

Drawing Games and Diagrams in L^AT_EX*

Richard Loeza
University of Wisconsin
loeza@wisc.edu

September 25, 2013

This document introduces a number of ways to draw game matrices, decision and interaction trees, spatial models, and other diagrammatic representations of aspects of strategic behavior. The code can look a bit dense and intimidating at first, but once you get familiar with what each command (and each part of each command) is doing, these packages provide very flexible ways to draw just about any diagram you want.

1 Strategic Form Games

To make strategic form games, the ‘game’ environment (using the `.sgamevar` package) behaves much like tabular environments (in fact, you could just use tables instead if you’re sophisticated in how you manage the lines). But using this package does incorporate a few nice features: it automatically places lines in the appropriate places, it does the alignment and dimensions for you, and it centers the player and name labels horizontally and vertically. Syntactically, it behaves exactly like the tabular environment except:

- You use the ‘figure’ float instead of the ‘table’ float.
- The `\>` replaces `&` as the divider between cells.
- The initial operators are different - rather than `\begin{tabular}{l|cc}...\end{tabular}` it is:
`\begin{game}{rows}{columns}[player 1 label][player 2 label][figure label]...\end{game}`

		2		
		Left	Center	Right
1	Up	3,2	4,4	5,1
	Down	0,0	2,6	4, 3

		USSR						
		Retreat	Negotiate	Sanctions	Blockade	Airstrike	Invade	Nuclear
US	Retreat	2,2	2,6	1,5	4,3	2,3	3,4	3,7
	Negotiate	6,2	8,8	3,5	3,2	1,1	4,0	2,4
	Sanctions	5,1	5,3	2,2	2,4	0,4	2,5	0,1
	Blockade	3,4	2,3	4,2	2,2	7,6	4,7	0,1
	Airstrike	3,2	1,1	4,0	6,7	4,4	5,5	0,1
	Invade	4,3	0,4	5,2	7,4	5,5	6,6	0,3
	Nuclear	7,3	4,2	1,0	1,0	1,0	3,0	-5,-5

If you like, strategic games can be quite large

*This handout borrows heavily from the department’s 2010 L^AT_EX handout authored by Dave Ohls. The 2010 handout is available at <https://mywebspace.wisc.edu/ohls/web/teaching.html>

Most stylistic commands also work inside games. Using circles to indicate best responses is problematic because the circles tend to cut off some of the number¹.

				2				
				Rock	Paper	Scissors	Lizard	<i>Spock</i>
1	Rock		1,1	0,2	2,0	2,0	0,2	
	Paper		2,0	1,1	0,2	0,2	2,0	
	Scissors		0,2	2,0	1,1	2,0	0,2	
	Lizard		0,2	2,0	0,2	1,1	2,0	
	<i>Spock</i>		2,0	0,2	2,0	0,2	(1,1)	

A silly game

You can cluster multiple strategic games together by including them within the same ‘figure’ float, or by having them clustered. Here, I used the `\hspace{}` and `\vspace{}` commands to have them spaced how I wanted.

			2					2		
			Polisci	Econ				IR	Theory	
1	Polisci		4,4	2,1		1	Comparative	5,6	3,3	
	Econ		1,2	3,3			American	2,3	4,0	

(a) (c)

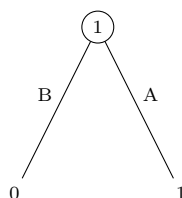
			2					2		
			Dayton	Mifflin				Batman	Superman	
1	Gorham		-4,-2	-2,-5		1	Luthor	2,1	3,4	
	Johnson		-1,-4	-3,-1			Joker	3,4	2,1	

(b) (d)

2 Extensive Form Games

Previous versions of this workshop used the `.egamesps` package to create extensive form games. That package makes creating extensive games simpler, but it is less easily manipulated and is extremely fragile/prone to errors. If you are interested in using `egames` to make extensive form games, review the 2010 LaTeX workshop handout, available at <https://mywebspace.wisc.edu/ohls/web/teaching.html>.

Instead, we are going to cover how to create extensive games using the ‘tikzpicture’ environment, from the `.tikz` package. The `tikz` package is designed for drawing in general, and is very flexible. The package can accommodate nearly any shape or arrangement you desire. The basic syntax of the coding language involves drawing a series of objects (lines, geometric shapes, text), specifying their location on a coordinate scale, and specifying the properties you’d like them to have (size, style, color, shading, special features).



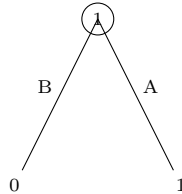
What the commands are doing:

- `[scale=1]` Scales the figure with the dimensions given (in centimeters). Increase (or decrease) the scale to expand (or shrink) the figure.

¹For those wondering: Scissors cuts paper, paper covers rock, rock crushes lizard, lizard poisons Spock, Spock smashes scissors, scissors decapitates lizard, lizard eats paper, paper disproves Spock, Spock vaporizes rock, rock crushes scissors.

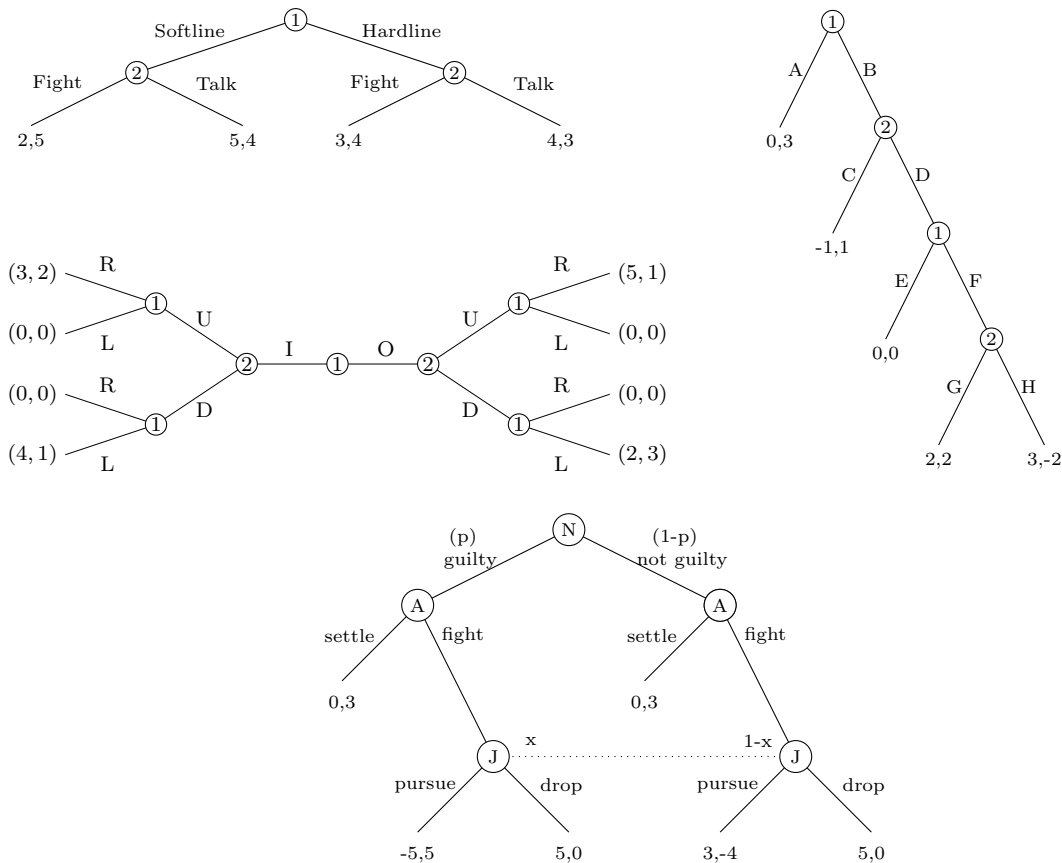
- `\draw (0,2) -- (1,0);`: Draws a straight line between coordinate points (0,2) and (1,0).
- `\draw (.7,1.1) node{\scriptsize{A}};`: Creates a node with the small text A at the coordinates (.7,1.1).
- `\filldraw[fill=white](0,2) circle (6pt) node{\scriptsize{1}};`: Creates a circle, 6 points in diameter and fills the inside white. It then creates a small 1 in the middle of the circle. Note that this will cover up anything under the filled circle.

With tikz, the order in which objects are rendered is important. A filled circle will go over anything at the same coordinates if it is listed after those things. Note how the above figure looks when the ordering is changed.



Note that every line in a tikzpicture ends in a semicolon. A common error is forgetting to put one in.

With these basic tools, you can make games in a variety of shapes.



`\draw[dotted] (3,1) -- (7,1);` creates a dotted line from (3,1) to (7,1).

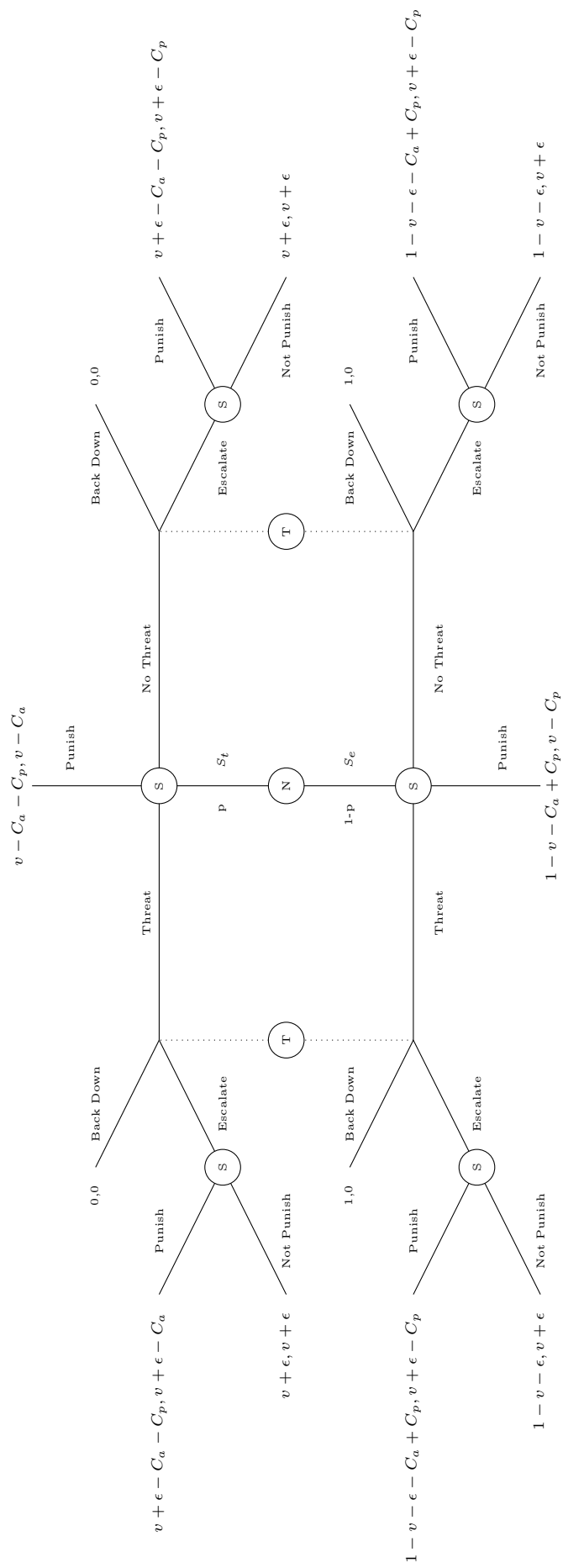
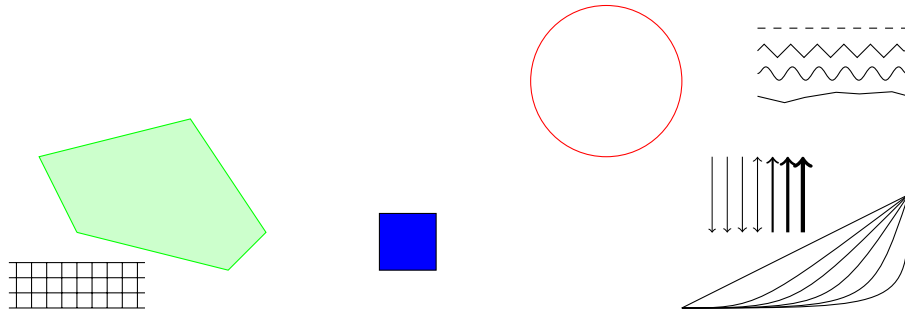


Figure 1: Big Ol' Model

3 Diagrams

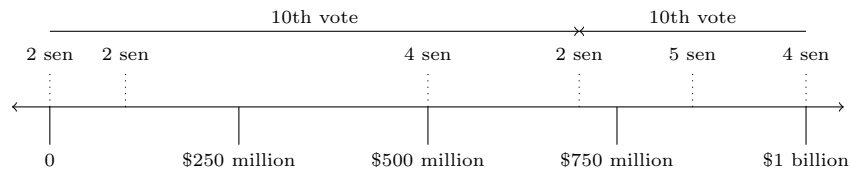
For some game theory applications, and for papers in general, you may want to draw diagrams or pictures - spatial models, functional forms, flow charts of processes, stylized examples of data, etc. Tikz's versatility comes in handy for these more complex types of drawings as well.



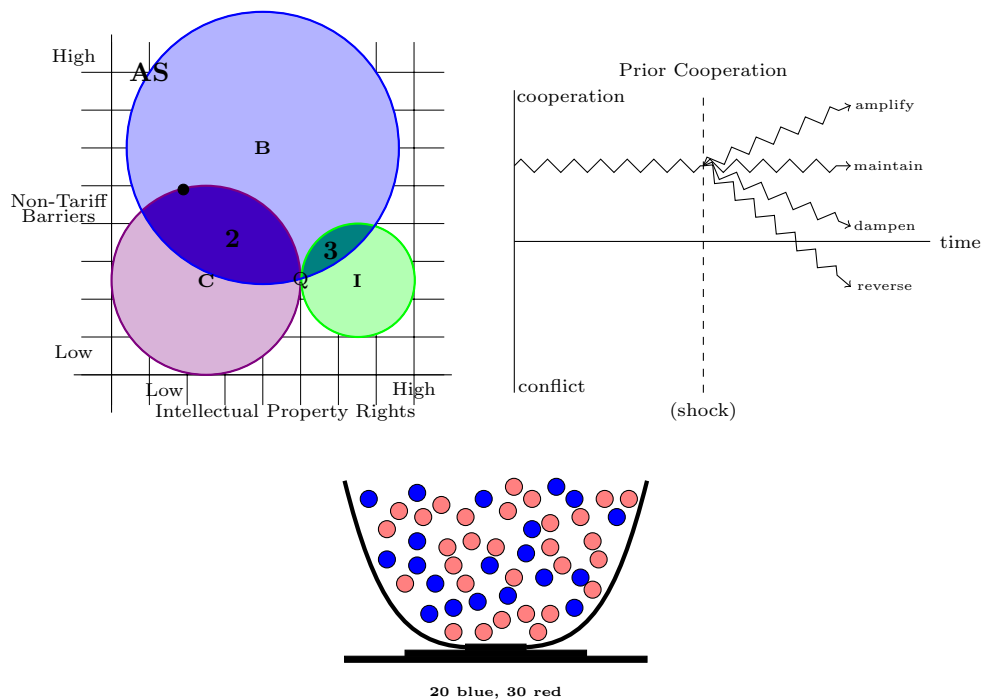
What the commands are doing:

- `\draw[red] (8,3) circle(1cm);`: Draws a circle colored red, centered at (8,3) and with radius of 1 centimeter.
- `\draw[fill=blue] (5,0.5) rectangle (5.75,1.25);`: Draws a rectangle (in this case a square) defined by corners at (5,0.5) and (5.75,1.25), and fills it blue.
- `\draw[green, fill=white!80!green] (1,1) -- (3,0.5) ... (0.5,2) -- (1,1);`: Draws a polygon region defined by the coordinates given, colored green and shaded a lighter green (80% white, 20% green).
- `\draw[->, ultra thin] ...`: Draws a series of lines of varying thicknesses with arrows on one or both ends.
- `\draw[dashed]`: Draws a dashed line.
- `\draw[decorate,decoration=zigzag]`: Draws an object where the lines are 'decorated' or modified in some way (zigzag, snake, bumpy, random steps, text, a series of various symbols, etc).
- `\draw (9,0) .. controls (12,0) .. (12,1.5);`: Draws a curved line going from (9,0) to (12,1.5) with a 'control point' at (12,0) that pulls it into a bend. The line starts out toward the control point, then curves to the end point.
- `\draw[step=.2] (0.1,0) grid (1.9,.6);`: Draws a grid with the specified step width covering the specified area. Note that if you want the grid closed (like the top and bottom) you should include limits divisible by the step width, and if you want the grid open (like the left and right) you should include limits which are not.

This can be used to create a one-dimensional spatial model with some key points highlighted and labeled:



You can extend this, along with many additional features, to draw anything else you need. It works quite well for anything you'd like to show on a coordinate axis, such as two-dimensional spatial models or patterns over time, as well as any other drawings or flow charts you need..



A few additional commands introduced here that are useful for diagrams with shapes that interact with each other:

- `\def\chinacircle{(1.25,1.25) circle(1.25cm)};`: 'Defines' an object and names it without drawing it (yet), so you can use it in other operations and commands.
- `\filldraw[...]` ...: Draws an object and fills it in at the same time, subject to commands specified in the rest of the command (or wherever the object is defined). Note that the package draws each subsequent object on top of everything before it, so if you have two objects in the same space (e.g. a circle that is shaded in with the a label in the middle) make sure you order the draw commands such that objects on top come later.
- `\begin{scope} ... \end{scope}`: Defines the 'scope' of an area for a fill command as the space where the 'brazilcircle' and 'indiacircle' objects overlap.

Unnecessary Drawing

The website <http://www.texample.net/tikz/> contains a number of more advanced tikz drawings. Check it out to see both how to make these beautiful works of art, and how to replicate them. These work especially well with your beamer presentations, where they can add much needed illustration and color to an otherwise monotonous speech. Two examples, taken from the site, are below.

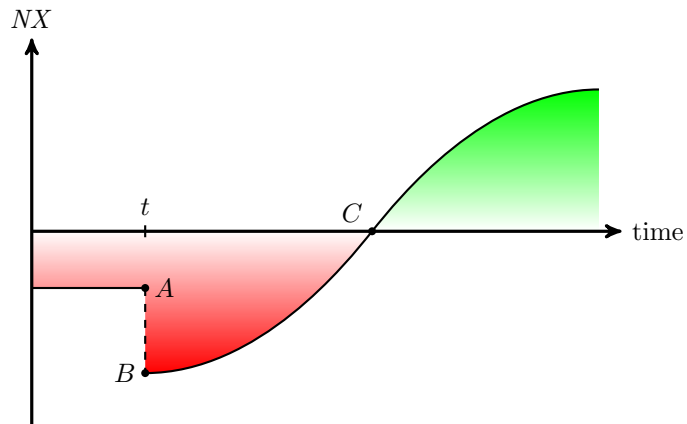


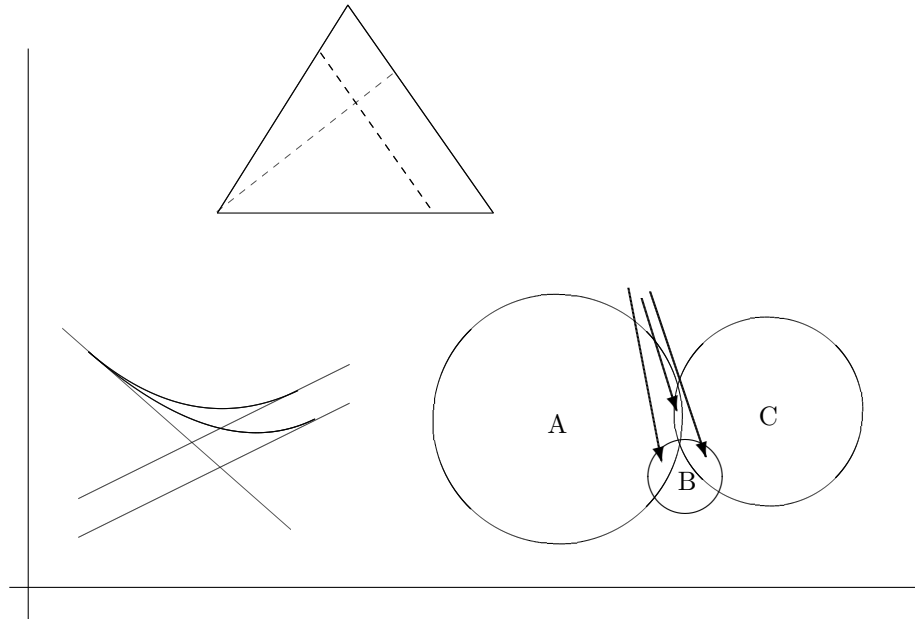
Figure 2: Trade balance following a devaluation.



Figure 3: A Sunset!

4 Quick and Dirty Freehand

Finally, if you want a diagram but don't feel like wading through the coding, a quick and dirty way to do it is using a free program TeXCAD that handles these moderately well. It works somewhat like MS Paint but much crisper, includes a few handy features for technical drawings, and it plays well with L^AT_EX. It will produce a file that you can put directly into your document using `input filename.pic` or it will give you code that will produce your picture, which you can then edit as needed. Just to get a sense of what it can do:



(un-comment the picture command to display the picture if you have the file)

5 References

The `.sgamevar` and `.egameps` packages were written by Martin Osborne, an economist at the University of Toronto, so the best resource for them is his website: <http://www.economics.utoronto.ca/osborne/latex/>. The `sgame.pdf` and `egame.pdf` guide documents are particularly useful, both for debugging and for discovering more advanced features of the packages.

The `.tikz` and `.pgf` packages were designed by Till Tantau, a computer scientist at the University of Lübeck. The best resource for learning how it works is his extensive online manual, available at <http://www.ctan.org/tex-archive/graphics/pgf/base/doc/generic/pgf/pgfmanual.pdf>. It is 560 pages long, but fortunately has a table of contents and an index and is searchable. Most of the information you might want about how to do various things is in the ‘I. Tutorials and Guidelines’ section, which is very user-friendly (and surprisingly funny). There are also examples of code for all sorts of fun stuff in `tikz` at www.texample.net.

TeXCad is available for free download at <http://texcad.sourceforge.net/>.