LPS30 Supplementary Notes: Decision Tree and Truth Table

Zesheng (Jason) Chen

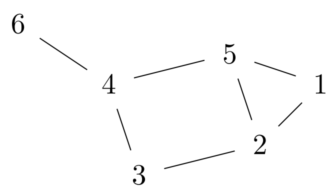
April 2019

Tree graphs are a useful tool for solving or representing a number of problems. For the sake of being precise, below I'll try to define what graph-theoretic trees are, but those definitions can be safely skipped. For the purpose of this note, just keep in mind that graph-theoretic trees are graphs that look like real trees. In this note, we see that tree graphs can help us exhaust all the True/False combinations in a truth table. The content of this note will not be tested. [1]

Definition. A *graph* is a collection of points and lines between them.

Definition. If we can start from some point A and step through lines and points to arrive at point B, then we say there is a *path* between point A and point B.

Example.



The above is a graph with a path, for instance, between 1 and 6 (start from 1, go through 5,4,6). We write this path as $\langle 1, 5, 4, 6\rangle$.

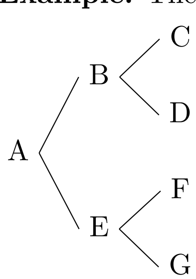
Definition. If a point in a graph has a path to itself, then we say the graph has a *cycle*.

For example, the above graph has at least one cycle: $\langle 1, 2, 5, 1\rangle$.

Definition. A (graph-theoretic) *tree* is a graph with points and lines between the points, which satisfies the following two conditions:
1. it doesn't have any cycles
2. every two distinct points are connected by exactly one path

Definition. An *end point* on a tree is a point to and from which there is only one line (i.e., it does not branch off further)

Example. The following is a very simple tree, with end points C, D, F, G:



[1] Many thanks to Adam Chin for helpful suggestions

where the topmost path can be written as $\langle A, B, C\rangle$.

Now, on to the important stuff.

In particular, we are interested in problems of the following form.

Problem. How do we fill in the question marks below?

P	Q	P & Q
?	?	
?	?	
?	?	
?	?	

Now this truth table has two distinct sentence letters, and we want to write out all the possibilities. One (tedious and not-so-reliable) way is to just dive in and write all that you can think of: True True, True False, False True, False False. This might be easy for two (distinct) sentence letters, but the number of question marks grows really fast when you have more letters. For example:

Problem. How do we fill in the question marks below?

P	Q	R	S	((P & Q) & R) & S
?	?	?	?	
?	?	?	?	
?	?	?	?	
?	?	?	?	
?	?	?	?	
?	?	?	?	
?	?	?	?	
?	?	?	?	
?	?	?	?	
?	?	?	?	
?	?	?	?	
?	?	?	?	
?	?	?	?	
?	?	?	?	
?	?	?	?	
?	?	?	?	

As you can see, this is a very long table, with lots of possibilities! (Plus, here the total number of possibilities is given. Imagine if you have to figure out how many possibilities there are to begin with!) Understandably, "writing out all the possibilities" is a difficult method for this, because you might accidentally slip in a few repeats, or you might not exhaust all the possibilities. What do we do then?

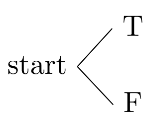
To see how trees can help, first we need to think about what we do when we fill in these question marks. Let's consider a 3-letter case:

P	Q	R	(P & Q) & R
?	?	?	
?	?	?	
?	?	?	
?	?	?	
?	?	?	
?	?	?	
?	?	?	
?	?	?	

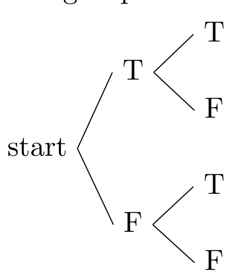
Suppose I'm at the red question mark. What do I put there? Well, there are really just two options: I could put T, or I could put F. Suppose I put a T there, so we'll get the following:

P	Q	R	(P & Q) & R
T	?	?	
?	?	?	
?	?	?	
?	?	?	
?	?	?	
?	?	?	
?	?	?	
?	?	?	

Notice how, in filling in the first question mark, I'm essentially making a decision. That is, there are two possibilities of how things might have gone for the first question mark. We capture these two possibilities with a tree (which we call a decision tree). [2]

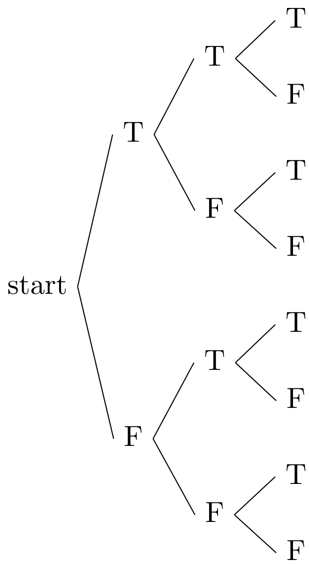


Now, for each possibility I make for the first letter, I will have to decide what I need to put for the second letter. Again, for the second letter I can put a T, or I can put an F. We capture this branching of possibilities again with a larger tree:



[2]Helpful Analogy: Bendersnatch, or Detroit: Become Human, or any game/movie/visual novel where you have to make decisions and your decisions will impact the later development.

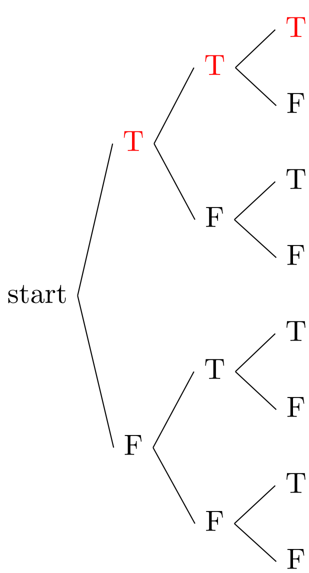
At this point, I've mapped out all the decisions for the first two letters. Now we need to decide what to do for the third. You've probably guessed what that tree might look like:



The above big tree represents all the possible moves and decisions I can make when filling out the question marks. Now, the key claim is the following:

Each path from "start" to an endpoint will be an answer to a row of question marks.

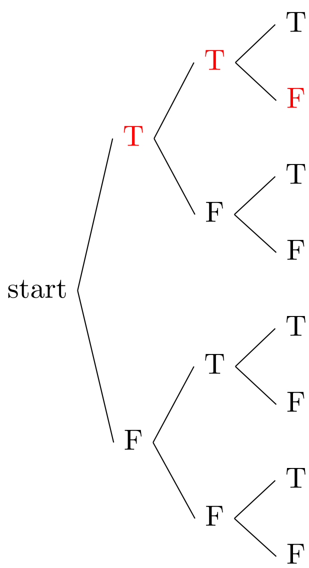
Let's put that idea to use. First look at the topmost path



This path is $\langle \text{start}, T, T, T \rangle$. This will be how we fill in the first row of question marks:

P	Q	R	(P & Q) & R
T	T	T	
?	?	?	
?	?	?	
?	?	?	
?	?	?	
?	?	?	
?	?	?	
?	?	?	

Moving on, we hit the second path in the tree



This path is $\langle \text{start}, T, T, F \rangle$. This will be how we fill in the second row.

P	Q	R	(P & Q) & R
T	T	T	
T	T	F	
?	?	?	
?	?	?	
?	?	?	
?	?	?	
?	?	?	
?	?	?	

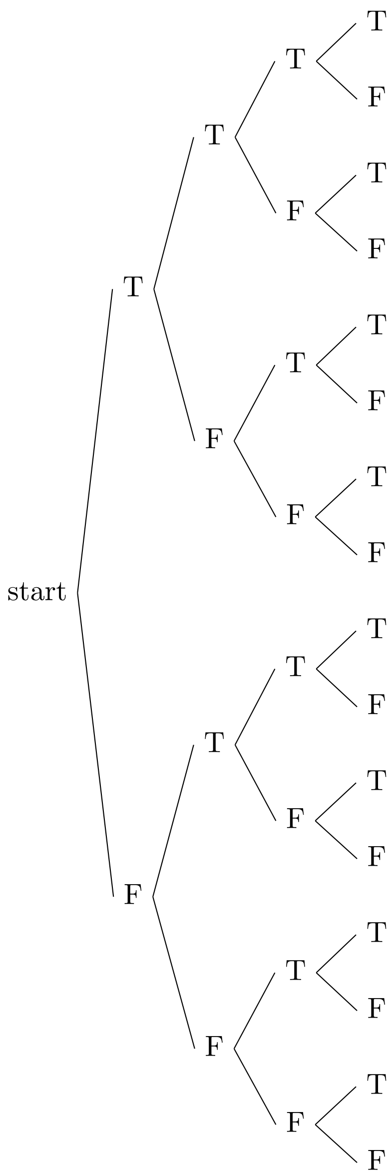
Hopefully, at this point you get the idea (exercise: finish the rest of the truth table using the decision tree).

We see that this works for 4 distinct letters as well (and indeed, for any arbitrary number of distinct sentence letters). Recall one of our earlier problems:

Problem. How do we fill in the question marks below?

P	Q	R	S	((P & Q) & R) & S
?	?	?	?	
?	?	?	?	
?	?	?	?	
?	?	?	?	
?	?	?	?	
?	?	?	?	
?	?	?	?	
?	?	?	?	
?	?	?	?	
?	?	?	?	
?	?	?	?	
?	?	?	?	
?	?	?	?	
?	?	?	?	
?	?	?	?	
?	?	?	?	

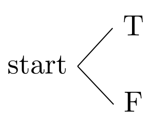
But now we know how to solve this! Our decisions for what to put in the first, second, third, and fourth sentence letters are captured by the following tree:


start
T
F
T
F
T
F
T
F
T
F
T
F
T
F
T
F
T
F
T
F
T
F
T
F
T
F
T
F
T
F

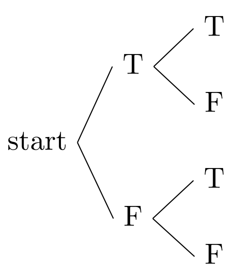
Each path indeed gives us an answer to one row. As we can see here:

P	Q	R	S	((P & Q) & S) & R
T	T	T	T	
T	T	T	F	
T	T	F	T	
T	T	F	F	
T	F	T	T	
T	F	T	F	
T	F	F	T	
T	F	F	F	
F	T	T	T	
F	T	T	F	
F	T	F	T	
F	T	F	F	
F	F	T	T	
F	F	T	F	
F	F	F	T	
F	F	F	F	

In fact, the decision trees tell us even more. Think back to when we try to make a decision for the first letter:



Here we have 2 possibilities. If I need to make decisions for a second letter, each of the two possibilities will branch off into two possibilities (that is, the number of paths in the new tree is the number of paths in the old tree times 2).



So here we have $2 \times 2 = 4$ paths. That is, a truth table with 2 distinct sentence letters will have $2 \times 2 = 4$ rows.

Now, whenever I try to make decisions for a new letter, I am branching off the tree at each of the end points. So if a tree has y end points, branching it off at each of the end points will result in a new tree with $2y$ end points. This means that adding a new sentence letter to a truth table results in the number of rows being multiplied by 2. This gives us the following fact:

Fact: a truth table with n distinct sentence letters will have 2^n rows.