Notes on Destructive Intuitionistic Tableaus

Melvin Fitting

February 15, 2018

In class I sketched completeness and soundness for Intuitionistic tableaus. I thought I had a treatment of this already published that I could post as notes, but it turns out that all my published discussions make use of a different kind of completeness argument, and I don't want to confuse things by bringing it in here. So I'm writing up what was done in class. Please let me know about any typos or other problems you may find.

1 The General Idea

Proof systems and semantics are connected by *soundness* and *completeness* results. Soundness says that any formula having a proof in the proof system will be valid in the semantics. Completeness says that anything valid in the semantics will have a proof in the proof system. The two together say that the proof system proves exactly the formulas valid in the semantics. Putting this another way, they say that any formula will either have a proof or a counter-model.

Here we sketch soundness and completeness results connecting destructive Intuitionistic tableaus and the semantics given by Kripke Intuitionistic models. These were defined in class, and this is not repeated here.

2 Soundness

All tableau soundness results follow the same pattern, no matter what the logic. They always begin by defining what it means for a tableau to be satisfiable. Loosely, a tableau is satisfiable if the formulas on at least one branch 'could happen'. Here is a more precise definition, for destructive Intuitionistic tableaus and Kripke models.

Definition 2.1 (Satisfiable) This is a 'top down' characterization.

1. A tableau is satisfiable if at least one of its branches is.
2. A tableau branch is satisfiable if the set of signed formulas on it is.
3. A set of signed formulas S is satisfiable if there is some Kripke intuitionistic model and some state Γ in it such that:
   (a) if $T\,X \in S$ then $\Gamma \Vdash X$;
   (b) if $F\,X \in S$ then $\Gamma \not\Vdash X$.

Now there are two basic, and simple results that we need.

Proposition 2.2 *No tableau can be both closed and satisfiable.*

Proof Suppose some tableau is both closed and satisfiable. Then some branch is satisfiable, so if S is the set of formulas on that branch, there is some state Γ in some Kripke model at which the T signed formulas in S hold and the F signed formulas fail. But the tableau is closed, so S must contain both $T\,A$ and $F\,A$ for some A, and thus A must both hold and fail at Γ, and this is impossible. ∎

Proposition 2.3 *If a tableau is satisfiable and any tableau rule is applied, the resulting tableau is also satisfiable.*

Proof Suppose we have a satisfiable Intuitionistic tableau. Then some branch, say $\mathcal{T}$, is satisfiable, say at state Γ of some Kripke intuitionistic model. Now apply a rule to the tableau. If we apply it to a formula on some branch other than $\mathcal{T}$, then $\mathcal{T}$ is still in the resulting tableau, so we still have a satisfiable tableau. This is the trivial case.

Now suppose a tableau rule is applied to a formula on branch $\mathcal{T}$ itself. We have many cases, one for each rule. We only do two cases, $T\vee$ and $F\supset$. These should give the idea.

Case $T\vee$ Suppose $\mathcal{T}$ looks like this.

$$\begin{array}{c} \vdots \\ T\,X \vee Y \\ \vdots \end{array}$$

Also suppose $\mathcal{T}$ is satisfied at state Γ, so in particular, $\Gamma \Vdash X \vee Y$. Now we apply the tableau rule, turning the branch into the following.

$$\begin{array}{c} \vdots \\ \boldsymbol{T\,X \vee Y} \\ \vdots \\ \diagup \quad \diagdown \\ \boldsymbol{T\,X} \qquad \boldsymbol{T\,Y} \end{array}$$

But since $\Gamma \Vdash X \vee Y$, either $\Gamma \Vdash X$ or $\Gamma \Vdash Y$. If it's the first, the left branch is satisfiable at Γ, and if it's the second the right branch is. Either way, at least one branch is satisfiable.

Case $F\supset$ Now suppose $\mathcal{T}$ looks like this.

$$\begin{array}{c} S \\ F\,X \supset Y \end{array}$$

For this case it matters what the other signed formulas are, so we have used S to stand for the set of them instead of using dots. Let's say the branch is satisfied at Γ, so $\Gamma \not\Vdash X \supset Y$, but all the signed formulas in S behave appropriately at Γ. On applying the rule the branch gets replaced, as indicated in the following.

$$\begin{array}{c} S \\ F\,X \supset Y \\ \hline S_T \\ T\,X \\ F\,Y \end{array}$$

> Since $\Gamma \not\Vdash X \supset Y$ there must be some state Δ with $\Gamma\mathcal{R}\Delta$ so that $\Delta \Vdash X$ but $\Delta \not\Vdash Y$. Also whatever holds at a state will also hold at any accessible state, so since Γ satisfies S, any accessible state will satisfy S_T. Then the new branch is still satisfied, at Δ instead of the original state Γ.

■

Now we can show soundness easily. The proof is by contradiction. Suppose X has a proof, but is not valid. Since it has a proof, there must be a closed tableau $\mathcal{T}$ that starts with $F\,X$. The construction of $\mathcal{T}$ starts with the trivial tableau with just one branch and one signed formula.

$$F\,X$$

Since X is not valid, there must be some state Γ of some model at which X fails; that is, $\Gamma \not\Vdash X$. But then, at Γ, the set $\{F\,X\}$ will be satisfied, hence the tableau proof of X begins with a satisfiable tableau. Then every tableau we get by further rule applications must be satisfiable. In particular, $\mathcal{T}$ must be satisfiable. But $\mathcal{T}$ is closed. This is impossible.

3 Completeness

I'll give a discussion accompanied by an example. The example was done in class, but I'm doing it more carefully now. The formula we use is $\neg(P \wedge Q) \supset (\neg P \vee \neg Q)$, which is not provable.

To construct a proof we need to create a tableau in which every branch closes. To construct a counter-model we need to find a branch we cannot close. But there are places where we may have real choices about what to do next on a branch. These involve applications of the $F\supset$ and the $F\neg$ rule, both of which cause F signed formulas to be deleted. I'll call these rules *destructive rules.* It is destructive rule applications that give us alternate states in our counter-model. In constructing a proof, I've been crossing out F signed formulas when destructive rules come up. Instead here I'll draw a line and start a new 'stage' of the construction, copying T signed formulas over, instead of deleting F signed formulas.

A tableau construction for our formula begins with the following, which we label $\boxed{1}$, to indicate stage 1.

$$\boxed{1} \qquad F\,\neg(P \wedge Q) \supset (\neg P \vee \neg Q)$$

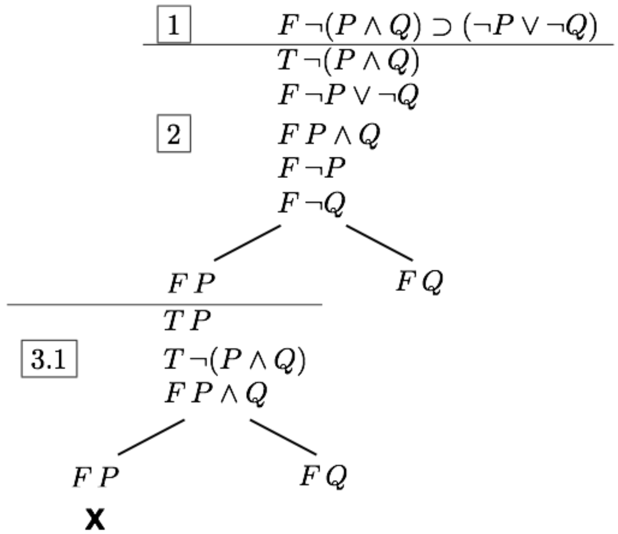
The only possibility now is to apply a destructive rule, moving to a new stage of the construction. We then apply all the rules we can that are *not* destructive. This includes branching.

$\boxed{1}$ $F\,\neg(P \wedge Q) \supset (\neg P \vee \neg Q)$

$T\,\neg(P \wedge Q)$
$F\,\neg P \vee \neg Q$
$\boxed{2}$ $F\,P \wedge Q$
$F\,\neg P$
$F\,\neg Q$

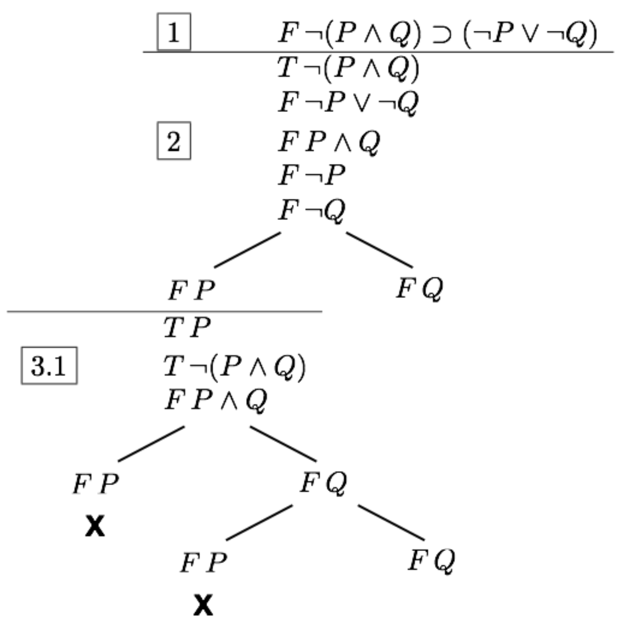
$F\,P$ $\qquad\qquad$ $F\,Q$

We have labeled this stage $\boxed{2}$, and we are (for now) only concerned with what is on the left branch including $F\,P$, but not the right $F\,Q$, and only below the horizontal line. If the left branch eventually closes, then we would shift our attention to the right branch. Let us say a tableau branch stage is *saturated* if all rules have been applied that are not destructive. The left branch (below the line) is saturated. We now have a choice of which destructive rule to apply, to $F\,\neg P$ or to $F\,\neg Q$. We can only apply one since working with either signed formula will eliminate the other. This is a *choice point*. We systematically explore each option, one at a time. We begin by applying a destructive rule to $F\,\neg P$. The rule application gets us $T\,P$, and we keep all T signed formulas, in this case just $T\,\neg(P \wedge Q)$. We then apply all non-destructive rules, including branching. The leftmost branch is closed, so we move on to the second branch from the left.



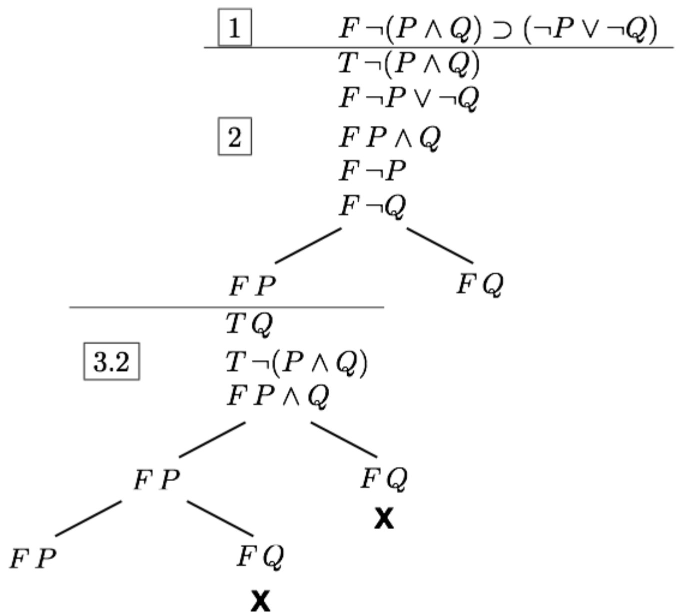
We have labeled this stage $\boxed{3.1}$ because it is stage 3 of our construction, but we had a choice point and we will backtrack if necessary to try a different choice, which we will call $\boxed{3.2}$. But that's for a little later. In the meantime, please note that we are talking about stages of construction for *a branch*, not for the tableau itself. Right now we are talking about the first open branch from the left.

We apply all possible non-destructive rules to the leftmost open branch, saturating it. In fact, this only amounts to applying the $F \wedge$ rule, introducing one more branching.

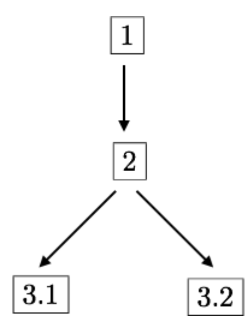


At this point we notice that the set of signed formulas on the lengthened open branch is the same as the set we had before we applied the branching rule. This branch will never close, and we can stop working on it. If we informally think of the set of signed formulas on an open branch as a partial description of a state in an Intuitionistic model, this repetition corresponds to a state being accessible to itself.

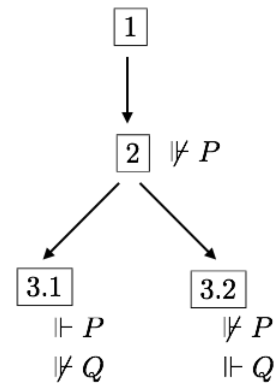
We did not get closure. But at the end of stage [2] we had a choice of destructive rule: work with $F\,\neg P$ or work with $F\,\neg Q$. We've now tried the first. Now we try the other, getting what we label [3.2]. Here it is, with the details of construction left out.



Again we find ourselves repeating, so this won't close either. We can't get a closed tableau at all, because we can't get the left branch from stage [2] to close. We now use this to construct a counter-model. The worlds or states are our stages. Accessibility is branch continuation after a destructive rule application. We get the following structure.



Arrows show the accessibility relation, though reflexive arrows are not shown. At the atomic level we do what the tableau says. For instance, at [3.2] we should have Q but not P because that's what the open branch for this stage says. We get the following. Atomic cases that are not specified won't matter.



Now it is left to you to check that we can work our way up each of the open branches we have been constructing in our tableau, verifying that each signed formula behaves in the model the way the branch says, so to speak. For instance, $F\, P \wedge Q$ is on the branch in stage $\boxed{3.2}$, and in the model, $\boxed{3.2} \nVdash P \wedge Q$ because $\boxed{3.2} \nVdash P$. We leave this to you, but after working our way up the branch the final step verifies that $\boxed{1} \nVdash \neg(P \wedge Q) \supset (\neg P \vee \neg Q)$, so we have our counter-model.

This has been an example. But a sketch of the general completeness argument goes as follows. Suppose X does not have a tableau proof. Start a tableau construction with $F\, X$, and construct it in a systematic way, as we did above, keeping track of the stages of the construction, based on choice points. Each branch will do one of three things. It will close. It will run out of new rules to apply. It will repeat an earlier stage. If the first two don't happen we could run the construction forever, but we must repeat because there are only a finite number of formulas that can appear, since all are subformulas of X itself. Thus our procedure must terminate, provided we stop once we see we are repeating.

Use the information from the tableau construction to create a model. The stages become the worlds or states. Accessibility comes out of the tableau construction directly. Atomic truth or falsity at states comes from what the open branch tells us. Finally, we can verify that what the branch tells us actually happens in the model. If you go through the example carefully you will see that everything we need to do this is right in front of us, because we never started a new stage until we had reached saturation, and a saturated branch has what we need to 'climb upwards'.