Testing syntactic sequents with tableau

Marianne: Hi there, I'm Marianne Talbot and in this video we are testing syntactic sequents with tableau. (Slide 2)This video supplements the fourth podcast in the Formal Logic series and as you go through, as always, be ready to pause the video as you attempt to answer the questions for yourself. It's always better to attempt to answer the questions first before going on to my answers.

(Slide 3) Here is a syntactic sequent. As you can see, it's a turnstile with just one crossbar, and as usual we read it, the set, curly brackets, counterexample set, is closed. (Slide 4) The set in the curly brackets is the counterexample set for this sequent, and it is the result, just check this out, of you negate the formula on the right hand side of the sequent and then remove the sequent.

Check that out, check that you understand exactly where the counterexample set is coming from before you go on.

(Slide 5) To evaluate a sequent by means of a tableau, it is always necessary to start by creating a counterexample set. If you do Podcast Four, for example, there are some exercises on counterexample sets in there that you can do for yourself.

(Slide 6) The counterexample set of an argument consists in the premises of the argument plus the *negation* of the conclusion. It is a set of sentences, it is not itself an argument and it runs *down* the page not *across* the page. The sequent runs across the page, the counterexample set runs down the page, and the only time you will see a counterexample set running along the page is if it is in curly brackets as we saw in the previous slide.

(Slide 7) So, if your formalised argument is that, the counterexample set is that. (Slide 8) The tableau tells us whether the counterexample set is consistent or not, and if the counterexample set is consistent, (so we are talking about if the set consisting of the premises plus the negation of the conclusion is inconsistent), then the argument is valid.

The counterexample is inconsistent if and only if there is no logically possible situation in which the sentences of the set are all true together. If the counterexample set is consistent then the argument is *invalid*. So a counterexample set is consistent if and only if there is a logically possible situation in which the sentences of the set are all true together.

There are a lot of words on that slide and some of these words are going to be new to you. Make sure you understand everything, every word in that slide before you move on to the next slide.

(Slide 9) Be very careful of the counter intuitive nature of this. If a counterexample set is *consisten*t, it means the argument is *invalid*. If the counterexample set in *inconsistent*, it means the argument is *valid*.

Of course actually we think that consistency and validity ought to go together but that's not the case. It is counter intuitive, make sure you understand that before you move on.

(Slide 10) The tableau works because it is a tree-like representation of the way the truth conditions of the sentences of the counterexample set can be combined. If one of the branches of the tree closes, it is because the branch contains an inconsistency, and you'll see what I mean by that in a minute.

(Slide 11) So, a branch contains an inconsistency whenever it contains a sentence letter and the negation of that sentence letter, so 'P' the sentence letter, 'not P' the negation of that sentence letter. As you can see, those two are inconsistent. If that appears anywhere in any of the branches of the tableau, the branch closes.

(Slide 12) Once we have created the counterexample set, our next task is to generate the tableau and we do this by following the tableau rules of propositional logic. (Slide 13) These rules are generated by the truth table definitions of the truth functors and they capture the truth conditions of each formula in the counterexample set and they demonstrate the impact of combining them.

(Slide 14) On page 20 of your handout booklet you will find all the rules for the propositional calculus and you will need those rules so you'll need that handbook if you are going to find the answers for yourself as you go through.

(Slide 15) Let's do a very simple tableau. Here we have one with just two premises and a conclusion. Two formulae on the left hand side of the sequent and one formula on the right hand side. The first step, as always, is to produce a counterexample and you need to do that before you move on to the next slide.

(Slide 16) Here is the counterexample set, down the page of course as counterexample sets always are. The second step is to start applying the rules, and here we have only got one rule to apply because two of the formulae are simple. We've only got one complex formula, so only one formula to which we can apply the rules. This is the rule for the conditional and you need to work out whether it is a stick going down the page or a branch, so taking the tableau into a branch like that.

What is the rule for the conditional? Is it a stick or a branch? Before you move on.

(Slide 17) The arrow rule is a branching rule, so you've got a left hand branch and a right hand branch, and what you need to know now is what goes on the left hand branch, before you change the slide again.

(Slide 18) Here we are. So, on the left hand branch we put a ‘not P’ because the negation of the antecedent is what goes on the left hand branch whenever you are applying the arrow rule. But we now have a branch in which we find both ‘P’ and ‘not P’. Clearly inconsistent, we can close this branch.

(Slide 19) Here we have a failed attempt to draw a possible situation in which all the sentences of the counterexample set are true, so we can close that off. We have still got an open branch; we’ve got the right hand branch open.

You need to look again at the rule for arrow and find out what goes on the right hand branch before you go to the next slide.

Oops, we’ve had that already.

(Slide 21) Okay, it’s a ‘Q’ on the right hand branch. Put that on, but now we’ve got a ‘Q’ and a ‘not Q’ which is another inconsistency. We can close that off and it’s another failed attempt to draw a possible situation in which all the sentences of the counterexample set are true together.

(Slide 22) So, our completed tableau tells us that every attempt to draw a possible situation in which the sentences of the counterexample set are all true together has failed. (Slide 23) The sentences of the counterexample set of the argument are inconsistent so the argument claim is, is it true or false? The argument is, is it valid or invalid?

Fill out both those blanks before moving on to the next slide.

(Slide 24) The sentences in the counterexample set of the argument are inconsistent so the argument claim is true and that means that the argument is valid. We know that for absolute certainty.

(Slide 25) Now let’s do a complex syntactic sequent. I think you will agree that this is a nice complex one. It only has two premises but both of the premises are very complex. A very simple conclusion, very simple formula on the right hand side of the sequent but two nasty complex ones on the left hand side of the sequent.

(Slide 26) Our first step as always is to create the counterexample set. Here it is. (Slide 27) There should be a line between the second premise and the conclusion, it would have made it just easier to see, but can you see you've got the two premises there exactly as they were in the counterexample set and then the negation of the conclusion? That is the counterexample set, set down the page.

(Slide 28) Now we have got to apply the rules to the formulae of this counterexample set. When you have had more practice you might be able to tell which formula you would apply the rules to first. The reason for choosing one over another is just to keep your tableau nice and manageable, it's nice if it doesn't branch too much because if you have too many branches you have got to fill them all in each time and that's a nuisance. But it doesn't really matter because you've got to apply the rules to all the formulae, so that's something that will just come with experience, don't worry about it at this stage.

(Slide 29) In applying the *truth table* rules to a formula, you apply them to the truth functor with the smallest scope first. But in applying the *tableau* rules to a formula, you apply them to the truth functor with the largest scope. That is a big difference between truth table rules and tableau rules.

(Slide 30) Let's start by applying the tableau rules to the first formula here, and that means we have got to decide which is the truth functor of that first, very complex formula. Which one of them has the largest scope? So we are only looking at Formula 1 and we are saying, "Which of those three truth functors has the largest scope?"

(Slide 31) There are three truth functors in Formula 1; there are two arrows and one conjunction. The one with the largest scope is the conditional between the two bracketed formulae. Can you see that? So the first arrow, the scope of that one is just the ‘P and Q’, the brackets over ‘P and Q’, and the scope of the conjunction is just the ‘S’ and the ‘R’, again the brackets will tell you that. The scope of the second conditional is the whole formula.

Make sure you understand that before you move on. We always apply the tableau rules in the largest scope first, next largest scope second and so on.

(Slide 32) If we are looking at Formula 1 and we are looking at the truth functional conditional with the largest scope, the middle one, do we need a stick rule or a branch rule for this conditional? Decide that before you move on to the next slide.

(Slide 33) The rule for this conditional is a branch rule and now you need to decide which formula should go on which branch.

(Slide 34) Now you will see that on the left hand side of the screen I am actually writing down what I'm doing on the screen, so I hope that helps you as you work out what is happening.

On the right hand side of the screen, here we have got the formula and what I've done is, on the left hand branch I've got the negation of the antecedent, check that out, the negation of the antecedent. On the right hand branch I've got the consequent exactly as it is. You will notice that I've put a little yellow tick there to show that I have finished with Formula 1.

Our next task is to apply the rules to the new formulae at level 4; you see the number 4 on the left hand side. We need to apply the rules both to the left hand branch and to the right hand branch.

So, you need to decide what the overall form is of both of these formulae and decide whether they are sticks or branches or one of each, one on each side. Decide that before you move on.

(Slide 35) Here we go again. Now again, we have a slide with a lot on it, so you'll need to linger over this slide and decide what everything on it means. You will see that on the left hand branch where I've got the negation of the antecedent of Formula 1, so that's 'not the case P arrow Q', that's a negated conditional, 'not the case P arrow Q'. As such, it's a stick rule. Check that out before you move on.

On the single branch we put the antecedent without a negation sign. Now, that immediately closes this branch, because again we've got a 'P' in the branch and a 'not P' in the counterexample set. There is no possible situation in which 'P' is true and 'not P' is true, so we can close off this branch. We don't have to bother with anything else.

But we have still got a branch open which is the formula on the right hand side. So, 'S and R' is a conjunction which is a stick rule, and on the single branch we put the 'S' and the 'R'. I see the 'S' has got slightly out of sync there with the 'R', but they are both on the same branch.

As I say, there is a lot on that slide, make sure you understand it. On the left hand side you've got the explanation of what's on the right hand side, make sure you understand everything here before you move on to the next slide.

(Slide 36) Here is the next slide. We have finished with Formula 1, we have analysed all of Formula 1, applied the rules to all of the formula and sub-formulae of it. Now we move on to Formula 2.

Again, we need to decide what is the overall form of this formula. Which is the truth functor with the largest scope? You'll see again, there are one, two, three, four, five truth functors on this one. There is a disjunction - an arrow, two arrows in fact - and two negation signs. Which one of those has the largest scope?

Having decided that, look at the rule for it. Is it a stick or is it a branch? Decide all of that before you go on to the next slide.

(Slide 37) You will see that actually the truth functor with the largest scope is the first conditional, so the scope of the disjunction side is just the 'R' and the 'T', and the scope of the first negation is just 'P', the scope of the second negation is just 'S', and the scope of the second conditional is 'not P if not S'. So it's the one in the middle that is the one with the largest scope. It's a conditional, therefore it's a branch rule.

Once again we put the branches on and you need to decide before we move on to the next slide, what we put on the left hand branch and what we put on the right hand branch.

(Slide 38) Here we are. On the left hand branch we put the negation of the antecedent of Formula 2, so check out that that's what is on there, and on the right hand branch we put the consequent of Formula 2 exactly as it is. Now we have applied the conditional rule to Formula 2.

We need to decide at level 6 what are the rules for each of these formulae here. We have got a negated disjunction on one side and a conditional on the other side. Don't be misled by those negation signs, they are both within the brackets so the overall form of the formula on the right hand side is just a conditional.

You need to decide, "Are those sticks or are they branches?" before we move on to the next slide.

(Slide 39) On the left hand side we have got a negated disjunction which is a stick rule, and on the right hand side we've got a conditional which is a branch rule. You need to apply the rules, decide which formulae go on which of those branches, before we move on to the next slide.

(slide 40) Here we are, we have done that. On left hand side we are putting just an ‘R’ because the negated disjunction is a stick rule and on one side goes an ‘R’. Notice something important about that branch before we move on…..

If we look at the other side, we’ve got the negation of the antecedent of the conditional, so the antecedent is ‘not P’. We’ve got its negation which is ‘not-not-P’ and then on the right hand branch of the conditional we’ve got a ‘not-S’ which is just the consequent of the conditional exactly as it is.

Check those out and then have a look and see if you can find something important before we move on to the next slide.

(Slide 41) What is important of course is that actually there is an inconsistency in every branch. If you look at the second branch we’ve got a ‘not R’ and an ‘R’, and on the third branch we’ve got a ‘not-not-P’ and a ‘not P’ up there in the counterexample set. In the fourth branch we’ve got a ‘not-S’ and slightly further up we’ve got an ‘S’. So all of those branches close and you should be able to see what the impact of that is.

The argument claim is, is it true is it false? The argument is, valid or invalid? Decide that before we move to the next slide.

(Slide 42) Good. So all the branches close, so the argument claim is correct and the argument is valid. We know that for absolute certainty, we have applied the rules and that’s what we have got.

(Slide 43) So, you know now how to use tableau to test syntactic sequents. Well done.