A Quick TikZ Guide for Modal Logicians

Alex Kocurek

January 28, 2017 (v. 2)

The following is a very brief guide for making diagrams in TikZ. This guide will be particularly geared towards those making modal models (state machines, automata, etc.). The best way to learn TikZ is by looking at a bunch of examples, but if you'd like a systematic guide to TikZ, I highly recommend checking out the official TiKZ manual (it's quite large, so I'd only recommend reading, say, sections 2–3 to start).

§1 Setup

To start, include the following lines in the preamble of your .tex file (before `\begin{document}`):

```
\usepackage{tikz}
\usetikzlibrary{positioning,arrows,calc}

\tikzset{
modal/.style={>=stealth',shorten >=1pt,shorten <=1pt,auto,node distance=1.5cm,
semithick},
world/.style={circle,draw,minimum size=0.5cm,fill=gray!15},
point/.style={circle,draw,inner sep=0.5mm,fill=black},
reflexive above/.style={->,loop,looseness=7,in=120,out=60},
reflexive below/.style={->,loop,looseness=7,in=240,out=300},
reflexive left/.style={->,loop,looseness=7,in=150,out=210},
reflexive right/.style={->,loop,looseness=7,in=30,out=330}
}
```

All I've done here is make a bunch of useful "styles" of nodes and arrows. We'll see how to tweak them later on, if you wish, but this just makes the exposition easier.

§2 Nodes

§2.1 Creating Nodes

The first step to drawing any modal diagram is to draw the worlds in your model. With the styles above, this is pretty easy. You just use the command `\node`, which has the following form (don't forget the semicolon at the end! TikZ crashes otherwise):

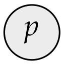
```
\node[⟨options⟩] (⟨name⟩) [⟨more options⟩] {⟨text⟩};
```

The ⟨*options*⟩ parameters are (of course) optional; you don't need to specify them, though most of the time, we will. The ⟨*name*⟩ parameter is a unique name that you'll want to give your node, so that you can refer back to it within the diagram. The ⟨*text*⟩ parameter is written inside the node.

Example 2.1: *Hello World*

```
\begin{center}
\begin{tikzpicture}[modal]

  \node[world] (w) {};

\end{tikzpicture}
\end{center}
```

Example 2.2: *Hello p-World*



```
\begin{center}
\begin{tikzpicture}[modal]

  \node[world] (w) {$p$};

\end{tikzpicture}
\end{center}
```

The option `world` can be replaced with `point` to draw a point.

Example 2.3: *Hello Point*

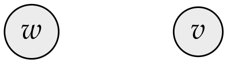
```
\begin{center}
\begin{tikzpicture}[modal]

  \node[point] (w) {};

\end{tikzpicture}
\end{center}
```

Note: from now on, I will not include the `center` and `tikzpicture` environments unless they are needed; I will simply write the body of the environment to save space.

§2.2 Positioning

Now, let's add more nodes. The position of the first node doesn't need to be specified, but the position of the rest of the nodes do. It's fairly easy to specify the relative locations of nodes.

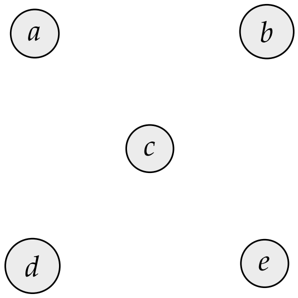
Example 2.4: *Hello Worlds*



```
\node[world] (w) {$w$};
\node[world] (v) [right=of w] {$v$};
```

You can replace `right` with `above`, `below`, `left`, `above right`, `above left`, etc.

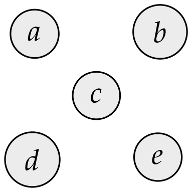
Example 2.5: *Hello Five Worlds*



```
\node[world] (c) {$c$};
\node[world] (a) [above left=of c] {$a$};
\node[world] (b) [above right=of c] {$b$};
\node[world] (d) [below left=of c] {$d$};
\node[world] (e) [below right=of c] {$e$};
```

There are two ways to adjust the distances of the nodes. First, one can adjust the node distance for all nodes at once by including it as an extra parameter after `modal`.

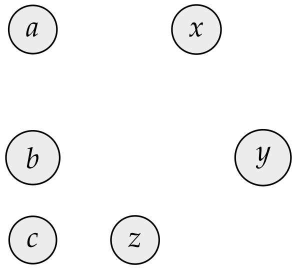
Example 2.6: *Adjusting All Distances*



```
\begin{tikzpicture}[modal,node distance = 5mm]

  \node[world] (c) {$c$};
  \node[world] (a) [above left=of c] {$a$};
  \node[world] (b) [above right=of c] {$b$};
  \node[world] (d) [below left=of c] {$d$};
  \node[world] (e) [below right=of c] {$e$};

\end{tikzpicture}
```

Second, you can also specify the exact distance your node is from another node before the `of`.

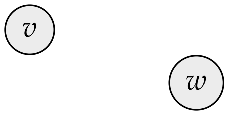
Example 2.7: *Specifying Distances Individually*



```
\node[world] (a) {$a$};
\node[world] (b) [below=1cm of a] {$b$};
\node[world] (c) [below=4mm of b] {$c$};

\node[world] (x) [right=of a] {$x$};
\node[world] (y) [right=2.3cm of b] {$y$};
\node[world] (z) [right=7mm of c] {$z$};
```

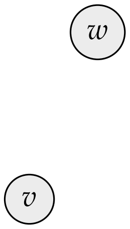
You can also "shift" the position of nodes along the x-axis or y-axis using `xshift` and `yshift` respectively.

Example 2.8: *Shifting Nodes Vertically*



```
\node[world] (w) {$w$};
\node[world] (v) [left=of w,yshift=7mm] {$v$};
```

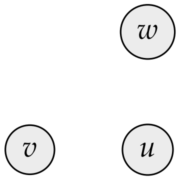
Example 2.9: *Shifting Nodes Horizontally*



```
\node[world] (w) {$w$};
\node[world] (v) [below=of w,xshift=-9mm] {$v$};
```

Finally, you can also align nodes relative to some other nodes using `at (`⟨*horizontal*⟩ `-|` ⟨*vertical*⟩`)` or `at (`⟨*vertical*⟩ `|-` ⟨*horizontal*⟩`)`.

Example 2.10: *Aligning Nodes*



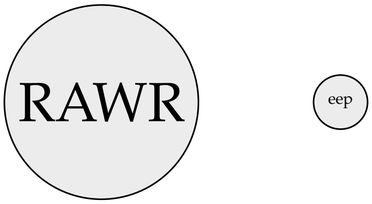
```
\node[world] (w) {$w$};
\node[world] (v) [below left=of w] {$v$};

\node[world] (u) at (v -| w) {$u$};
```

The "`at (v -| w)`" part tells TikZ to align the node `u` horizontally with `v` and vertically with `w`. That way, you don't have to fritz with the exact distances between nodes.

§2.3 Size of Node

The size of a node is in part determined by the material inside the node. This means the size of the node can vary if you put too much into ⟨*text*⟩ parameter.

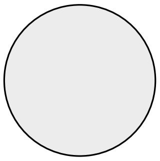
Example 2.11: *The Downside of Autosize*



```
\node[world] (tiny) {\tiny eep};
\node[world] (huge) [left=of tiny] {\huge RAWR};
```

You can change the size of a single node using either `minimum size`.

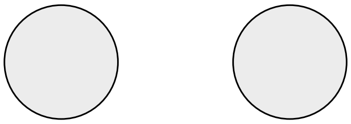
Example 2.12: *Hello Big World*



```
\node[world,minimum size=2cm] (w) {};
```

If you want to adjust the size of all of the worlds at once, you can do so using something like the following:

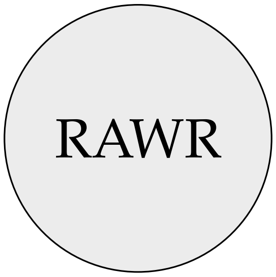
Example 2.13: *Adjusting Size of All Worlds*



```
\begin{tikzpicture}[modal,world/.append style=
{minimum size=1.5cm}]

  \node[world] (w) {};
  \node[world] (v) [right=of w] {};

\end{tikzpicture}
```

Another problem. Notice in **Example 2.11** that even though the size of the node adjusted as the text inside got bigger, the space between the edge of the node and the text stayed the same. To fix this, you can add buffer between the edge of the node and the text inside:

Example 2.14: *Adding Buffer*

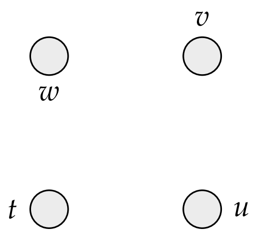


```
\node[world,inner sep=5mm] (w) {\huge RAWR};
```

§2.4 Labels

Placing labels around a node is also done via the optional parameters.

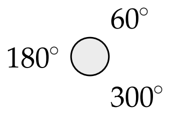
Example 2.15: *Hello Labeled Worlds*



```
\node[world] (w) [label=below:{$w$}] {};
\node[world] (v) [label=above:{$v$},right=of w] {};
\node[world] (u) [label=right:{$u$},below=of v] {};
\node[world] (t) [label=left:{$t$},left=of u] {};
```

If you want the node to be at a more specific point around the node, you can specify it with degrees instead of things like above (negative degrees are allowed).

Example 2.16: *Angle Labels*



```
\node[world] (w) [label=60:{$60^\degree$},
                  label=180:{$180^\degree$},
                  label=300:{$300^\degree$}] {};
```

This also illustrates how to put multiple labels on a node. (The extra space isn't necessary; I just wrote it that way for readability.)

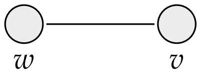
§3 Arrows and Lines

§3.1 Creating Paths

To draw a line from one node to another, one just needs a command of the following form:

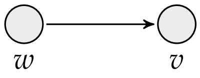
```
\path[⟨options⟩] (⟨start node⟩) edge[⟨options⟩] (⟨end node⟩);
```

Example 3.1: *Normal Edge*



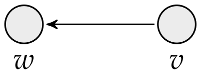
```
\node[world] (w) [label=below:$w$] {};
\node[world] (v) [label=below:$v$,right=of w] {};

\path[-] (w) edge (v);
```

Example 3.2: *One-Way Arrow*



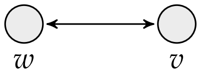
```
\node[world] (w) [label=below:$w$] {};
\node[world] (v) [label=below:$v$,right=of w] {};

\path[->] (w) edge (v);
```

Example 3.3: *Other-Way Arrow*



```
\node[world] (w) [label=below:$w$] {};
\node[world] (v) [label=below:$v$,right=of w] {};

\path[<-] (w) edge (v);
```

Example 3.4: *Two-Way Arrow*

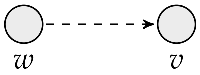


```
\node[world] (w) [label=below:$w$] {};
\node[world] (v) [label=below:$v$,right=of w] {};

\path[<->] (w) edge (v);
```

§3.2 Style

You can make a line or arrow `dashed` or `dotted` by including those words in the optional parameters (note: in my experience, dotted arrows do not show up in print very well).

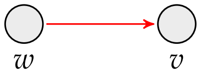
Example 3.5: *Dashed Arrow*



```
\node[world] (w) [label=below:$w$] {};
\node[world] (v) [label=below:$v$,right=of w] {};

\path[->] (w) edge[dashed] (v);
```

You can also color an arrow using `color` and then specifying your color.

Example 3.6: *Colored Arrow*

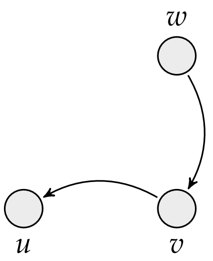


```
\node[world] (w) [label=below:$w$] {};
\node[world] (v) [label=below:$v$,right=of w] {};

\path[->] (w) edge[color=red] (v);
```

§3.3 Shape

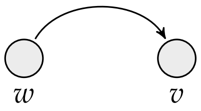
To make the arrow "bend", you can include `bend left` or `bend right` in the optional parameters. (`bend left` means from the tail end of the arrow looking towards the head, the arrow is bent to the left. That means that if the arrow goes from top to bottom, `bend left` will bend the arrow to the right of the page.) To make it bend more, you can write something like `bend left=`⟨*degree*⟩.

Example 3.7: *Bent Arrows*



```
\node[world] (w) [label=above:$w$] {};
\node[world] (v) [label=below:$v$,below=of w] {};
\node[world] (u) [label=below:$u$,left=of v] {};

\path[->] (w) edge[bend left] (v);
\path[->] (v) edge[bend right] (u);
```

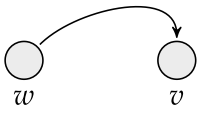
Example 3.8: *More Bent Arrow*



```
\node[world] (w) [label=below:$w$] {};
\node[world] (v) [label=below:$v$,right=of w] {};

\path[->] (w) edge[bend left=60] (v);
```

You can control where the tail emerges from the source node and where the head enters the target node using `in` and `out` and then specifying the angle in degrees.

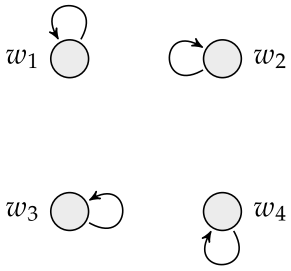
Example 3.9: *Enter and Exit Angles Specified*



```
\node[world] (w) [label=below:$w$] {};
\node[world] (v) [label=below:$v$,right=of w] {};

\path[->] (w) edge[out=45,in=90] (v);
```

Drawing reflexive arrows is easy with our predefined styles.

Example 3.10: *Reflexive Arrows*

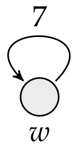


```
\node[world] (w1) [label=left:$w_1$] {};
\node[world] (w2) [label=right:$w_2$,right=of w1] {};
\node[world] (w3) [label=left:$w_3$,below=of w1] {};
\node[world] (w4) [label=right:$w_4$,right=of w3] {};

\path[->] (w1) edge[reflexive above] (w1);
\path[->] (w2) edge[reflexive left] (w2);
\path[->] (w3) edge[reflexive right] (w3);
\path[->] (w4) edge[reflexive below] (w4);
```

§3.4 Labelling Arrows

If you want to label an arrow, you can do so by including a `node` command after `edge`.

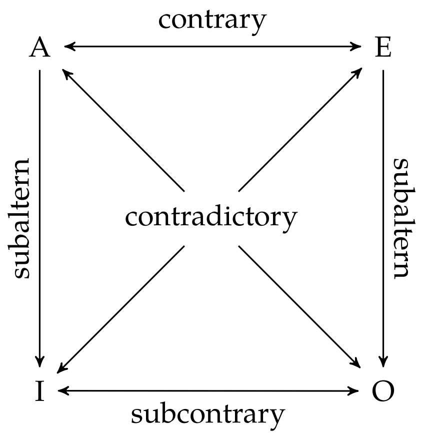
Example 3.11: *Labeled Reflexive Arrow*



```
\node[world] (w) [label=below:$w$] {};

\path[->] (w) edge[loop] node[above] {7} (w);
```

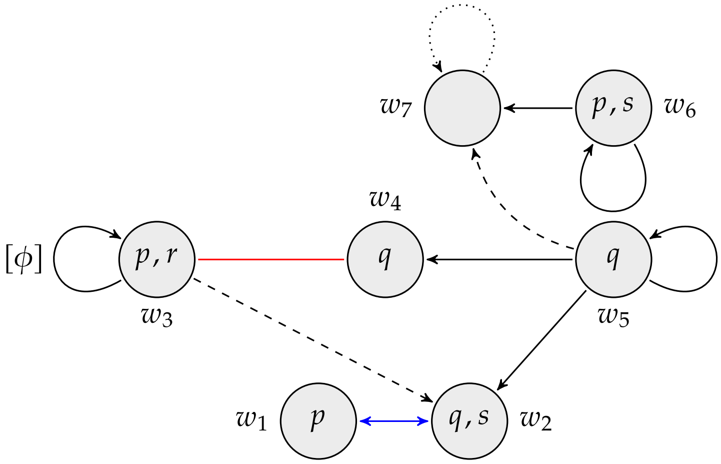
§3.5 Extended Examples

Example 3.12: *Square of Opposition*



```
\begin{tikzpicture}[modal,node distance=4cm]

  \node (A) {A};
  \node (E) [right=of A] {E};
  \node (I) [below=of A] {I};
  \node (O) [below=of E] {O};

  \coordinate (CENTER) at ($(A)!0.5!(O)$);
  \node (contra) at (CENTER) {contradictory};

  \path[<->] (A) edge node[above] {contrary} (E);
  \path[<->] (I) edge node[below] {subcontrary} (O);
  \path[->] (A) edge node[above,rotate=90] {subaltern} (I);
  \path[->] (E) edge node[above,rotate=-90] {subaltern} (O);
  \path[->] (contra) edge (A);
  \path[->] (contra) edge (E);
  \path[->] (contra) edge (I);
  \path[->] (contra) edge (O);
\end{tikzpicture}
```

Example 3.13: *Fancy Modal Diagram*



```
\begin{center}
\begin{tikzpicture}[modal,node distance=2cm,world/.append style={minimum
size=1cm}]

  \node[world] (w1) [label=left:$w_1$] {$p$};
  \node[world] (w2) [label=right:$w_2$,right of=w1] {$q,s$};
  \node[world] (w3) [label=below:$w_3$,above left=of w1]  {$p,r$};
  \node[world] (w4) [label=above:$w_4$,right=of w3] {$q$};
  \node[world] (w5) [label=below:$w_5$,right=of w4] {$q$};
  \node[world] (w6) [label=right:$w_6$,above of=w5] {$p,s$};
  \node[world] (w7) [label=left:$w_7$,left of=w6] {\phantom{$p$}};

  \path[<->] (w1) edge[color=blue] (w2);
  \path[->] (w3) edge[reflexive left] node[left] {$[\phi]$} (w3);
  \path[->] (w5) edge[reflexive right] (w5);
  \path[->] (w3) edge[dashed] (w2);
  \path[-]  (w4) edge[color=red] (w3);
  \path[->] (w5) edge (w4);
  \path[->] (w5) edge (w2);
  \path[->] (w5) edge[dashed,bend left] (w7);
  \path[->] (w6) edge (w7);
  \path[->] (w7) edge[reflexive above,dotted] (w7);
  \path[->] (w6) edge[reflexive below] (w6);

\end{tikzpicture}
\end{center}
```