5 Consistency of Analytic Tableaux

Last week we introduced the tableau method for proving formulas correct. Let us briefly review the concepts.

A tableau proof for a formula X is constructed by starting with a $\boxed{\text{signed formula}}$ FX. We then $\boxed{\text{extend}}$ the basic tableau, using proof rules to either add $\boxed{\text{direct consequences}}$ of a signed formula to the tree or two $\boxed{\text{branches}}$, indicating two alternative possibilities. We proceed until each branch is either $\boxed{\text{closed}}$, i.e. a formula and its $\boxed{\text{conjugate}}$ occur on it, or $\boxed{\text{complete}}$, i.e. it cannot be extended anymore. If the tableau is close, we have a $\boxed{\text{proof}}$, otherwise we find a counterexample in one of the branches.

For the 4 connectives and the two signs there are altogether 8 $\boxed{\text{proof rules}}$, each of them derived from the axioms of boolean valuations. One of the fascinating things about Smullyan's approach is that he simplifies the the description of the proof calculus by introducing a $\boxed{\text{uniform notation}}$ for these rules. We associated $\boxed{\text{types } \alpha \text{ and } \beta}$ to signed formulas according to the behavior of the rules. Because of this, we only need two basic rules:

$$\frac{\alpha}{\begin{array}{c}\alpha_1\\ \alpha_2\end{array}} \qquad \frac{\beta}{\beta_1 \mid \beta_2}$$

This notation will turn out to be extremely helpful when we have to prove properties of the calculus, like correctness and completeness.

Tableaux systems for unsigned formulas

The use of signs T and F in tableaux proofs is heuristically helpful, but not absolutely necessary. One could also work on unsigned formulas, deleting every T and replacing F by a negation symbol and get the same results.

We could do this either in the 8 rules or in the definition of formula types. For the latter, the tables would be as follows.

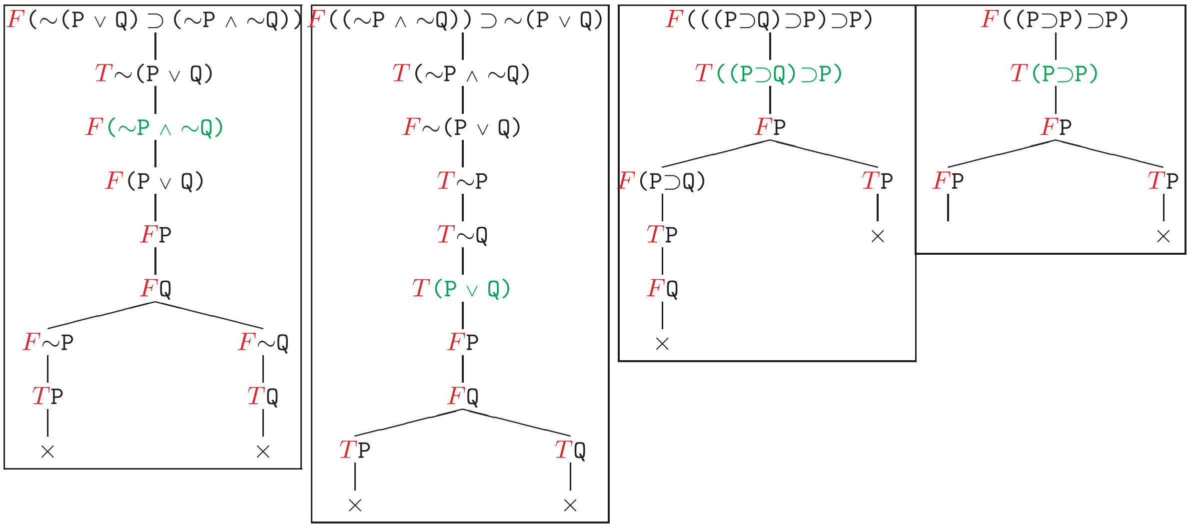
α	$(X \wedge Y)$	$\sim(X \vee Y)$	$\sim(X \supset Y)$	$\sim\sim X$
α_1	X	$\sim X$	X	X
α_2	Y	$\sim Y$	$\sim Y$	–

β	$\sim(X \wedge Y)$	$(X \vee Y)$	$(X \supset Y)$
β_1	$\sim X$	X	$\sim X$
β_2	$\sim Y$	Y	Y

Apart from that the tableaux method for unsigned formulas is identical to the one for signed formulas. This is another advantage of the abstract classification.

Examples

It is important that you practice developing tableau proofs. Let's look at two examples, $(\sim(P \vee Q) \supset (\sim P \wedge \sim Q))$ and it's reverse $((\sim P \wedge \sim Q)) \supset \sim(P \vee Q)$. We'll follow the "α-nodes first" strategy (β nodes are marked green)



Here is a third one, $(((P \supset Q) \supset P) \supset P)$. Before we try to prove it or find a counterexample, let's take a poll. Who thinks this strange formula is true "*if the fact that P implies something implies P, then P is true*"? Well, let's check it out.

Actually, the Q didn't really matter in the proof as we see. Does that mean we can simply take it out? No, since in this case we get FP as a counterexample.

5.1 Proving Properties of formal Systems

Now that we understood how the method works and have defined tableau proofs, we need to make sure that the tableau method really works. We have to show that any formula that has a tableau proof is in fact true (a tautology), and that any true formula does have a tableau proof. Logicians call this *correctness* and *completeness* of the logical system.

Often it is easier to prove the *consistency* of a proof method instead of proving its correctness. A proof method is consistent, if it doesn't contradict itself, that is if we cannot prove both a formula and its negation in that system – in other words, we cannot prove the same formula both true and false.

If a proof system is complete, then consistency and correctness are the same and that's why the two notions are often used interchangeably.

Q: *Why does consistency and completeness imply correctness?*

Showing consistency and completeness strongly depends on they way a logical system has been constructed and the methods for proving these properties vary drastically. It is fairly easy for truth-tables, since every row in the table corresponds to exactly one interpretation.

Correctness of truth-table proofs: If we have a truth-table proof for X then every row ends with a `t` for the formula X. Since every entry in every row stands for the value of the corresponding formula under the interpretation, the value of X is true under every interpretation and X is a tautology.

Completeness of truth-table proofs: If X is a tautology, then it is true under every interpretation, and hence every row in the truth table for X ends with a `t`. Hence X has a truth-table proof.

However, for more elaborate and efficient proof systems, showing consistency and completeness becomes increasingly complicated, as the notion of "true under every interpretation" becomes buried under the technicalities of the proof method. Proving these properties for the tableau method is considered of low to moderate difficulty.

5.2 Consistency

To prove the correctness of the tableau method, we have to show that *the origin of a closed tableau is unsatisfiable*.

The rationale for that is the following. If X has a tableau proof then there is a closed tableau for FX. If that implies that FX is unsatisfiable, then X must be a tautology.

We proceed by contraposition, showing that a tableau is satisfiable and cannot be closed whenever the formula at its origin is satisfiable. Let us attempt a rigorous proof – you will find a careful proof in plain English in Smullyan's book.

Notation: $\mathsf{true}(\theta,\mathsf{v}_0) \equiv \forall \mathsf{Y}{:}\mathsf{S\text{-}FORM}.\ \mathsf{Y}\ \mathsf{on}\ \theta \mapsto \mathsf{S\text{-}value}(\mathsf{Y},\mathsf{v}_0)\mathsf{=t}$

Lemma 1: $\forall \mathsf{X}{:}\mathsf{FORM}.\ \forall \mathcal{T}{:}\mathsf{Tableaux}_X.\ \mathsf{closed}(\mathcal{T}) \mapsto \forall \mathsf{v}_0{:}\mathsf{Var}_X{\to}\mathbb{B}.\ \neg(\exists\theta{:}\mathsf{path}(\mathcal{T}).\ \mathsf{true}(\theta,\mathsf{v}_0))$

proof: $\mathsf{closed}(\mathcal{T}) \mapsto \forall\theta{:}\mathsf{path}(\mathcal{T}).\ \exists \mathsf{Y}{:}\mathsf{S\text{-}FORM}.\ \mathsf{Y}\ \mathsf{on}\ \theta \wedge \bar{\mathsf{Y}}\ \mathsf{on}\ \theta$
$\qquad\qquad\qquad \mapsto \forall \mathsf{v}_0{:}\mathsf{Var}_X{\to}\mathbb{B}.\ \forall\theta{:}\mathsf{path}(\mathcal{T}).\ \exists \mathsf{Y}{:}\mathsf{S\text{-}FORM}.\ \mathsf{S\text{-}value}(\mathsf{Y},\mathsf{v}_0)\mathsf{=f}.$

Lemma 2: $\forall \mathsf{X}{:}\mathsf{FORM}.\ \forall \mathcal{T}{:}\mathsf{Tableaux}_X.\ \forall \mathsf{v}_0{:}\mathsf{Var}_X{\to}\mathbb{B}.$
$\mathsf{S\text{-}value}(\mathsf{origin}(\mathcal{T}),\mathsf{v}_0))\mathsf{=t} \mapsto \exists\theta{:}\mathsf{path}(\mathcal{T}).\ \mathsf{true}(\theta,\mathsf{v}_0)$

proof: Structural induction on tableau trees.

base case: If $\mathcal{T}$ has just a single point, then chose $\theta = [\mathsf{origin}(\mathcal{T})]$.
Now $\mathsf{S\text{-}value}(\mathsf{origin}(\mathcal{T}),\mathsf{v}_0))\mathsf{=t}$ implies $\mathsf{true}(\theta,\mathsf{v}_0)$

step case: Assume the lemma holds for some $\mathcal{T}$ and let $\mathcal{T}_1$ be a direct extension of $\mathcal{T}$ and $\mathsf{v}_0{:}\mathsf{Var}_X{\to}\mathbb{B}$ such that $\mathsf{S\text{-}value}(\mathsf{origin}(\mathcal{T}_1),\mathsf{v}_0))\mathsf{=t}$.
Since $\mathsf{origin}(\mathcal{T}_1)\mathsf{=}\mathsf{origin}(\mathcal{T})$, there is some $\theta{:}\mathsf{path}(\mathcal{T})$ such that $\mathsf{true}(\theta,\mathsf{v}_0)$.
Consider 3 cases:
- If $\mathcal{T}_1$ does *not* extend $\mathcal{T}$ at θ, then $\theta \in \mathrm{path}(\mathcal{T}_1)$ and $\mathsf{true}(\theta,\mathsf{v}_0)$.
- If $\mathcal{T}_1$ extends $\mathcal{T}$ at θ by some α_i, then we know α on θ and $\mathsf{S\text{-}value}(\alpha,\mathsf{v}_0)\mathsf{=t}$, thus $\mathsf{S\text{-}value}(\alpha_i,\mathsf{v}_0)\mathsf{=t}$. Choose $\theta_1\mathsf{=}\theta\circ\alpha_i$ then $\theta_1 \in \mathsf{path}(\mathcal{T}_1)$ and $\mathsf{true}(\theta_1,\mathsf{v}_0)$.
- If $\mathcal{T}_1$ extends $\mathcal{T}$ at θ by β_1 and β_2 then we know β on θ and $\mathsf{S\text{-}value}(\beta,\mathsf{v}_0)\mathsf{=t}$, thus $\mathsf{S\text{-}value}(\beta_i,\mathsf{v}_0)\mathsf{=t}$ for some i. Choose $\theta_1\mathsf{=}\theta\circ\beta_i$ then $\theta_1 \in \mathsf{path}(\mathcal{T}_1)$ and $\mathsf{true}(\theta_1,\mathsf{v}_0)$.

Corollary 3: $\forall$X:FORM. $\forall\mathcal{T}$:Tableaux$_X$. closed($\mathcal{T}$) $\mapsto$ $\forall$v$_0$:Var$_X\rightarrow\mathbb{B}$. S-value(origin($\mathcal{T}$),v$_0$)=f

Corollary 4: $\forall$X:FORM. $\forall\mathcal{T}$:Tableaux$_X$. closed($\mathcal{T}$) $\mapsto$ $\forall$v$_0$:Var$_X\rightarrow\mathbb{B}$. Value(X,v$_0$)=t

Theorem 5: $\forall$X:SFORM. ($\exists\mathcal{T}$:Tableaux$_X$. closed($\mathcal{T}$)) $\mapsto$ X tautology

Lemma 2 represents the key idea of the proof, stating that the tableau method will always find a counterexample for X if FX is satisfiable. The main argument is based on the properties of Boolean valuations expressed in uniform notation, which saves us a lot of work.

We chose a fairly rigorous proof, because this shows how to present a completely formal account of the correctness of the tableau method. *In fact, Jim Caldwell has given such a formal account with our Nuprl proof development system. He has developed a formal constructive proof of the decidability of classical propositional logic and extracted a tableau prover from that proof. If you're interested, consult his thesis.*