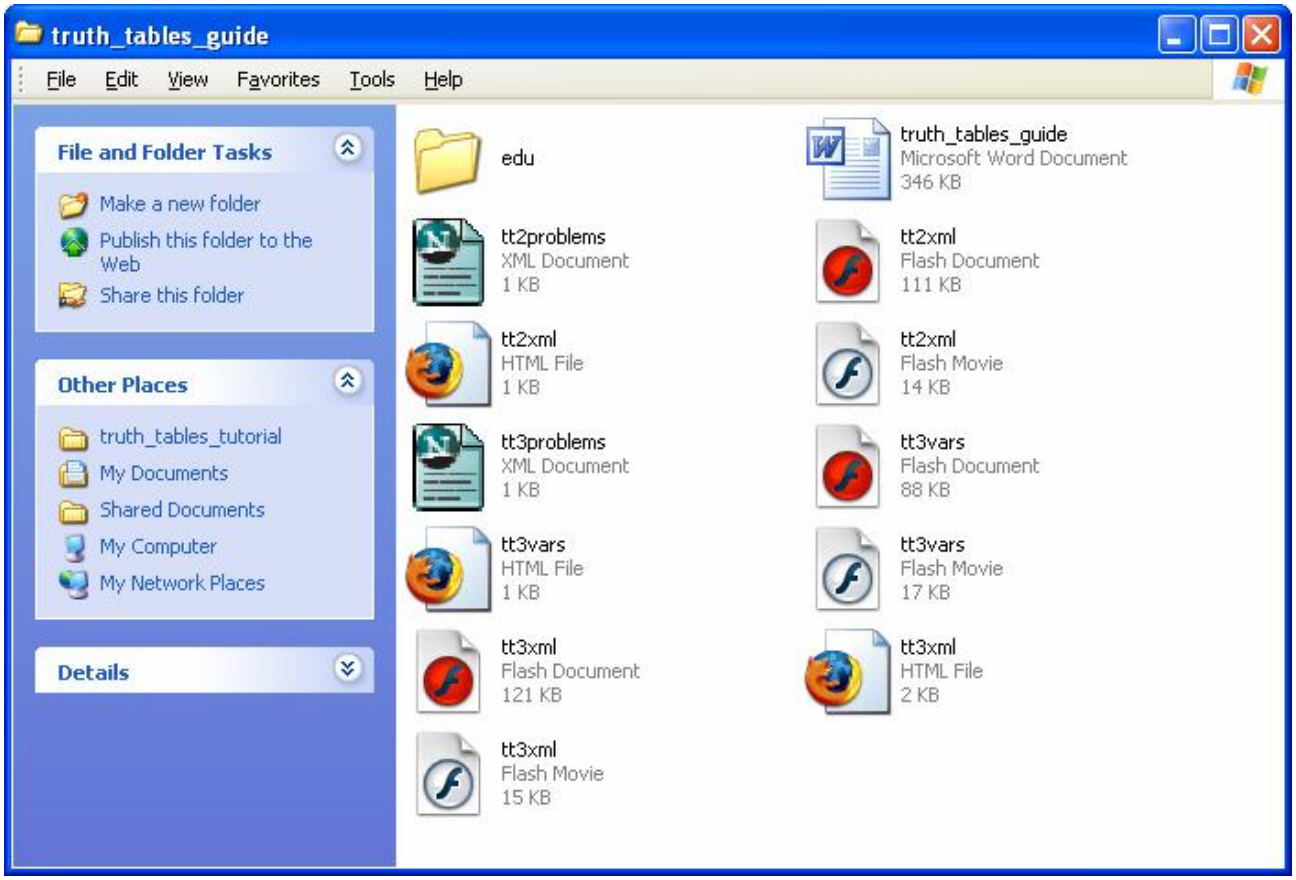
Flash Developer's Guide to Truth Tables

By Doug Ensley, Shippensburg University,
and Barbara Kaskosz, University of Rhode Island

This guide will introduce the user to two simple templates for exercises on the use of truth tables in a course in discrete mathematics or logic. Each template comes in two varieties depending on the number of propositional variables you would like to use.

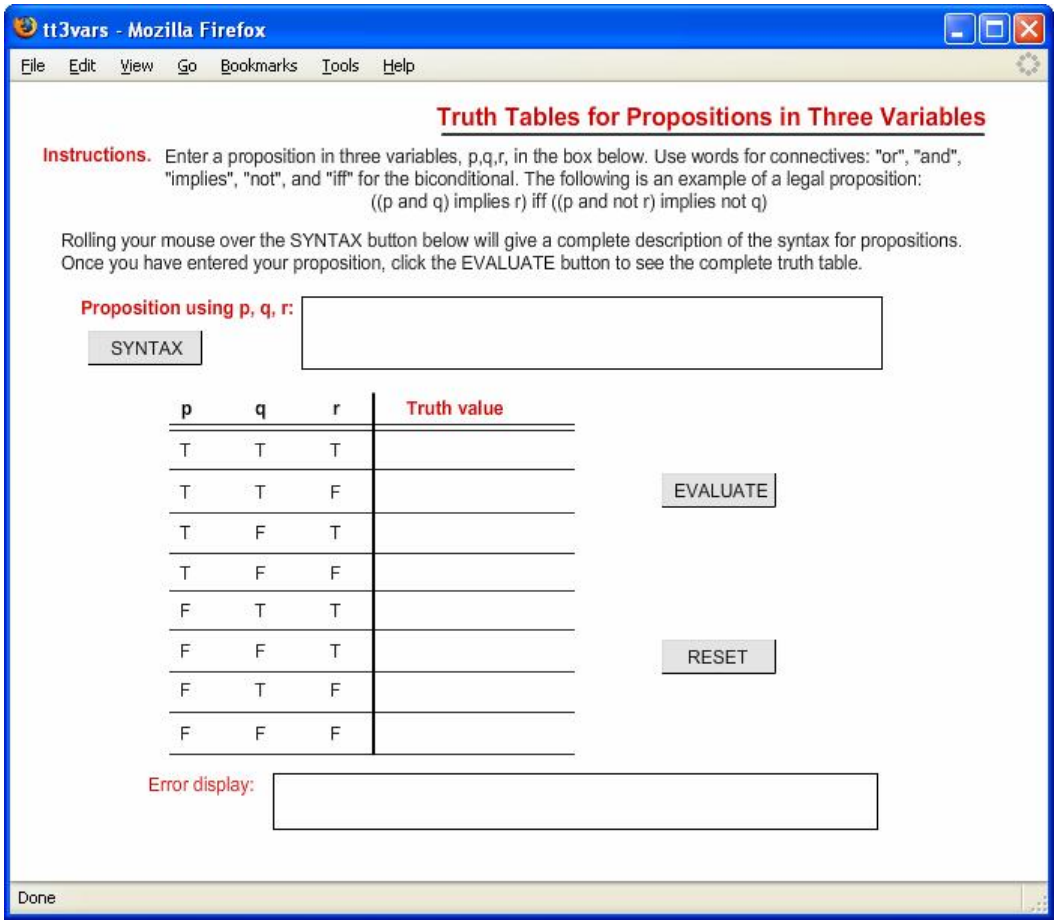
Getting started

As usual we begin by downloading a compressed folder with all of the relevant tutorial files. The file **truth_table_tutorial.zip** should be downloaded and unzipped to produce the folder **truth_table_tutorial** in an appropriate location on your computer. Once you do this, you will see the folder shown below.



The First Template

First open the file **tt3vars.html** and get a sense for the functionality of this applet. This template illustrates the basic functions of reading user input, parsing the expression, and building the appropriate truth table column. This template is meant to be extended into a pedagogically sound student activity such as the one given in the second template.

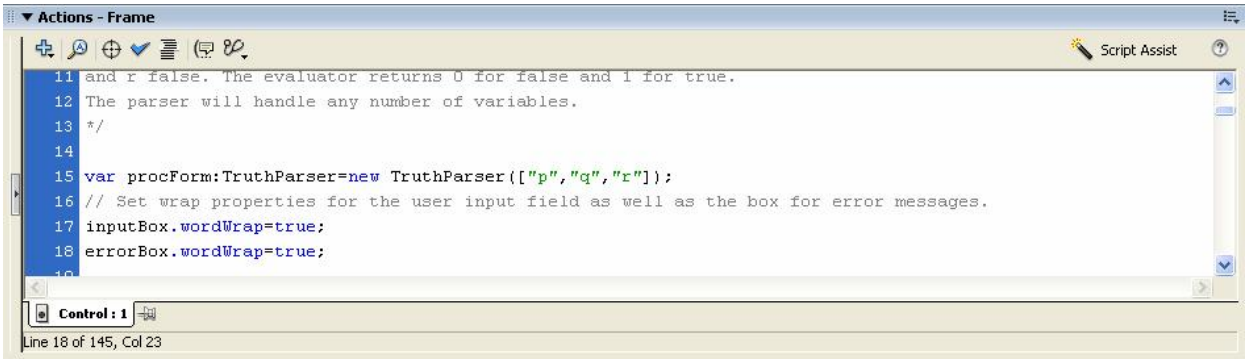


We will discuss the various components of this application by explaining how to make some changes to it.

Changing the names of the variables

If we would like to use variable names other than p, q and r, we need to make the change in (1) the instructions to the user, (2) the labels on the table, and (3) the **TruthParser** constructor. Let's change the variable names to "A", "B" and "C" to see how it can be done.

On the stage, change every occurrence of p, q and r to A, B and C, respectively. This will include the column headings. Now go to line 15 and change ["p", "q", "r"] to ["A", "B", "C"] in the call to the **TruthParser** constructor.



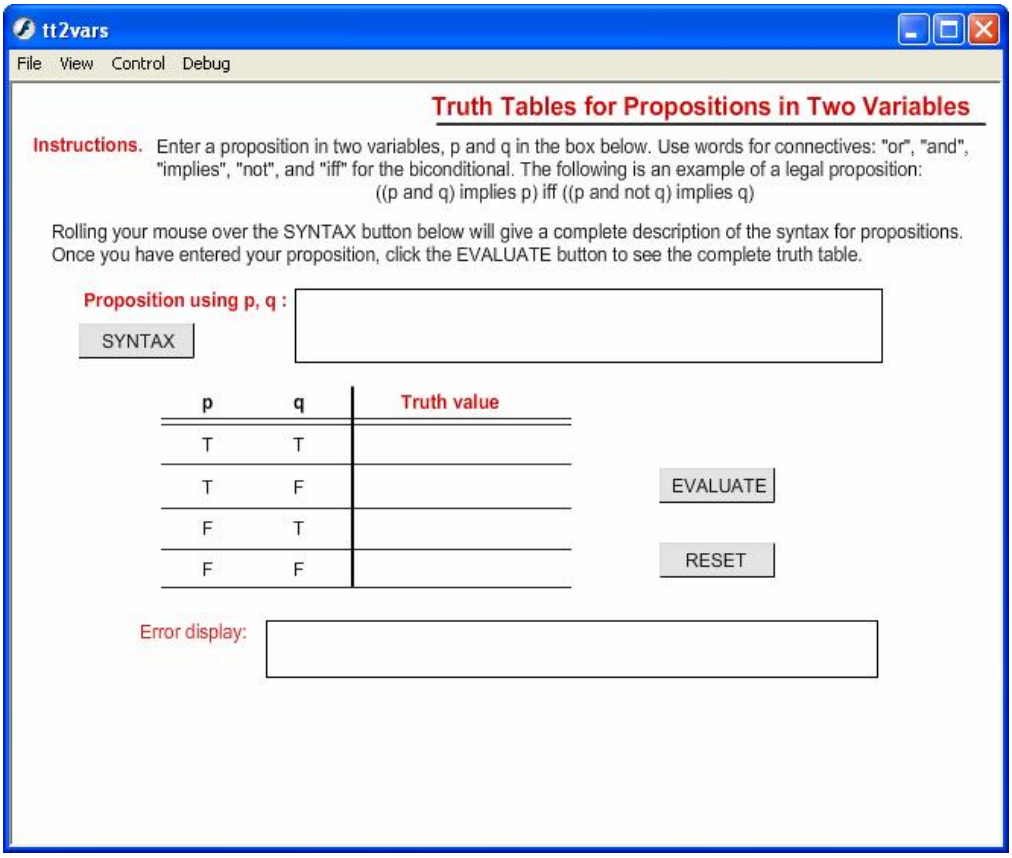
Test your movie to see that it works for expressions involving variables A, B and C.

Changing the number of variables

We can easily change the applet to use two variables instead of three by deleting some of the objects on the stage and making appropriate changes to the script as we shall describe.

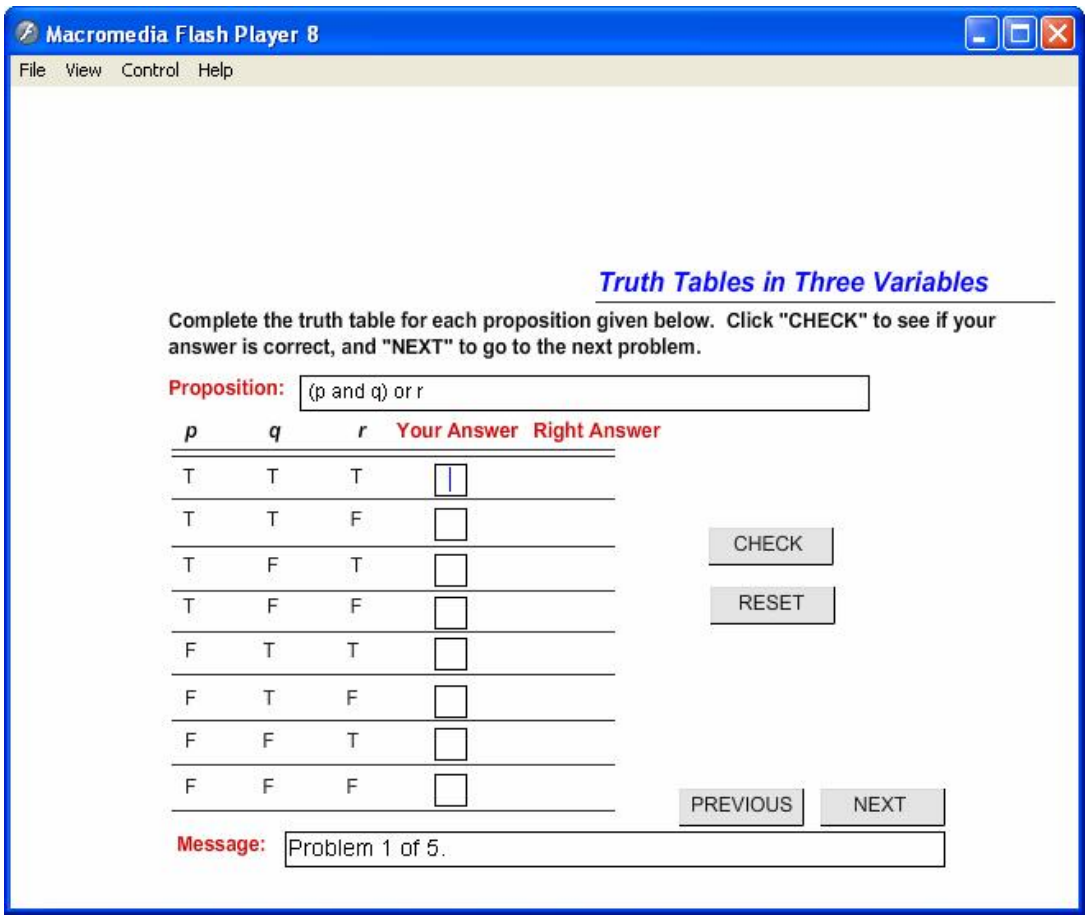
- From the File menu, choose Save As and give the name **tt2vars.fla** to the new Flash project.
- On the stage, delete completely the bottom four rows of the truth table and adjust the lines accordingly. Change the static text for the title and instructions to be consistent with the two-variable version.
- Delete the "p" column from the truth table. Use the variable names "p" and "q" as the column labels for the remaining two columns. (Given the previous discussion, you can obviously use any letters you want for the variable names. We will use "p" and "q" in this example.)
- In the **TruthParser** constructor on line 15, use "p" and "q" only as the variables within the (square) array brackets.
- Each time the output boxes are cleared (lines 28-35, lines 62-69 and lines 123-130), we should delete the lines for **outBox5**, **outBox6**, **outBox7** and **outBox8** since those boxes have been deleted from the stage.
- On lines 83-86, the values of **val1**, **val2**, **val3** and **val4** should be evaluated with inputs [1,1], [1,0], [0,1], and [0,0], respectively. Lines 53-56 (for declaring variables **val5**, **val6**, **val7**, **val8**) and lines 87-90 (for the evaluation of **val5 – val8**) should be deleted.

The result of these steps is shown in the following screen shot taken when testing this new movie.



The Second Template

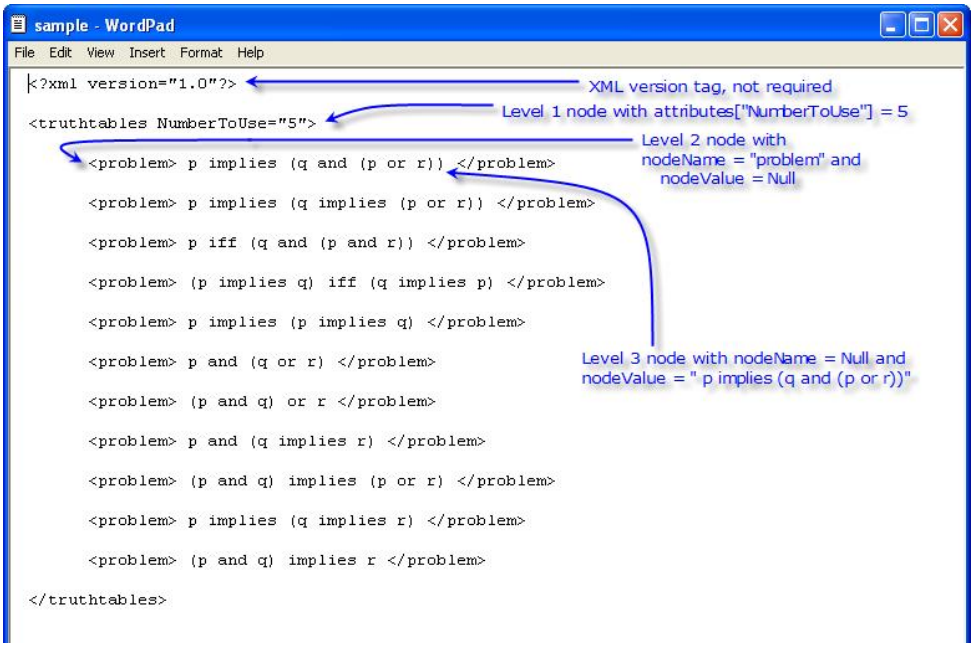
First open the file **tt3xml.html** and get a sense for the functionality of this applet. Notice that this applet is written so that the student completes the truth table and the applet checks the student's answer. The focus on this part of the tutorial will be on the way the problems (i.e., the propositional expressions) to be used are provided to the program.



Specifying the problems: Script vs. XML data file

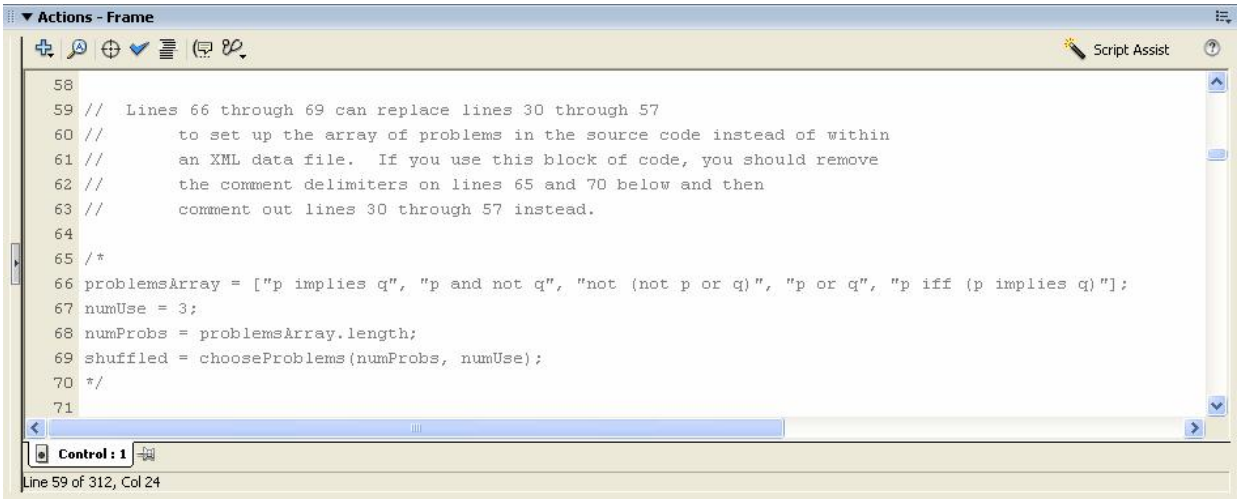
The propositional expressions to be used by this applet are stored in the variable **problemsArray**, which is an array of strings. There are two different ways to specify the expressions that will fill this array. The script in the **tt3xml.fla** file includes *both* of these methods, so we can simply "comment out" the method we do not wish to use. To comment a multiline section of code (as opposed to the // convention used to denote a single line comment), we put /* at the beginning of the section and */ at the end of the block. Lines 3-12 illustrate this commenting convention.

The default method for specifying the expressions is through an external XML data file. The file **tt3problems.xml** can be opened with any text editor (like *Notepad*). The screen shot below shows the basic structure of the file. Each expression to be used is placed between the tags `<problem>` and `</problem>`. The variable `NumberToUse` specifies the number of problems in the file that will be randomly used in the problem set delivered by the applet. (If `NumberToUse` is 0, it is assumed that all of the problems given in the XML file should be used.)



This method has the advantage that it makes the applet customizable by the instructor through a simple modification of the external XML file. In particular, the *Flash* authoring software is not necessary for changing the problems used by the applet.

The second method requires less *Actionscript* programming and is more direct. Simply define the variables **problemsArray** and **numUse** in the script. To do this in the given template file, remove the comment delimiters (the matched pair /* and */) on lines 65 and 70 and place them before/after the block of code on lines 30 through 57 instead. Using this method, you simply add strings to the definition of **problemsArray** and change **numUse** to whatever number you wish. This method requires that the *Flash* player file (swf) be rebuilt for each new set of problems, but it does nicely hide the problem list from the students.



The article [1] gives a more detailed explanation of the structure of this XML file and the *Actionscript* methods for acquiring the problem data from the file.

Reference

[1] Ensley, D. and Kaskosz, B. Using XML Data Files in Flash Programming, *MathDL Flash Forum* at http://mathdl.maa.org/mathDL/3/ (accessed July 18, 2006).