First Degree Entailment

Melvin Fitting

March 30, 2018

1 Truth Tables

First Degree Entailment is a four-valued logic whose importance has become ever more apparent over the years. It arose out of work on relevance logics, but we don't need to go into that now. In a famous paper, [1], Nuel Belnap gave it a plausible intuitive interpretation and presented interesting technical results concerning it. It is best represented by a lattice structure due, I understand, to Michael Dunn, and we will see this later. It also turns out to be the simplest non-trivial example of a *bilattice*, and we might discuss these later in the course.

In classical logic every formula has exactly one truth value, either $\mathbf{t}$ or $\mathbf{f}$. But now, suppose we are supplied with information about the real world, and that information can be incomplete or contradictory. A natural way of representing this is to allow a proposition to have a *set* of truth values: only $\mathbf{t}$, only $\mathbf{f}$, both, or neither. Think of such a set as representing what we've been told by various people. We'll use the following notation for these four cases.

$$\begin{array}{ccc} \bot & \text{for} & \emptyset \\ \textit{false} & \text{for} & \{\mathbf{f}\} \\ \textit{true} & \text{for} & \{\mathbf{t}\} \\ \top & \text{for} & \{\mathbf{f}, \mathbf{t}\} \end{array}$$

We'd like to think of these as truth values in a four-valued logic, so what should the truth tables be? Let's look at conjunction first. In classical logic we have the following two conditions.

$$\begin{array}{lll} X \wedge Y \text{ has truth value } \mathbf{t} & \text{if and only if} & X \text{ has value } \mathbf{t} \text{ and } Y \text{ has value } \mathbf{t} \\ X \wedge Y \text{ has truth value } \mathbf{f} & \text{if and only if} & X \text{ has value } \mathbf{f} \text{ or } Y \text{ has value } \mathbf{f} \end{array} \tag{1}$$

Exercise 1 *Assume the classical logic condition that every formula has exactly one of* $\mathbf{t}$ *or* $\mathbf{f}$ *as its truth value, and show that then each of the two conditions in (1) implies the other one.*

We are not working in classical logic because we've dropped the assumption that every formula has exactly one of $\mathbf{t}$ or $\mathbf{f}$ as its truth value. We are now allowing neither or both as values, but we can still use the conditions from (1). Suppose, as an example, that X has the value *true*, that is, $\{\mathbf{t}\}$ and Y has the value $\top$, that is, $\{\mathbf{f}, \mathbf{t}\}$. What value should we assign to $X \wedge Y$? Well, both X and Y have the value $\mathbf{t}$ so the conjunction should also have value $\mathbf{t}$. But also, one of X and Y has the value $\mathbf{f}$, namely Y, so the conjunction should have the value $\mathbf{f}$. So we should set the truth value of $X \wedge Y$ to be $\{\mathbf{f}, \mathbf{t}\}$, or $\top$. More briefly, $\textit{true} \wedge \top$ should be $\top$.

In this way we can fill in the entire conjunction table, as follows.

$\wedge$	*true*	$\bot$	$\top$	*false*
true	*true*	$\bot$	$\top$	*false*
$\bot$	$\bot$	$\bot$	*false*	*false*
$\top$	$\top$	*false*	$\top$	*false*
false	*false*	*false*	*false*	*false*

Exercise 2 *Verify the entries in the conjunction table (or at least some of them).*

Next we can do a similar thing with disjunction. From classical logic we have the following conditions.

$$\begin{array}{lll} X \vee Y \text{ has truth value } \mathbf{t} & \text{if and only if} & X \text{ has value } \mathbf{t} \text{ or } Y \text{ has value } \mathbf{t} \\ X \vee Y \text{ has truth value } \mathbf{f} & \text{if and only if} & X \text{ has value } \mathbf{f} \text{ and } Y \text{ has value } \mathbf{f} \end{array} \tag{2}$$

We can use the conditions in (2) to assign values to disjunctions. For instance, consider $\top \vee \bot$. Writing these out as sets of classical truth values we have $\{\mathbf{f}, \mathbf{t}\} \vee \emptyset$. One of the disjuncts contains $\mathbf{t}$ so the disjunction should have $\mathbf{t}$ as a value. But it is not the case that both disjuncts have $\mathbf{f}$, so the disjunction should not have $\mathbf{f}$ as value. Thus the disjunction should have value $\{\mathbf{t}\}$, or *true*. In this way we get the following table.

$\vee$	*true*	$\bot$	$\top$	*false*
true	*true*	*true*	*true*	*true*
$\bot$	*true*	$\bot$	*true*	$\bot$
$\top$	*true*	*true*	$\top$	$\top$
false	*true*	$\bot$	$\top$	*false*

Exercise 3 *Verify some of the entries in the disjunction table.*

Negation is easy. Classically it switches around $\mathbf{t}$ and $\mathbf{f}$. Then it should turn *false* $= \{\mathbf{f}\}$ into $\{\mathbf{t}\} =$ *true*, $\top = \{\mathbf{f}, \mathbf{t}\}$ into $\{\mathbf{t}, \mathbf{f}\} = \top$, and so on. We get the following.

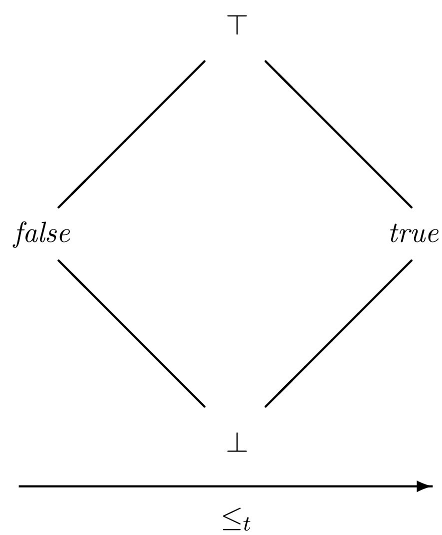
$\neg$	
true	*false*
$\bot$	$\bot$
$\top$	$\top$
false	*true*

Exercise 4 *Show that $X \vee Y$ and $\neg(\neg X \wedge \neg Y)$ have the same value, no matter which of false, true, $\bot$, or $\top$ are assigned to X and to Y. You can do this by checking all the cases, but it is easier to make use of the underlying principles in (1) and (2) themselves.*

One could define an implication, either $\neg X \vee Y$ or $\neg(X \wedge \neg Y)$ would do, but the connective turns out to be of no particular interest and we will not use it.

2 Hasse Diagrams

There is another way the truth tables for our four-valued connectives could be created, using a lattice representation called a *Hasse diagram*. We connect the four truth values in a way that represents 'degree of truth'. Usually this is done so that going upwards represents an increase (in a sense that we will explain shortly). But we will display a diagram in which this is represented by a move to the right (something that we will also explain shortly). Here is our so-called truth ordering.



The idea is, any move to the right either decreases falsehood or increases truth. For instance, moving right from *false* $= \{\mathbf{f}\}$ to $\bot = \emptyset$ eliminates $\mathbf{f}$, so falsehood decreases. moving from *false* $= \{\mathbf{f}\}$ to $\top = \{\mathbf{f}, \mathbf{t}\}$ adds $\mathbf{t}$, so truth goes up. A move from $\bot = \emptyset$ to *true* $= \{\mathbf{t}\}$ increases truth because it adds $\mathbf{t}$. A move from $\top = \{\mathbf{f}, \mathbf{t}\}$ to *true* $= \{\mathbf{t}\}$ removes $\mathbf{f}$ thus decreasing falsehood. There are no direct connections between $\bot$ and $\top$ because a move between them does not represent just a decrease in falsehood or an increase in truth. This is all quite informal, but it plausible motivation.

For any of the four truth values, call them a and b, we write $a \leq_t b$ if a is to the left of b in the diagram above. We use this ordering in a weak sense; that is, we also assume $a \leq_t a$ for each of the four values. Still, it is awkward language to keep saying "a is less true, or more false, or the same as b". We will just say "a is less true than b" as shorthand, even though it is not really accurate. We will also say "a is to the left of b", by which we mean it is to the left in the diagram, or is the same thing as.

Technically, the diagram showing the $\leq_t$ ordering is a *complete lattice*. We don't need to have a general definition of what this means, but this will help you connect with other accounts in the literature. It is the case that, in any complete lattice, there is a natural way of defining a *meet* operation, also called a *greatest lower bound*. In our case, for any a and b, the meet of a and b is the rightmost thing that is to the left of both a and b. In other words, the meet of a and b should be less true than either a or b, but it should be the truest such thing. Think about it. It has the characteristics we expect of conjunction: $a \wedge b$ should be the truest thing that is simultaneously less true than both a and b. Here are some examples.

Example 2.1 In the diagram showing the $\leq_t$ ordering, we have the following.

1. Consider *false* and *true*. The only thing (weakly) left of both in the diagram is *false*, so this must be the largest thing left of both. The meet of *false* and *true* is *false*.

2. Consider $\top$ and *true*. The items (weakly) left of both are *false* and $\top$. The rightmost of these is $\top$. The meet of $\top$ and *true* is $\top$.

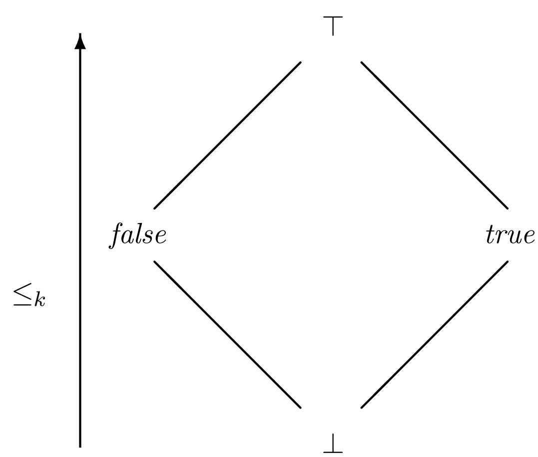
3. Consider $\top$ and $\bot$. Only *false* is left of both, so the meet of $\top$ and $\bot$ is *false*.

Notice that the values the lattice gives us for meet match those in the earlier table for $\wedge$. In fact, this is a second way of characterizing the operation $\wedge$.

There is an operation dual to meet, called *join*, also called *least upper bound*. For any a and b, the join of a and b is the leftmost thing that is to the right of both a and b. We have just switched the roles of left and right in the definition of meet. As a simple example, only *true* is right of both $\top$ and $\bot$, so it is the leftmost thing right of both. The join of $\top$ and $\bot$ is *true*. More generally, join agrees with the $\vee$ table we gave earlier.

3 Another Hasse Diagram

Belnap gave two Hasse diagrams in his paper, not one. We have been reading the diagram using a left-right ordering. We could also use an up–down ordering.



There is a good intuition here, just as there was in the left-right version. A move upward increases the information we have (the k subscript is meant to suggest 'knowledge'). At the bottom, $\bot$, we have no information about truth or falsity. A move upwards to *false* or *true* gives us one of **f** or **t**, and a move from there upwards to $\top$ gives us both. The $\leq_k$ ordering is one that reflects the degree of information we have.

Just as we did with $\leq_t$, we can define meet and join operations using the $\leq_k$ ordering. Now the meet of a and b is the highest thing (weakly) below both a and b. We write $a \otimes b$ for this meet; it is called *consensus*. For instance, *false* and *true* are both above only $\bot$, so that is the highest thing below both. Then *false* $\otimes$ *true* $= \bot$. In fact, if we think of the four truth values as sets of classical truth values, the operation $\otimes$ is simply intersection. Here is a table for it.

$\otimes$	*true*	$\bot$	$\top$	*false*
true	*true*	$\bot$	*true*	$\bot$
$\bot$	$\bot$	$\bot$	$\bot$	$\bot$
$\top$	*true*	$\bot$	$\top$	*false*
false	$\bot$	$\bot$	*false*	*false*

There is also a join operation for the $\leq_k$ ordering, denoted $\oplus$. It is called *gullability*. In this case it coincides with set union. Here is its table.

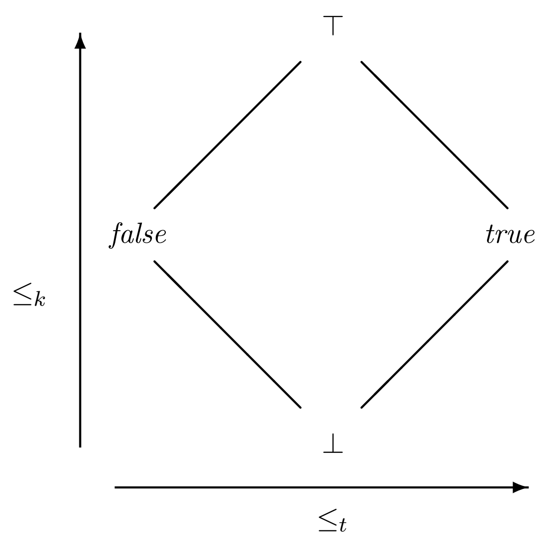
$\oplus$	*true*	$\bot$	$\top$	*false*
true	*true*	*true*	$\top$	$\top$
$\bot$	*true*	$\bot$	$\top$	*false*
$\top$	$\top$	$\top$	$\top$	$\top$
false	$\top$	*false*	$\top$	*false*

Finally, just as the negation operation $\neg$ reversed things left-right, there is an operation $-$ that reverses things up-down. It is called *conflation*. Here is its table.

$-$	
true	*true*
$\bot$	$\top$
$\top$	$\bot$
false	*false*

The operations coming from the $\leq_t$ ordering are considered *logical*, and make up what Belnap called "a useful four-valued logic." The operations coming from the $\leq_k$ ordering are not logical, but have to do with information. One place where they come up in a natural way is in Kripke's Theory of Truth, but that is too far afield to go into here. From now on, the logic we consider uses $\wedge$, $\vee$ and $\neg$ only, though connections between all the operations are interesting and important.

As we said, Dunn and Belnap gave two different orderings. Today they are often combined into a single *double Hasse diagram*, as follows.



We have four binary operations on $\{false, \bot, \top, true\}$: $\wedge$, $\vee$, $\otimes$, and $\oplus$. Then we can consider *distributive laws*. The following hold simply because that's the way lattices work, and each of the four items below involves operations from the same lattice ordering.

1. $a \wedge (b \vee c) = (a \wedge b) \vee (a \wedge c)$
2. $a \vee (b \wedge c) = (a \vee b) \wedge (a \vee c)$

3. $a \otimes (b \oplus c) = (a \otimes b) \oplus (a \otimes c)$

4. $a \oplus (b \otimes c) = (a \oplus b) \otimes (a \oplus c)$

Rather remarkably, distributive laws hold even if we mix operations defined using both the truth and the information orderings. The following are true.

5. $a \wedge (b \otimes c) = (a \wedge b) \otimes (a \wedge c)$

6. $a \wedge (b \oplus c) = (a \wedge b) \oplus (a \wedge c)$

7. $a \vee (b \otimes c) = (a \vee b) \otimes (a \vee c)$

8. $a \vee (b \oplus c) = (a \vee b) \oplus (a \vee c)$

9. $a \otimes (b \wedge c) = (a \otimes b) \wedge (a \otimes c)$

10. $a \otimes (b \vee c) = (a \otimes b) \vee (a \otimes c)$

11. $a \oplus (b \wedge c) = (a \oplus b) \wedge (a \oplus c)$

12. $a \oplus (b \vee c) = (a \oplus b) \vee (a \oplus c)$

Exercise 5 *Pick one of the 'mixed' distributive laws above, and verify it.*

4 The Logic FDE

As a logic, we are only interested in $\wedge$, $\vee$, and $\neg$. Ignore the $\leq_k$ based operations. We have four truth values, so to get a many-valued logic we must say which of them are *designated*. These are $\{true, \top\}$, the two values containing $\mathbf{t}$. Informally, these represent 'at least true'. We are primariuly interested in consequence, defined as follows. If S is a set of formulas and X is a single formula, $S \vdash_{FDE} X$ provided that every valuation that gives the members of S a designated truth value also gives X a designated truth value.

One can establish that a particular consequence does, or does not, hold by constructing a truth table. These can be extraordinarily long. There are a couple of tableau systems for FDE, and we will use the one described in Graham Priest's book. The description is not repeated here. However, here is a small sample of what a truth table approach is like.

Example 4.1 $\{\neg(P \vee Q), \neg R\} \vdash_{FDE} \neg(P \vee R)$ holds. You are invited to try constructing a truth table to show this. It will have 64 lines!

Example 4.2 $\{P, \neg P \vee Q\} \vdash_{FDE} Q$ does not hold. This is one of the reasons that defining $P \supset Q$ to be $\neg P \vee Q$ is not desirable—we don't have *modus ponens*. Here is a full truth table. Notice that on line 10, P is $\top$ and $\neg P \vee Q$ is *true*, both designated values, but Q is $\bot$ which is not designated.

Line 12 is another counter-example.

	P	Q	$\neg P$	$\neg P \vee Q$
1	*true*	*true*	*false*	*true*
2	*true*	$\bot$	*false*	$\bot$
3	*true*	$\top$	*false*	$\top$
4	*true*	*false*	*false*	*false*
5	$\bot$	*true*	$\bot$	*true*
6	$\bot$	$\bot$	$\bot$	$\bot$
7	$\bot$	$\top$	$\bot$	*true*
8	$\bot$	*false*	$\bot$	$\bot$
9	$\top$	*true*	$\top$	*true*
10	$\top$	$\bot$	$\top$	*true*
11	$\top$	$\top$	$\top$	$\top$
12	$\top$	*false*	$\top$	$\top$
13	*false*	*true*	*true*	*true*
14	*false*	$\bot$	*true*	*true*
15	*false*	$\top$	*true*	*true*
16	*false*	*false*	*true*	*true*

References

[1] Nuel D. Belnap Jr. “A useful four-valued logic”. In: *Modern Uses of Multiple-Valued Logic*. Ed. by Jon Michael Dunn and George Epstein. D. Reidel, 1977.