

Truth Tables in L^AT_EX

Randy Ridenour

<2018-02-10 Sat>

Typesetting truth tables has never been easy. L^AT_EX is the gold standard for displaying logic and mathematics, but tables are awkward to edit at best. Tables are much simpler in Microsoft Word, but displaying formulas is a horrible experience.¹ Here is my current workflow.

The text that I’m using this semester is Introduction to Formal Logic with Philosophical Applications by Russell Marcus. Instead of arrows and the ampersand, it uses the horseshoe, triple bar, and dot. So, I add the following lines to my L^AT_EX preamble to simplify entering the symbols.²

```
\newcommand{\lneg}{\mathord{\sim}}
\renewcommand{\land}{\bullet}
\newcommand{\lif}{\supset}
\newcommand{\liff}{\equiv}
```

Then, I enter the truth table in either Excel or Numbers. For example, this would be a simple one line table determining the truth value of a formula for a given valuation:

P	Q	R	P	\vee if	\neg Q	\wedge	R)
1		1	1	0	0	0	1

Copy the cells that you want included in the truth table. Go to Tables Generator and select “L^AT_EX Tables” from the top menu bar. Below the top menu bar is a drop-down menu bar. Click on “File” then “Paste table data...” and paste the table data. Table Generator will generate a nicely

¹Apple’s Pages now allows users to add formulas with L^AT_EX. It’s looking like a good solution for those who like more traditional word processors.

²The AMS L^AT_EX packages already include a command called “ \vee ” for entering the vee or wedge.

formatted L^AT_EX table; be sure to uncheck “Escape special T_EX symbols” and select the option to remove all borders.

```
\begin{table}[]
\begin{tabular}{cccccccc}
\hline
P & Q & R & P & \lif & (\lneg Q & \land & R) \\
1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\
\end{tabular}
\end{table}
```

I delete the first and last line, leaving just the table data and the lines declaring the tabular environment:

```
\begin{tabular}{cccccccc}
\hline
P & Q & R & P & \lif & (\lneg Q & \land & R) \\
1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\
\end{tabular}
```

At this point, typesetting will fail because the symbols need to be in math mode. So, I’ve found two options. The first is to put all the commands for the symbols in math mode:

```
\begin{tabular}{cccccccc}
\hline
P & Q & R & P & \(\lif\) & \(\lneg\) Q & \(\land\) & R) \\
1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\
\end{tabular}
```

The second option is to change “tabular” to “array” and put the entire table into math mode:

```
\[
\begin{array}{cccccccc}
\hline
P & Q & R & P & \lif & (\lneg Q & \land & R) \\
1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\
\end{array}
\]
```

Arrays are centered on the page. If you would prefer them printed at the left margin, add “fleqn” to the document class options: `\documentclass[fleqn]{article}`. Since the array is in math mode, the letters will be italicized. I use the newtxmath font package, and it has a “frenchmath” option that sets the math font to non-italic. Other math fonts may have a similar option. Finally, whichever option is used, we need to add two lines. Adding a vertical line character to the table or array formatting options will place a vertical line between the valuation section and the rest of the truth table. Adding the booktabs package to the preamble will allow us to separate the sentence letters from the rest of the truth table with a “|” and, if desired, add a border line under the first row of the table.. This gives us the final version,

```
\[
\begin{array}{cc|cccc}
\hline
P & Q & R & P & \text{\tiny $\neg$} Q & \text{\tiny $\wedge$} R & \text{\tiny $\midrule$} \\
1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & \text{\tiny $\backslash$} \\
\end{array}
\]
```

which produces this:

P	Q	R		P	\supset	$(\sim Q \bullet R)$
1	1	1		1	0	0 0 1