

# Tree drawing tutorial

Suzana Fong  
[sznfong@mit.edu](mailto:sznfong@mit.edu)

June, 2021

## 1 Drawing trees with an online generator

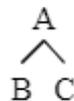
We can generate syntactic trees online for free. Two options are provided below:

- Syntax Tree Generator: <http://mshang.ca/syntaxtree/>
- RSyntaxTree: <https://yohasebe.com/rsyntaxtree/>

Both generators use bracket notation. The codes below can be used in either generator. The Syntax Tree Generator creates the tree as we write the code; in RSyntaxTree, we have to click on *draw PNG*.

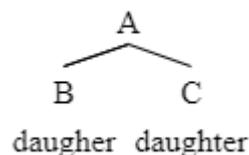
Each pair of brackets represents a node. Use a letter right after the left bracket to label the node. Daughters of this node are also enclosed in brackets. Consider, for instance, that we want to draw a node A, which branches into the daughters B and C. These are the code and resulting diagram:

```
[A [B] [C]]
```



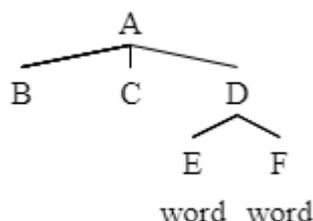
We can also include terminal nodes to the daughters B and C:

```
[A [B daughter] [C daughter]]
```



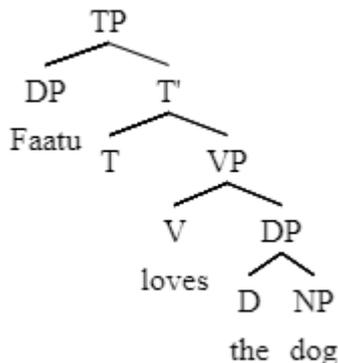
We can draw *n*-ary trees (not just binary, as in the examples so far). We can also create branching nodes from a node that is itself a daughter of another node. For example, we can draw a tree with the topmost node A, which then branches into B, C, and D. D itself branches into E and F:

```
[A [B] [C] [D [E word] [F word]]]
```



Here is a tree that looks more like what we may see in a syntax course:

[TP [DP Faatu] [T' [T] [VP [V loves] [DP [D the] [NP dog]]]]]



## 2 Drawing trees in $\text{\LaTeX}$

In this section, we will see how we can draw trees with  $\text{\LaTeX}$ , using the package `tikzpicture`. Before that, I will briefly talk about  $\text{\LaTeX}$  editors.

### 2.1 Getting started

I recommend using [Overleaf](#), an online  $\text{\LaTeX}$  editor, though you can also download some editor into your computer. You can use Overleaf for free. There are certain limitations for free accounts (e.g. the number of people you can collaborate with on a project), but it suffices to draw trees and type simple documents (e.g. homework, final paper).

Overleaf loads the preamble for you. You can see a minimal working example in the [Appendix](#) below.

### 2.2 Drawing trees with `tikz-qtrees`

What I am about to say below is quite basic. For a more thorough description of the package, consult its [documentation](#).

`tikz-qtrees` can also work with bracket notation, just like the online generators described earlier. Use the command `\Tree` and use a dot `.` before the label. Remember to include a space between a label and a right bracket.

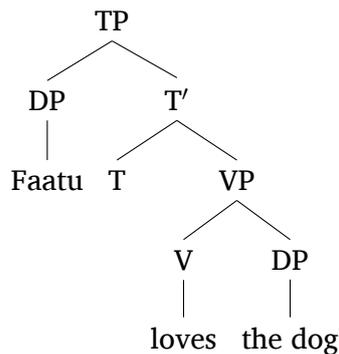
```
\Tree [.A [.B ] [.C ]]
```



Our tree for *Faatu loves the dog* can be drawn as follows:

```
\Tree [.TP [.DP Faatu ] [.T'$ [T] [.VP [.V loves ] [.DP the{ }dog ] ] ] ]
```

This will generate the following diagram:



tikz in general is a very powerful package and it allows for a fine-grained control of the diagram you're drawing. The code above is a simplified syntax. You can also use the following environment:

```

\begin{tikzpicture}
  \Tree [.TP [.DP Faatu ] [.T'$ [ .T ] [.VP [.V loves ] [.DP the{ }dog ] ] ] ]
\end{tikzpicture}

```

It will give rise to the same tree above for *Faatu loves the dog*.

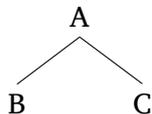
This environment will allow us to modify certain properties of the trees we are drawing, including the length of the branches. Use `\tikzset{...}` and then `level distance` for the vertical space between a node and its daughter(s) and `sibling distance` for the horizontal distance between two nodes that have the same mother.

```

\begin{tikzpicture}
  \tikzset{level distance=3em, sibling distance=4em}
  \Tree [.A [.B ] [.C ] ]
\end{tikzpicture}

```

This code produces the following tree:



We can also draw movement arrows with this environment. Enclose the nodes to be connected by the arrow with `\node(label){...};`, where 'label' is the name of you will assign to the node. Inside the curly brackets is the node as it appears the the diagram you will generate. Don't forget the semicolon! Finally, use a `\draw` command to draw the arrow.

Consider the tree below. Suppose we want to draw and arrow from G to B.

```

\begin{tikzpicture}
  \tikzset{level distance=3em, sibling distance=3em}
  \Tree [.A [.B ] [.C [.D ] [.E [.F ] [.G ] ] ] ]
\end{tikzpicture}

```

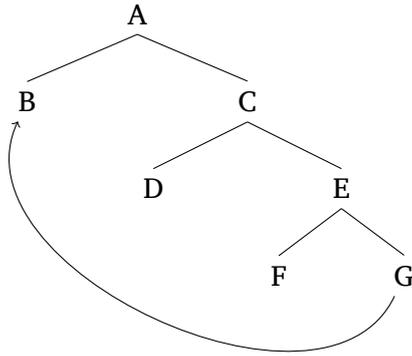
First, enclose B and G `\node(label){...};`. For convenience, I will label these nodes *b* and *g*, respectively. Then add a `\draw` command connecting these two points. Don't forget a semicolon at the end!

This is the final result:

```

\begin{tikzpicture}
  \tikzset{level distance=3em, sibling distance=3em}
  \Tree [.A [. \node(b){B}; ] [.C [.D ] [.E [.F ] [. \node(g){G}; ] ] ] ]
  \draw[<-] (b) to [bend right=90] (g);
\end{tikzpicture}

```



Inside the square brackets, you can add details of how you want your arrow. For instance, a dashed arrow with a solid triangle as its tip would look like the following:

```
\draw[dashed,latex-]
```

For a detailed description of `tikz`'s uses for linguistics, see this phenomenal [manual](#) by James Crippen.

### 2.3 Another useful package: `gb4e`

The package `gb4e` is used for numbered examples. This is the package I will illustrate here. Another packages used for the same purpose are `linguex` and `expex`. These packages allow us to automatically number data and refer to them in the main text. They are also handy in providing alignment for examples with glosses.

With `gb4e`, this is the code for a single example:

```
\begin{exe}
  \ex{This is a basic example.}
\end{exe}
```

You can use `\label{...}` to name an example and then refer back to it in the main text, using `\ref{...}`. With `gb4e`, you have to insert parentheses manually.

```
\begin{exe}
  \ex{This is a another basic example, but with a label.}
  \label{Simplex}
\end{exe}
```

In example (`\ref{Simplex}`), ...

As mentioned, examples are numbered automatically, so you don't have to worry about deleting or inserting examples. The same holds for crossreferencing examples in the main text, as long as you are using `\label{...}` and (`\ref{...}`).

You can also create a paradigm by embedding `xlist` into `exe`.

```
\begin{exe}
  \ex{\begin{xlist}
    \ex{First example.}
    \ex{Second example.}
    \ex{Third example.}
  \end{xlist}}
\end{exe}
```

You can label individual items and refer to them in the main text too.

Finally, you can also use glosses with `gb4e`. Make sure to use `\gll` before the example and to skip lines with `\\`.

```
\begin{exe}
  \ex{\gll   Esse é um exemplo em português brasileiro.\\
      this be.\textsc{pres.3sg} one example in Portuguese Brazilian\\
      `This is an example in Brazilian Portuguese.'}
\end{exe}
```

Conveniently, the word-by-word alignment is also automatic.

## Appendix: A minimal example

```
\documentclass{article}
\usepackage[utf8]{inputenc}
```

```
\usepackage{tikz-qtrees}
\usepackage{gb4e}
```

```
\title{Title}
\author{Your name}
\date{June 2021}
```

```
\begin{document}
```

```
\maketitle
```

The sentence (`\ref{faatlovdog}`) is an example of an English sentence.

```
\begin{exe}
  \ex{Faatu loves the dog.}
  \label{faatlovdog}
\end{exe}
```

It can be diagrammed as follows:

```
\begin{figure}
  \centering
  \begin{tikzpicture}
    \Tree [ .TP [ .DP Faatu ] [ .T$'$ [ .T ] [ .VP [ .V loves ] [ .DP the{ }dog ] ] ] ]
  \end{tikzpicture}
  \caption{A tree diagram}
  \label{fig:faatlovdog}
\end{figure}

\end{document}
```