Drawing Games and Diagrams in LaTeX*

Sarah Bouchat

September 21, 2016

This guide emphasizes a series of ways to draw game matrices and decision trees—both of which will be useful for 835 and 836—as well as providing some basic building blocks for drawing using LaTeX. This should be considered a very rudimentary introductory guide—there are many more things to learn about drawing and figures in LaTeXthat are omitted here.

1 Strategic Form Games

Writing strategic form games using the `sgamevar` package creates diagrams that looks a lot like normal tabular environments, but it has the advantage of automatically tailoring the dimensions and labels to the alignment you want, rather than forcing you to do it by hand. There are a couple of things that distinguish it from a typical table environment:

- Use the "figure" float instead of the "table" float.
- The `\>` replaces `&` as the divider between cells. Note that in the `sgame` package, you use `&` still.
- 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	Top	3,2	4,4	2,1
	Bottom	1,1	5,7	4, 3

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 numbers. You can also use highlighting.

		2				
		Rock	Paper	Scissors	Flamethrower	Water Canon
	Rock	1,1	0,2	2,0	2,0	0,2
	Paper	2,0	1,1	0,2	0,2	2,0
1	Scissors	0,2	2,0	1,1	2,0	0,2
	Flamethrower	0,2	2,0	0,2	1,1	2,0
	Water Canon	2,0	0,2	2,0	0,2	①,①

Ammunition

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 very prone to errors and much less flexible than

*This handout borrows heavily from the department's 2010 LaTeX handout authored by Dave Ohls and 2013 materials from Richard Loeza.

		Jim		
		A	B	C
Bob	X	8,6	6,8	0,0
	Y	6,8	8,6	0,0
	Z	0,0	0,0	5,5

using the tikz approach. Be aware that using egamesps may require you to use a different compilation engine for your LaTeXdocument (e.g., XeTeX).

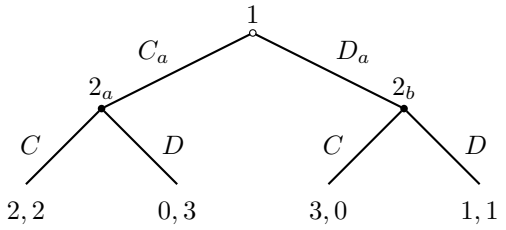
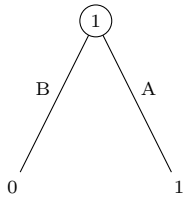


Figure 1: Sequential PD

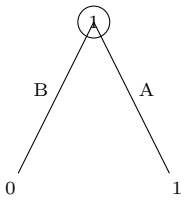
Instead of dealing with these issues, you can create extensive form games using the `tikzpicture` environment, from the tikz package. The tikz package is designed for drawing in general, and is very flexible. The package can accommodate nearly any shape or arrangement you desire. The basic syntax of 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 want 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.
- `\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.