

A Quick Guide to TikZ for Modal Logicians

Alexander W. Kocurek

February 28, 2023 (v. 4)

The following is a brief guide to making diagrams in TikZ. This guide is specifically written for those looking to learn how to draw modal diagrams (state machines, automata, etc.) in \LaTeX . Another great guide is [Justin Khoo's guide to drawing causal diagrams](#). If you'd like a more thorough guide to TikZ, you can check out [the official TikZ manual](#).

Contents

1 Setup	2
2 Nodes	2
2.1 Creating Nodes	2
2.2 Positioning	3
2.3 Size of Node	5
2.4 Labels	6
2.5 Splitting and Rotating	7
3 Arrows and Lines	8
3.1 Creating Paths	8
3.2 Style	9
3.3 Bending Arrows	10
3.4 Reflexive Arrows	10
3.5 Labelling Arrows	12
4 Extended Examples	13

1 Setup

To start, include the following lines in the preamble (before `\begin{document}`):

```
\usepackage{tikz}
\usetikzlibrary{arrows,calc,patterns,positioning,shapes}
\usetikzlibrary{decorations.pathmorphing}

\tikzset{
  modal/.style={>=stealth',shorten >=1pt,shorten <=1pt,auto,
    node distance=1.5cm,semithick},
  world/.style={circle,draw,minimum size=1cm,fill=gray!15},
  point/.style={circle,draw,fill=black,inner sep=0.5mm},
  reflexive/.style={->,in=120,out=60,loop,looseness=#1},
  reflexive/.default={5},
  reflexive point/.style={->,in=135,out=45,loop,looseness=#1},
  reflexive point/.default={25},
}
```

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, 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.

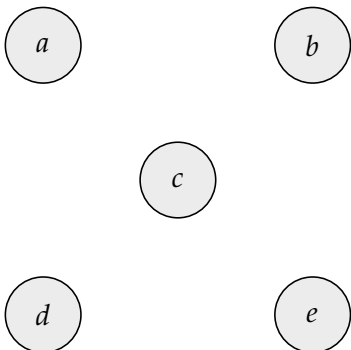
Example 2.4 (Hello Worlds)



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

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

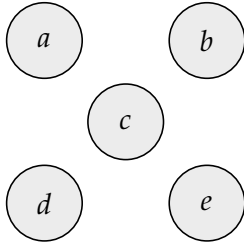
Example 2.5 (Hello Five Worlds)



```
\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, you 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)



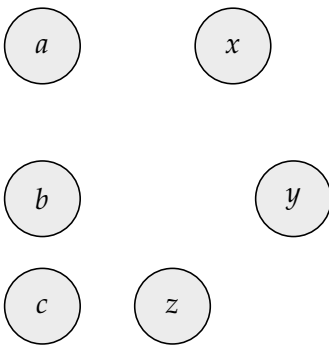
```
\begin{tikzpicture}[modal,node distance = 5mm]

\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 specify the exact distance your node is from another node before the `of`.

Example 2.7 (Specifying Distances Individually)

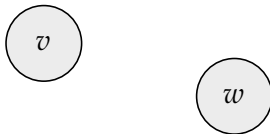


```
\node[world] (a) {$a$};
\node[world] (b) [below=1cm of a] {$b$};
\node[world] (c) [below=4mm of b] {$c$};

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

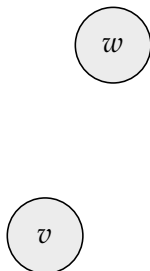
You can also shift the position of nodes along the x -axis or y -axis using `xshift` and `yshift`.

Example 2.8 (Shifting Nodes Vertically)



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

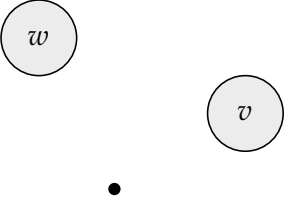
Example 2.9 (Shifting Nodes Horizontally)



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

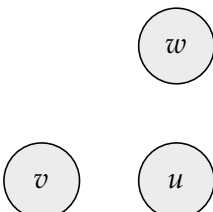
An alternative way to specify the position of a node is using `at <<position>>`. The `<position>` is a coordinate (either Cartesian or polar).

Example 2.10 (Positioning Nodes Absolutely)

	<pre>\node[point] {}; % at (0,0) \node[world] (w) at (-1,2) {\$w\$}; % (x,y) \node[world] (v) at (30:2) {\$v\$}; % (deg:radius)</pre>
---	---

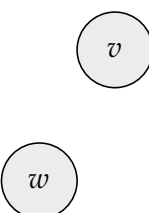
You can also align nodes relative to some other nodes using `at <<horizontal>> -| <vertical>>` or `at <vertical>> |- <horizontal>>`. The “`at (v -| w)`” says to align the node `u` horizontally with `v` and vertically with `w`.

Example 2.11 (Aligning Nodes)

	<pre>\node[world] (w) {\$w\$}; \node[world] (v) [below left=of w] {\$v\$}; \node[world] (u) at (v - w) {\$u\$};</pre>
---	---

If you want to position one node at an angle relative to another, here’s how you could do it:

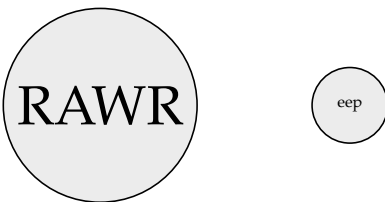
Example 2.12 (Positioning Nodes at Angles of Other Nodes)

	<pre>\node[world] (w) {\$w\$}; \node[world] (v) at (\$(w) + (60:2)\$) {\$v\$};</pre>
---	--

2.3 Size of Node

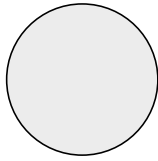
The size of a node is in part determined by the material inside the node. This means the size of the node can vary if you put too much into `<text>` parameter.

Example 2.13 (The Downside of Autosize)

	<pre>\node[world] (tiny) {\tiny eep}; \node[world] (huge) [left=of tiny] {\huge RAWR};</pre>
---	--

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

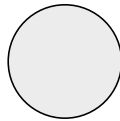
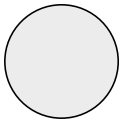
Example 2.14 (Hello Big World)



```
\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 2.15 (Adjusting Size of All Worlds)



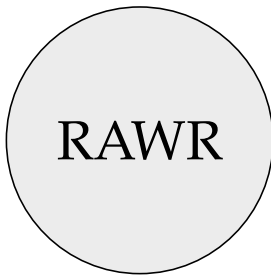
```
\begin{tikzpicture}[modal,world/.append style=
{minimum size=1.5cm}]

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

\end{tikzpicture}
```

Another problem: in [Example 2.13](#), 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 2.16 (Adding Buffer)



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

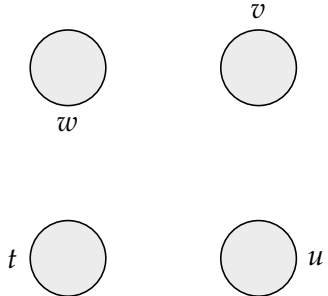
2.4 Labels

Placing labels around a node is also done via the optional parameter `label`. The syntax is:

```
label=<position>:{<content>}
```

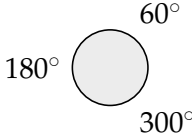
As before, the `<position>` can be `above`, `below`, `left`, or `right`.

Example 2.17 (Hello Labeled Worlds)

	<pre> \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] {}; </pre>
---	---

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 2.18 (Angle Labels)

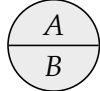
	<pre> \node[world] (w) [label=60:{\$60^\degree\$}, label=180:{\$180^\degree\$}, label=300:{\$300^\degree\$}] {}; </pre>
---	---

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

2.5 Splitting and Rotating

You can divide the interior of a node into different parts using `circle split` and `\nodepart`.

Example 2.19 (Divided World)

	<pre> \node[world,circle split] (w) {\$A\$ \nodepart{lower} \$B\$}; </pre>
---	--

If you want to color the different circle halves, you can do this, but you need to add this bit of code to the preamble (borrowed from [here](#)).

```

\makeatletter
\tikzset{circle split part fill/.style args={#1,#2}{%
alias=tmp@name, %
postaction={%
insert path={
\pgfextra{%
\pgfpointdiff{\pgfpointanchor{\pgf@node@name}{center}}%
{\pgfpointanchor{\pgf@node@name}{east}}%
\pgfmathsetmacro\insiderad{\pgf@x}
\fill[#1] (\pgf@node@name.base) ([xshift=-\pgflinewidth]\pgf@node@name.east) arc
(0:180:\insiderad-\pgflinewidth)--cycle;
\fill[#2] (\pgf@node@name.base) ([xshift=\pgflinewidth]\pgf@node@name.west) arc
(180:360:\insiderad-\pgflinewidth)--cycle;
}
}
}
\makeatother

```

```
}}}}
\makeatother
```

Example 2.20 (Colorfully Divided World)



```
\node[world,circle split,
circle split part fill={red!25,blue!25}] (w)
{$A$ \nodepart{lower} $B$};
```

Example 2.21 (Colorfully Rotated Divided World)



```
\node[world,rotate=45,circle split,
circle split part fill={red!25,blue!25}] (w)
{$A$ \nodepart{lower} $B$};
```

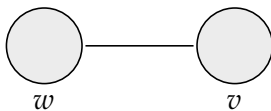
3 Arrows and Lines

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>);
```

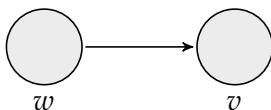
Example 3.1 (Normal Edge)



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

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

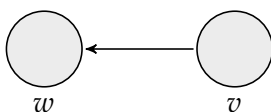
Example 3.2 (One-Way Arrow)



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

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

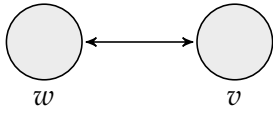
Example 3.3 (Other-Way Arrow)



```
\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)

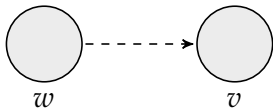


```
\node[world] (w) [label=below:$w$] {};  
\node[world] (v) [label=below:$v$,right=of w] {};  
  
\path[<->] (w) edge (v);
```

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).

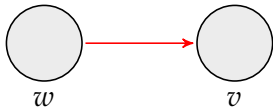
Example 3.5 (Dashed Arrow)



```
\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)

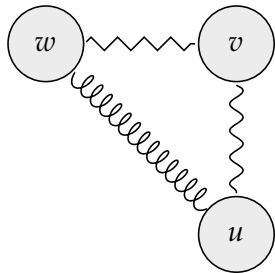


```
\node[world] (w) [label=below:$w$] {};  
\node[world] (v) [label=below:$v$,right=of w] {};  
  
\path[->] (w) edge[color=red] (v);
```

With the decorations TikZ library installed, we can also do more fancy stylistic arrows. For your convenience, I've defined three useful ones. Just add this to the preamble:

```
\tikzset{  
  coil/.style={decorate, decoration={coil,amplitude=4pt,segment length=5pt}},  
  snake/.style={decorate, decoration={snake}},  
  zigzag/.style={decorate, decoration={zigzag}}  
}
```

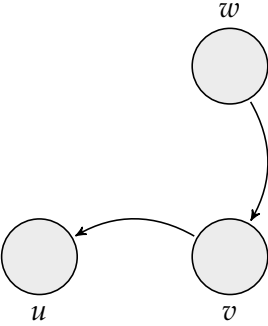
Example 3.7 (Arrows in Style)

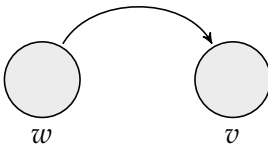


```
\node[world] (w) {$w$};  
\node[world] (v) [right=of w] {$v$};  
\node[world] (u) [below=of v] {$u$};  
  
\path (w) edge[zigzag] (v);  
\path (v) edge[snake] (u);  
\path (u) edge[coil] (w);
```

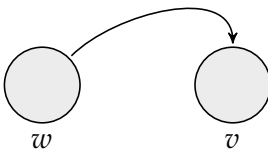
3.3 Bending Arrows

To make an 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 3.8 (Bent Arrows)	
	<pre>\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);</pre>

Example 3.9 (More Bent Arrow)	
	<pre>\node[world] (w) [label=below:\$w\$] {}; \node[world] (v) [label=below:\$v\$,right=of w] {}; \path[->] (w) edge[bend left=60] (v);</pre>

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.

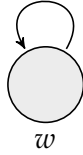
Example 3.10 (Enter and Exit Angles Specified)	
	<pre>\node[world] (w) [label=below:\$w\$] {}; \node[world] (v) [label=below:\$v\$,right=of w] {}; \path[->] (w) edge[out=45,in=90] (v);</pre>

Drawing reflexive arrows is easy with our predefined styles.

3.4 Reflexive Arrows

I've added some styles for reflexive arrows specifically. The simplest is `reflexive`.

Example 3.11 (Simple Reflexive Arrow)



```
\node[world] (w) [label=below:{$w$}] {};  
\path[->] (w) edge[reflexive] (w);
```

I had to make a different style for reflexive arrows on points, since otherwise they're too small.

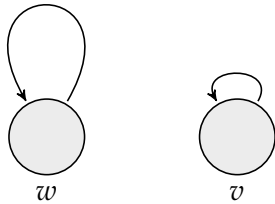
Example 3.12 (Simple Reflexive Arrow (Point))



```
\node[point] (w) [label=below:{$w$}] {};  
\path[->] (w) edge[reflexive point] (w);
```

If you want the loop to be bigger or smaller, you can do that by changing the `looseness` option. So you could write `looseness= $\langle n \rangle$` and choose an n to your liking. I've built in an optional argument for that into the reflexive arrow style (for `reflexive`, the default value is 5; for `reflexive point`, it's 25).

Example 3.13 (Adjusting the Size of the Reflexive Loop)

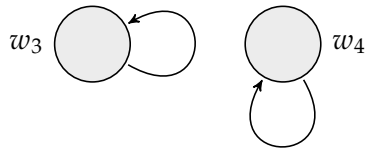
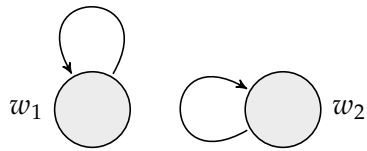


```
\node[world] (w) [label=below:{$w$}] {};  
\node[world] (v) [label=below:{$v$},right=of w] {};  
  
\path[->] (w) edge[reflexive=10] (w);  
\path[->] (v) edge[reflexive=3] (v);
```

You can also be more specific about whether you want the reflexive arrow to be above, below, left, or right of the node. Just add this to the preamble (obviously, you can tweak to your liking; and of course, you could add similar styles for `reflexive point`):

```
\tikzset{  
  reflexive above/.style={->,loop,in=120,out=60,looseness=#1},  
  reflexive above/.default={7},  
  reflexive below/.style={->,loop,in=240,out=300,looseness=#1},  
  reflexive below/.default={7},  
  reflexive left/.style={->,loop,in=150,out=210,looseness=#1},  
  reflexive left/.default={7},  
  reflexive right/.style={->,loop,in=30,out=330,looseness=#1},  
  reflexive right/.default={7}  
}
```

Example 3.14 (Positioning Reflexive Arrows)

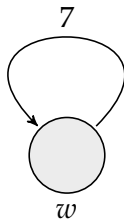


```
\node[world] (w1) [label=left:$w_1$] {};  
\node[world] (w2) [label=right:$w_2$,right=of w1] {};  
\node[world] (w3) [label=left:$w_3$,below=of w1] {};  
\node[world] (w4) [label=right:$w_4$,right=of w3] {};  
  
\path[->] (w1) edge[reflexive above] (w1);  
\path[->] (w2) edge[reflexive left] (w2);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

3.5 Labelling Arrows

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

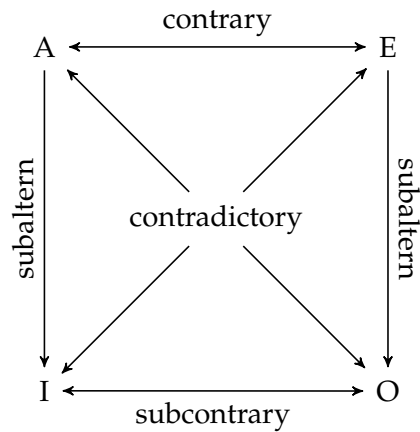
Example 3.15 (Labeled Reflexive Arrow)



```
\node[world] (w) [label=below:$w$] {};  
  
\path[->] (w) edge[loop] node[above] {7} (w);
```

4 Extended Examples

Example 4.1 (Square of Opposition)



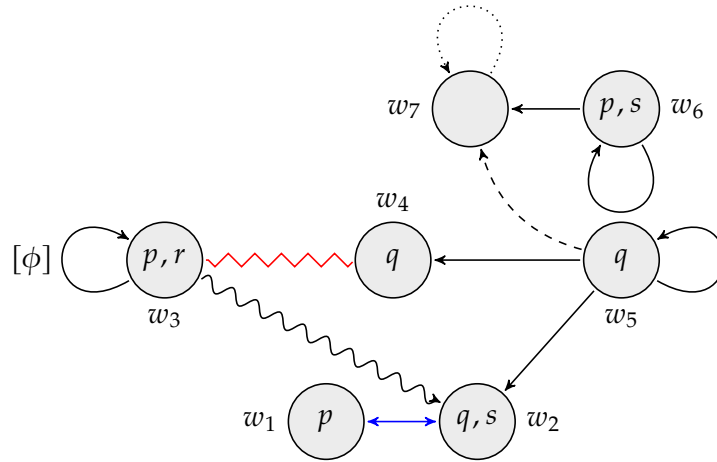
```
\begin{tikzpicture}[modal,node distance=4cm]

  \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 4.2 (Fancy Modal Diagram)



```
\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[snake] (w2);
\path[-] (w4) edge[color=red,zigzag] (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}
```