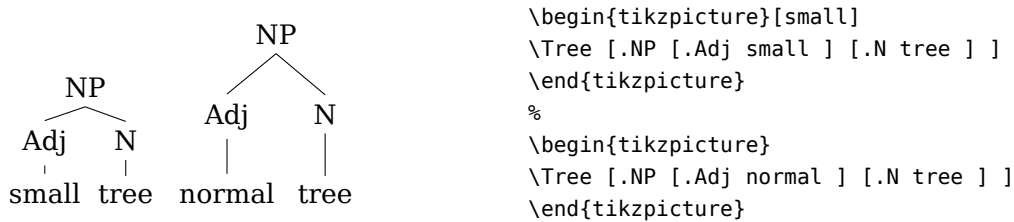
Note that in sideways trees, `level distance` is horizontal and `sibling distance` is vertical. Sideways trees do take a little extra adjusting to look right, since the defaults are geared towards vertically growing trees. The meaning of the option `execute at begin node=\strut` is, before typesetting the label of every node, insert the command `\strut`, which is an invisible box that maximizes the height and depth of the node.

<sup>4</sup>Allowing options after `\Tree` would have made sense, but there would be no way to disambiguate the two uses of square brackets.

**Styles** TikZ lets you define *styles* which encapsulate multiple options:

```
\tikzset{small/.style={level distance=20pt,sibling distance=0pt}}
```

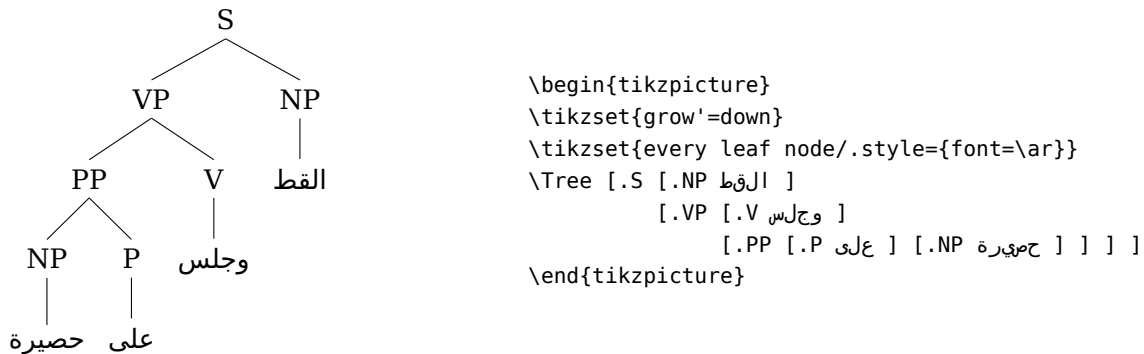
Then the option `small` causes the options in its definition to be used:



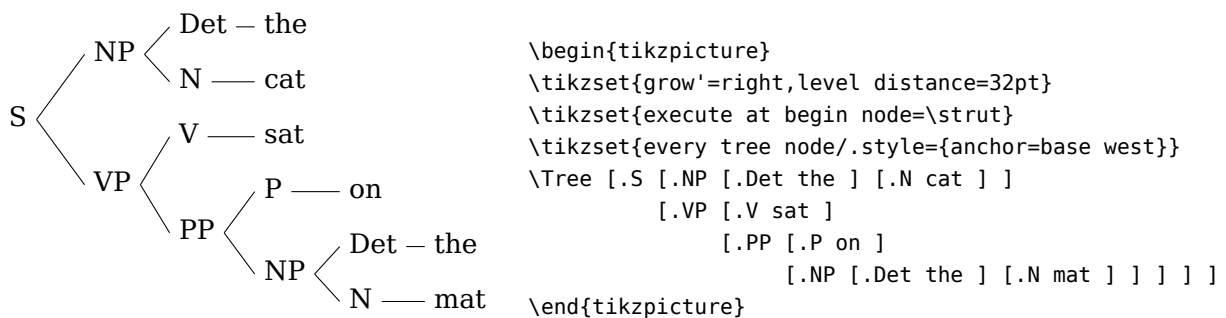
The following TikZ styles are automatically applied inside trees, providing a hook for you to change the appearance of particular kinds of tree parts:

- `every tree node` to every (internal and leaf) node (default: `anchor=base`)
- `every internal node` to every internal node
- `every leaf node` to every leaf node
- `edge from parent` to every edge (default: `draw`)

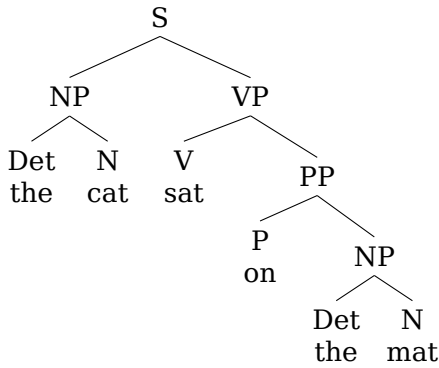
The options for nodes and edges are all handled by TikZ and are described in detail in the TikZ documentation. For example, if you have a font named `\ar` and want to set all the leaf labels in this font:



You can make the nodes in a sideways tree line up on their left edge using `anchor=base west`:



In Qtree, it was allowed to use a line break (`\`) inside a node, but TikZ nodes by default don't allow this. You can enable it like so:



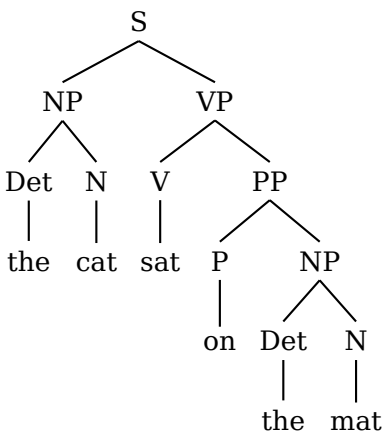
```

\begin{tikzpicture}
\tikzset{every tree node/.style={text width=2em,text centered}}
\Tree [.S [.NP Det\\the N\\cat ]
      [.VP V\\sat
        [.PP P\\on
          [.NP Det\\the N\\mat ] ] ] ] ]
\end{tikzpicture}

```

The downside is that you must specify a fixed node width. In the next release of TikZ, it will be possible to use instead a new `align=center` option without the fixed width.<sup>5</sup>

You can also define a style for all the edges in a tree. For example, if you want the edges to be a little darker:

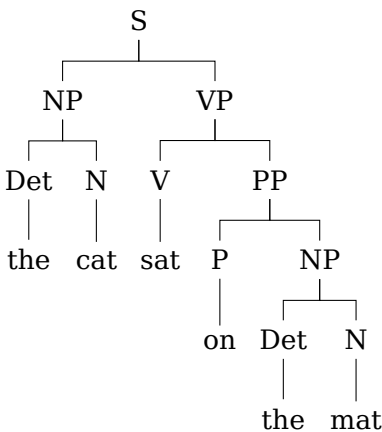


```

\begin{tikzpicture}
\tikzset{edge from parent/.style={draw,thick}}
\Tree [.S [.NP [.Det the ] [.N cat ] ]
      [.VP [.V sat ]
        [.PP [.P on ]
          [.NP [.Det the ] [.N mat ] ] ] ] ] ] ]
\end{tikzpicture}

```

The `draw` option is necessary, as by default they will not be drawn. As a more complex example, edges have an `edge from parent` path option which lets you change the shape of the edge. Its value is a TikZ path expressed in terms of `\tikzparentnode`, the parent node, and `\tikzchildnode`, the child node.



```

\begin{tikzpicture}
\tikzset{edge from parent/.style=
{draw,
edge from parent path={(\tikzparentnode.south)
-- +(0,-8pt)
-| (\tikzchildnode)}}}
\Tree [.S [.NP [.Det the ] [.N cat ] ]
      [.VP [.V sat ]
        [.PP [.P on ]
          [.NP [.Det the ] [.N mat ] ] ] ] ] ] ]
\end{tikzpicture}

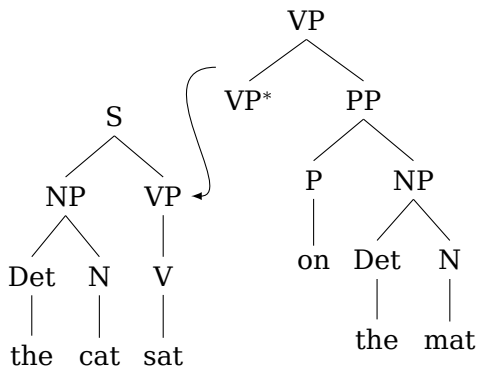
```

---

<sup>5</sup>Thanks to Alan Munn for figuring this out.



You can also refer to the whole subtree rooted at the node named *name* using `\subtreeof{name}`:

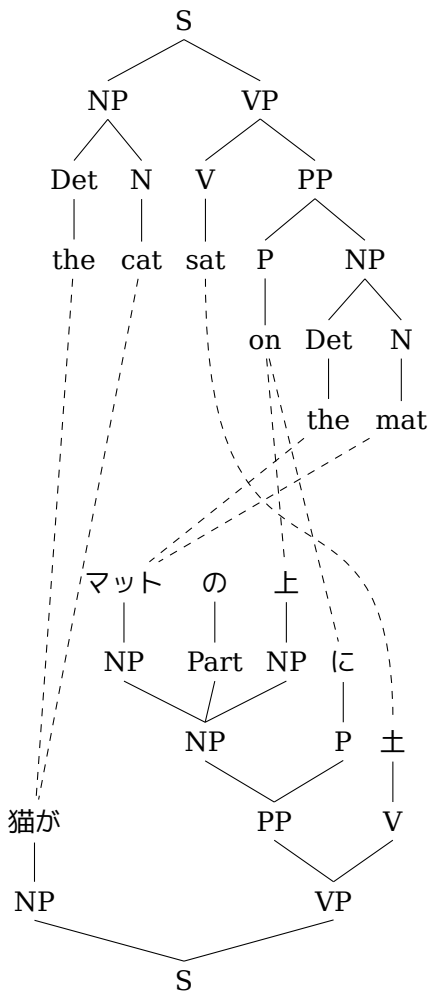


```

\begin{tikzpicture}
\Tree [.S [.NP [.Det the ] [.N cat ] ]
      [.\node{site}{VP}; [.V sat ] ] ]
\begin{scope}[shift={(1in,0.5in)}]
\Tree [.\node{root}{VP}; VP$^{\ast}$
      [.PP [.P on ]
            [.NP [.Det the ] [.N mat ] ] ] ]
\end{scope}
\draw[->](\subtreeof{root}.140) ..
controls +(west:1) and +(east:1) .. (site);
\end{tikzpicture}

```

Another example for machine translation people:



```

\begin{tikzpicture}
\Tree [.S [.NP [.Det \node{e1}{the}; ]
            [.N \node{e2}{cat}; ] ]
      [.VP [.V \node{e3}{sat}; ]
            [.PP [.P \node{e4}{on}; ]
                  [.NP [.Det \node{e5}{the}; ]
                        [.N \node{e6}{mat}; ] ] ] ] ]
\begin{scope}[yshift=-5in,grow'=up]
\tikzset{every leaf node/.style={font=\ja}}
\Tree [.S [.NP \node{j1}{猫が}; ]
      [.VP [.PP [.NP [.NP \node{j2}{マット}; ]
                    [.Part \node{j3}{の}; ]
                    [.NP \node{j4}{上}; ] ] ]
            [.P \node{j5}{に}; ] ]
      [.V \node{j6}{土}; ] ] ]
\end{scope}
\begin{scope}[dashed]
\draw (e1)--(j1);
\draw (e2)--(j2);
\draw (e3)..controls +(south:5) and +(north:4)..(j6);
\draw (e4)--(j4);
\draw (e4)--(j5);
\draw (e5)--(j2);
\draw (e6)--(j2);
\end{scope}
\end{tikzpicture}

```

### 3 Explicit edges

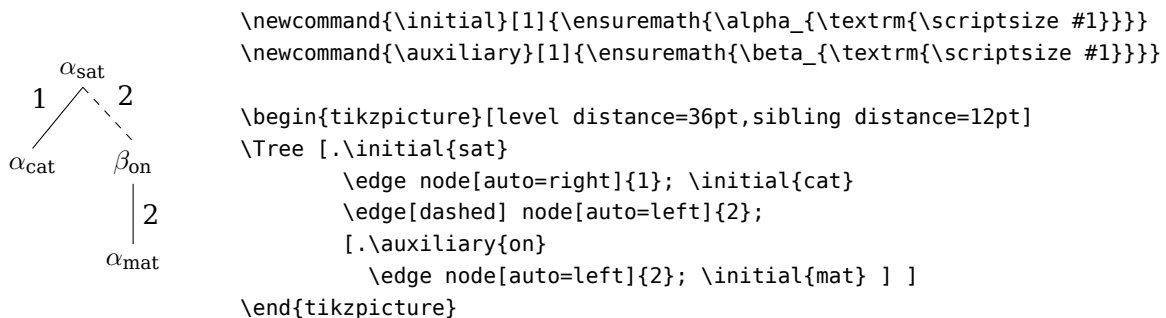
The edge from a parent to a child node is normally automatically drawn for you, but you can do it yourself with an `\edge` command *before* the corresponding child node. It is similar to the TikZ `edge from parent` command.<sup>7</sup>

```
\edge [options];
```

Again, don't forget the semicolon. The `[options]`, which are optional, let you change the appearance of the edge. You can also add an edge label:

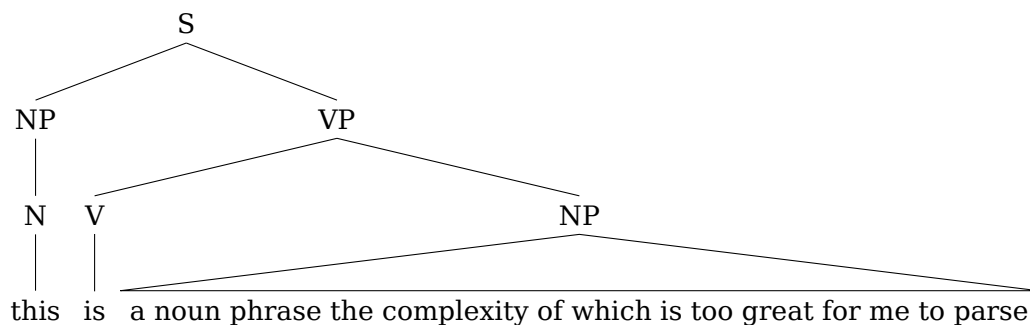
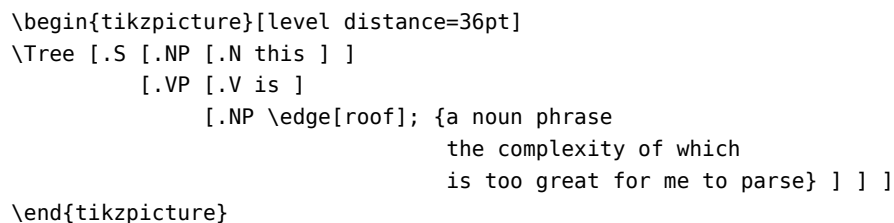
```
\edge [options] node [options] {label};
```

Typically one will use the `auto` option for edge labels, which places the label to the side of the edge.



The fact that `auto=left` draws a label on the right and `auto=right` draws a label on the left makes sense if you think about the tree growing from the root to the leaves.

There is a predefined style that draws a “roof” over a node, like Qtree’s `\qroof`:



<sup>7</sup>Except that a TikZ `edge from parent` comes after the child node. I thought it was more logical to put it before.

## 4 Qtree compatibility

For basic trees, `tikz-qtree` can be used as a drop-in replacement for `Qtree`, but most of `Qtree`'s advanced features are either not accessed in the same way in `tikz-qtree` or not implemented at all. There is a package `tikz-qtree-compat` which can be loaded to improve compatibility. Supported so far are:

- Superscripts and subscripts outside of math mode, and `\automath`
- The `\0`, `\1`, and `\2` commands, and `\qtreeprimes`
- The `\qroof` command

For unsupported commands, warning messages are printed, but at least your file should compile.

## Acknowledgements

This was all Dan Gildea's idea. Thanks to Alan Munn for his very helpful suggestions.

## Contact

Please send suggestions to [chiang@isi.edu](mailto:chiang@isi.edu).