MAU22C00 Lecture 6

John Stalker

Trinity College Dublin

Tableaux

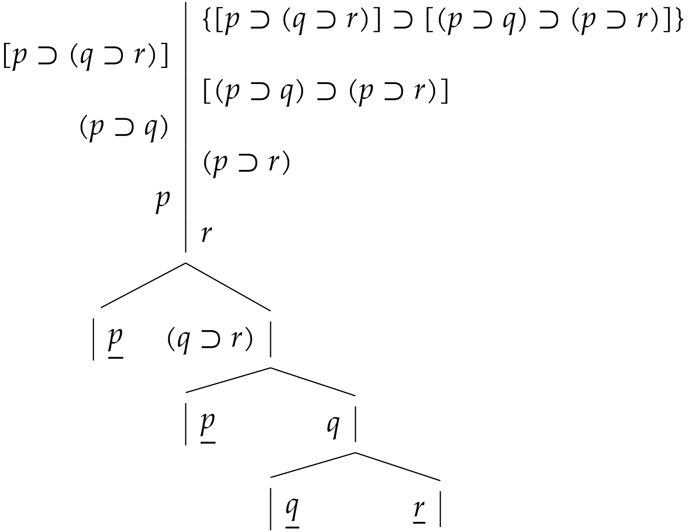

Figure 1: An analytic tableau

Notational conventions, procedure

Statements to the left of the vertical line are true. Statements to the right are false.

Since this is proof by contradiction we start with the statement we want to prove on the right.

We gradually draw out the consequences of each statement, paying attention to which side of the line it's on, asking ourselves "how could this be true?" or "how could this be false?".

When there are two possibilities we split the line, which gives us a tree structure.

When we see a contradiction, i.e. an expression appearing on both sides of the line, we underline it and close off that branch.

If we can close all branches then we are done.

For more examples and more detailed discussion see the notes.

Tableaux advice

The order in which you process statements won't affect whether the tableau closes, but it may affect how fast it closes.

Try to process all statements which don't require branching before any which do.

If you have to branch, try to find a statement where you'll be able to close a branch immediately.

Once a statement has been used (on a branch) there's no point in ever using it again (on that branch or any of its subbranches).

Once a branch is closed there's no point in extending it further.

What if we can't close a branch?

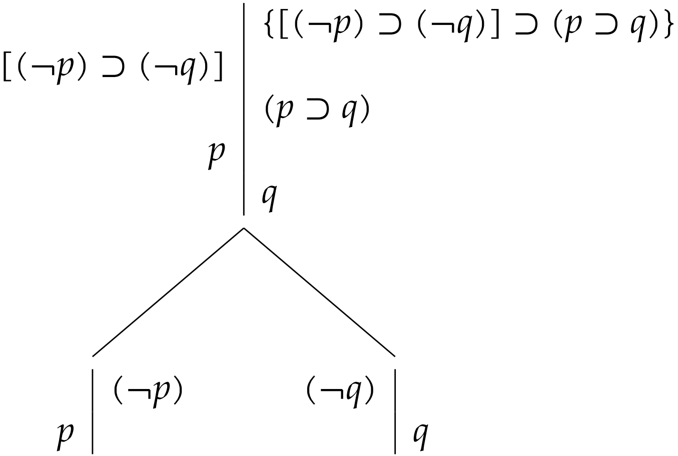

Figure 2: An analytic tableau which doesn't close

We've just failed to prove that $\{[(\neg p) \supset (\neg q)] \supset (p \supset q)\}$ is a tautology, because it isn't. Looking at the left branch we see that p true and q false gives a counter-example.

Why does the tableau method work?

Every time we process an expression we get one or two shorter expressions. Expressions can't get shorter than a single variable, so eventually we must stop.

If we get contradictions in all branches then we know the statement we started ith is unsatisfiable, if it was to the left of the line, or a tautology, if it was to the right.

If there's a branch with no contradiction then no variable appears to both the left and right of the line.

Assign every variable on the left the value true and every variable on the right the value false. If a variable doesn't appear then assign it either value.

This will give an example of the statement we started with, if it was on the left, or a counter-example, if it was on the right.

Why? Consider the *shortest* expression on the wrong side of the line, having assigned truth values as above. It's not a single variable, but it's also not a compound expression, because it's component(s) is/are on the correct side. So no expression is on the wrong side, including the one we started from.

What you can do with tableaux

- Show that a statement is a tautology: put it on the right and show that every branch closes.
- Give a counter-example to show that a statement is not a tautology: put it on the right, find a branch which doesn't close, and assign values to variables.
- Give an example to show that a statement is satisfiable: put it on the left, find a branch which doesn't close, and assign values to variables.
- Show that a statement is not satisfiable: put it on the left and show that every branch closes.
- Show that one statement is a consequence of others: put it on the right and them on the left and show that every branch closes.
- Give a counter-example to show that one statement is not a consequence of others: put it on the right and them on the left, find a branch which doesn't close, and assign values to variables.

Formal systems

A *formal system* consists of a formal language, a set of axioms, and a set of rules of inference.

There's a bit more to it. Either the axioms need to be a finite set or we need to be able to decide whether a given statement is an axiom. The rules of inference need to be such that we can determine whether on statement is derived from others via rules of inference.

A *proof* of a statement in a formal system is a list of statements, ending with that one, each of which is an axiom or follows from earlier statements by the rules of inference.

A statement for which there is a proof is called a *theorem*.

Interpretation is not part of a formal system, although there's usually at least one intended interpretation.

An interpretation is *sound* if the axioms are true and the rules of inference when applied to true statements give true statements.

Formal systems, continued

If the interpretation is sound then every theorem is true.

If every true statement is a theorem then the system is called *complete*.

A system is called *consistent* if it is free from contradictions, i.e. if no statement and its negation are both theorems.

Soundness and completeness depend on the interpretation. Consistency is mostly independent of the interpretation. We only need to know what negation looks like in our system.

If you believe that no statement is both true and false then soundness implies consistency.

Formal systems for zeroeth order logic

There are two main classes of formal systems for logic: axiomatic systems and "natural deduction" systems.

Axiomatic systems have few rules of inference, often only one, and rely mostly on axioms.

Historically they are older.

Natural deductive systems have few, often zero, axioms and rely mostly on rules of inference.

They arose as a reaction against axiomatic systems.

Informal proofs look much more like proofs in a natural deduction system than in an axiomatic system.

For example, it's generally easier to convert a tableau proof into a natural deduction proof than an axiomatic one.