Mini TikZ Tutorial

Suzana Fong
sznfong@alum.mit.edu

May 21, 2023

Contents

In this tutorial, I list some basic trees and some basic decorations like different types of arrows. I learned a lot from this phenomenal tutorial by Dr. James Crippen: `https://lingbuzz.net/lingbuzz/003379`. **Most of the code in this tutorial comes from this manual**. Things I learned over the years from `https://tex.stackexchange.com/` are impossible to list and credit.

I claim no credit nor novelty in this tutorial. It just conveniently lists codes that are frequently used. It also provides a guide for students taking syntax courses who may want to learn how to draw trees with `tikz` without having to spend too much time learning its syntax.

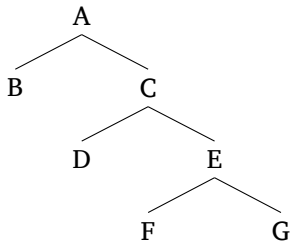
1 Trees and arrows

1.1 A basic head-initial tree. A head-initial tree can be drawn with the following code:

```
\begin{tikzpicture}
    [parent anchor=south,
    align=center,
    level distance=2em,
    anchor=north,
    sibling distance=5em,
    child anchor=north]
    \node {A}
    child {node {B}}
    child {node {C}
        child {node {D}}
        child {node {E}
            child {node {F}}
            child {node {G}}}}
    ;
\end{tikzpicture}
```

This is the output:

(1)



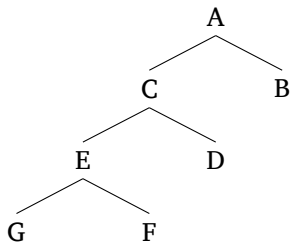
- Adjust the vertical distance between nodes with `level distance=2em` and the horizontal distance with `sibling distance=5em`.

1.2 A basic head-final tree. A head-final tree is drawn in the same way as the previous tree, except that the embedding is the left-most node.

```
\begin{tikzpicture}
    [parent anchor=south,
    align=center,
    level distance=2em,
    anchor=north,
    sibling distance=5em,
    child anchor=north]
    \node {A}
    child {node {C}
        child {node {E}
            child {node {G}}
            child {node {F}}}
        child {node {D}}}
    child {node {B}}
    ;
\end{tikzpicture}
```

This is the output:

(2)

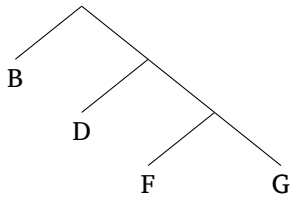


1.3 A tree without labels in branching nodes. In the trees above, each node (viz. the topmost node, the branching nodes, and the non-branching nodes) has a label. A tree can also be drawn that does not have labels in the branching nodes.

```
\begin{tikzpicture}
    [parent anchor=center,
    align=center,
    level distance=2em,
    anchor=north,
    sibling distance=5em,
    child anchor=north]
    \node {}
    child {node {B}}
    child {
        child {node {D}}
        child {
            child {node {F}}
            child {node {G}}}}
    ;
\end{tikzpicture}
```

This is the output:

(3)



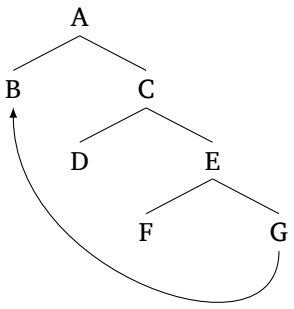
- Key specification: `parent anchor=center` (instead of `south`).
- Then delete `node {}` in a branching node (cf. the tree with labels in branching nodes).

1.4 A curved arrow connecting two nodes. There is more than one way to draw a curved arrow. Here is one:

```
\begin{tikzpicture}
    [parent anchor=south,
    align=center,
    level distance=2em,
    anchor=north,
    sibling distance=5em,
    child anchor=north]
    \node {A}
     child {node (b) {B}}
     child {node {C}
         child {node {D}}
         child {node {E}
             child {node {F}}
             child {node (g) {G}}}}
     ;
     \draw[latex-] (b) .. controls +(south:7em) and +(south:5em) .. (g);
\end{tikzpicture}
```

This is the output:

(4)



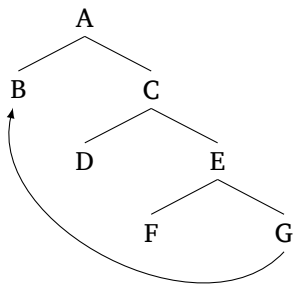
- Add labels to the nodes you want to connect with an arrow (e.g. `(b)` and `(g)`).
- Adjust the curve at `(south:7em)` and `(south:5em)`.

Here is another way:

```
\begin{tikzpicture}
    [parent anchor=south,
    align=center,
    level distance=2em,
    anchor=north,
    sibling distance=5em,
    child anchor=north]
    \node {A}
     child {node (b) {B}}
     child {node {C}
         child {node {D}}
         child {node {E}
             child {node {F}}
             child {node (g) {G}}}}
     ;
     \draw[latex-] (b) to [bend right=75] (g.south);
\end{tikzpicture}
```

This is the output:

(5)



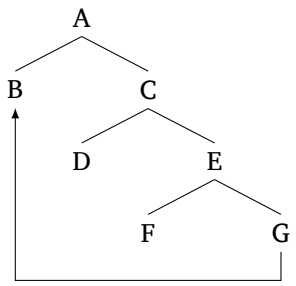
- Adjust the curve at `[bend right=75]`.

1.5 An angled arrow connecting B and G. Besides curved arrows, you can also draw angled arrows.

```
\begin{tikzpicture}
    [parent anchor=south,
    align=center,
    level distance=2em,
    anchor=north,
    sibling distance=5em,
    child anchor=north]
    \node {A}
     child {node (b) {B}}
     child {node {C}
         child {node {D}}
         child {node {E}
             child {node {F}}
             child {node (g) {G}}}}
     ;
     \draw[latex-] (b.south)--+(0,-6.5em)-|(g.south);
\end{tikzpicture}
```

This is the output:

(6)



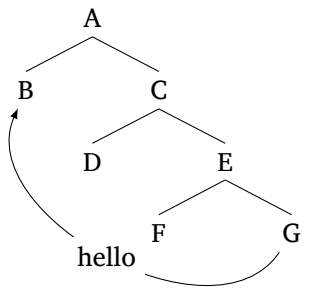
- The lowest point of the arrow can be adjusted at `(0,-6.5em)`.
- You can use [rounded corners = .25em] after `draw` to round the corners of the arrows. More precisely, add a comma , after `latex-`. This is the result: `[latex-, rounded corners=.25em]`.

1.6 Adding annotations to a curved arrow. TikZ allows for decorations in the arrows. Here is how to add *hello* to a curved arrow:

```
\begin{tikzpicture}
    [parent anchor=south,
    align=center,
    level distance=2em,
    anchor=north,
    sibling distance=5em,
    child anchor=north]
    \node {A}
     child {node (b) {B}}
     child {node {C}
         child {node {D}}
         child {node {E}
             child {node {F}}
             child {node (g) {G}}}}
     ;
     \draw[latex-] (b) to [bend right=85] node [anchor=center,pos=0.5,fill=white] {hello} (g);
\end{tikzpicture}
```

This is the output:

(7)



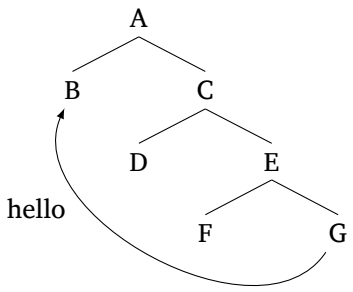
- Notice the `[fill=white]` at the node where *hello* was added.
- Instead of *hello*, you can use a symbol like ✗. You will need the package `pifont` in the preamble. You may want to delete `[fill=white]` in this case.
- Here is a comprehensive list of LaTeX symbols: `https://tug.ctan.org/info/symbols/comprehensive/symbols-a4.pdf`.

It is also possible to change the positioning of the annotation of the arrow. For instance:

```
\begin{tikzpicture}
    [parent anchor=south,
    align=center,
    level distance=2em,
    anchor=north,
    sibling distance=5em,
    child anchor=north]
    \node {A}
     child {node (b) {B}}
     child {node {C}
         child {node {D}}
         child {node {E}
             child {node {F}}
             child {node (g) {G}}}}
     ;
     \draw[latex-] (b) to [bend right=85] node [midway,pos=0.25,xshift=-1.5em] {hello} (g);
\end{tikzpicture}
```

This is the output:

(8)

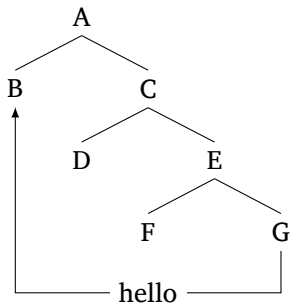


1.7 Adding annotations to an angled arrow. Annotation can also be added to angled arrows:

```
\begin{tikzpicture}
    [parent anchor=south,
    align=center,
    level distance=2em,
    anchor=north,
    sibling distance=5em,
    child anchor=north]
    \node {A}
     child {node (b) {B}}
     child {node {C}
         child {node {D}}
         child {node {E}
             child {node {F}}
             child {node (g) {G}}}}
     ;
     \draw[latex-] (b.south)--+(0,-7em)-|(g.south)
     node [anchor=center,pos=0.25,fill=white] {hello};
\end{tikzpicture}
```

This is the output:

(9)

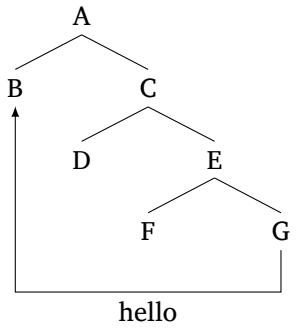


Just as in the curved arrow, the position of the annotation in angled arrows can also be modified. There is the code for annotation below the arrow:

```
\begin{tikzpicture}
    [parent anchor=south,
    align=center,
    level distance=2em,
    anchor=north,
    sibling distance=5em,
    child anchor=north]
    \node {A}
     child {node (b) {B}}
     child {node {C}
         child {node {D}}
         child {node {E}
             child {node {F}}
             child {node (g) {G}}}}
     ;
     \draw[latex-] (b.south)--+(0,-7em)-|(g.south)
     node [anchor=north,pos=0.25] {hello};
\end{tikzpicture}
```

This is the output:

(10)

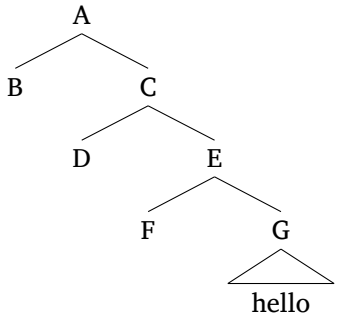


1.8 Triangle. If you want to abbreviate a node with a triangle, this is the code:

```
\begin{tikzpicture}
    [parent anchor=south,
    align=center,
    level distance=2em,
    anchor=north,
    sibling distance=5em,
    child anchor=north]
    \node {A}
    child {node {B}}
    child {node {C}
        child {node {D}}
        child {node {E}
            child {node {F}}
            child {node {G}
            [sibling distance=2em]
            child {coordinate (sideleft)}
            child {node {hello} edge from parent[draw=none]}
            child {coordinate (sideright)}}}}
    ;
    \draw (sideleft) -- (sideright);
\end{tikzpicture}
```

This is the output:

(11)



- Identify the node where you want your triangle, e.g. G:

```
child {node {G}}
```

- Add the code below right before the closing bracket `}`:

```
child {coordinate (sideleft)}
child {node {TEXT} edge from parent[draw=none]}
child {coordinate (sideright)}
```

- You can control the size of the triangle with `[sibling distance=2em]`, which should be placed before the code for the sides of the triangle.
- Add the following code after the tree to draw a horizontal line as the base of the triangle:

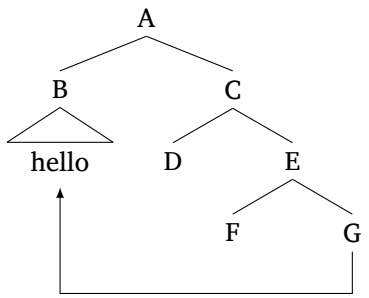
```
\draw (sideleft) -- (sideright);
```

You can also draw an arrow that targets a triangulated node:

```
\begin{tikzpicture}
    [parent anchor=south,
    align=center,
    level distance=2em,
    anchor=north,
    sibling distance=6.5em,
    child anchor=north]
    \node {A}
    child {node (b) {B}
    [sibling distance=2em]
    child {coordinate (sideleft)}
    child {node {hello} edge from parent[draw=none]}
    child {coordinate (sideright)}}
    child {node {C}
    [sibling distance=4.5em]
        child {node {D}}
        child {node {E}
            child {node {F}}
            child {node (g) {G}}}}
    ;
    \draw (sideleft) -- (sideright);
     \draw[latex-] ([yshift=-3em]b.south)--+(0,-5em)-|(g.south);
\end{tikzpicture}
```

This is the output:

(12)



- Notice the addition of `[yshift=-3em]` at the *B* node where the triangle is. This shifts the endpoint of the arrow to a lower position, so it won't go over the triangle.
- `[sibling distance=6.5em]` at the beginning of the TikZ environment sets the distance between the nodes for the overall tree. You can change the distance between individuals nodes, e.g. the `[sibling distance=4.5em]` below node `C`.

1.9 Linear representation with arrows. Besides hierarchical trees, it is possible to use TikZ to represents sentences linearly.

```
\tikzstyle{every picture}+=[remember picture, inner sep=0pt, baseline, anchor=base]%

{}[stuff \tikz\node(b){B}; [more stuff [even more stuff \tikz\node(g){G};]]]

\begin{tikzpicture}[overlay]
    \draw[latex-]([yshift=-.5em]b.south)--+(0,-1.5em)-|
    ([yshift=-.5em]g.south);
    \end{tikzpicture}
```

This is the output:

(13) [stuff B [more stuff [even more stuff G]]]

- The element `E` in the linear representation you want to connect with an arrow must be placed in this code below. Note the label `(e)` and the semicolon `;` at the end.

```
\tikz\node(e){E};
```

- The code below places the arrow lower down the elements it connects (i.e. it creates vertical space. Use + and - to shift the arrow up or down, respectively (i.e. in the *y* axis).

```
[yshift=-.5em]
```

- The code below creates horizontal space. This is particularly useful when the same node is connected by two arrows:

```
[xshift=-.5em]
```

- Use + and - to shift the arrow leftwards or rightwards, respectively (i.e. in the *x* axis).

Another example, with more than one node being connected:

```
\tikzstyle{every picture}+=[remember picture,
    inner sep=0pt,
    baseline, anchor=base]%
{}[stuff \tikz\node(b){B}; [more stuff \tikz\node(a){A}; [even more stuff \tikz\node(g){G};]]]
\begin{tikzpicture}[overlay]
    \draw[-latex]([yshift=-.5em]g.south)--+(0,-1em)-|
    ([yshift=-.5em,xshift=0.25em]a.south);
     \draw[-latex]([yshift=-.5em,xshift=-0.25em]a.south)--+(0,-1em)-|
    ([yshift=-.5em]b.south);
\end{tikzpicture}
```

This is the output:

(14) [stuff B [more stuff A [even more stuff G]]]

- Notice the addition of `[xshift=0.25em]` and `[xshift=-0.25em]` to the nodes *a* in the `draw` commands.
- They shift the arrow landing at or coming from this node rightwards and leftwards, respectively.

2 Other Useful Resources

- To align the top of the tree with the top of a numbered example, use the following command:

```
\leavevmode\vadjust{\vspace{-\baselineskip}}\newline
```

  For example, using `gb4e` to number data points:

```
\begin{exe}
    \ex{\leavevmode\vadjust{\vspace{-\baselineskip}}\newline
    \begin{tikzpicture}
    [parent anchor=south,
    align=center,
    level distance=2em,
    anchor=north,
    sibling distance=5em,
    child anchor=north]
    \node {A}
    child {node {B}}
    child {node {C}}
    ;
    \end{tikzpicture}}
\end{exe}
```

- Using symbols and special characters: this document uses a `fontspec` font. It allows you to input symbols like IPA symbols directly, without usint the package `tipa`, e.g. [ɸβθðʃʒʂʐ].
  - This is a useful IPA keyboard: `https://westonruter.github.io/ipa-chart/keyboard/`.
  - This website has symbols used in few different writing systems, e.g. Portuguese: `https://portuguese.typeit.org/`.

  If you are using a `fontspec` font, **compile your file with XeLaTeX**! On Overleaf: click on 'Menu' on the upper left corner, then go to 'Compiler' and choose XeLaTeX.

- Multiple columns with `minipage`:

```
\begin{minipage}[t]{0.5\textwidth}
    STUFF IN FIRST COLUMN
\end{minipage}%
\begin{minipage}[t]{0.5\textwidth}
    STUFF IN SECOND COLUMN
\end{minipage}%
```

  This can be a better option than `multicols` because you can adjust the size of each column. In the code above, the mini pages are of equal size (50/50), but you can change them to e.g. 55/45 by replacing the the instances of `0.5` above with `0.55` and `0.45`.