COMP0009, Properties of Tableaus

October 22, 2023

Termination of propositional tableau algorithm

Termination of propositional tableau algorithm

- When running tableau algorithm for ϕ, only new theories are enqueued.

Termination of propositional tableau algorithm

- When running tableau algorithm for ϕ, only new theories are enqueued.
- Let X be set of subformulas of ϕ and single negations of subformulas of ϕ — at most $2|\phi|$ of these.

Termination of propositional tableau algorithm

- When running tableau algorithm for ϕ, only new theories are enqueued.
- Let X be set of subformulas of ϕ and single negations of subformulas of ϕ — at most $2|\phi|$ of these.
- A theory is a subset of X — at most $2^{2|\phi|}$ of these.

Termination of propositional tableau algorithm

- When running tableau algorithm for ϕ, only new theories are enqueued.
- Let X be set of subformulas of ϕ and single negations of subformulas of ϕ — at most $2|\phi|$ of these.
- A theory is a subset of X — at most $2^{2|\phi|}$ of these.
- Algorithm terminates in $2^{2|\phi|}$ steps (at most).

Soundness for Tableaus

$$\left.\begin{array}{l} \vdash \phi \\ \text{Tab}(\neg\phi) \text{ closes} \\ \text{Tab}(\psi) \text{ closes} \end{array}\right\} \Rightarrow \left\{\begin{array}{l} \models \phi \\ \neg\phi \text{ unsat} \\ \psi \text{ unsat} \end{array}\right.$$

Soundness for Tableaus

$$\frac{\left.\begin{array}{l} \vdash \phi \\ \text{Tab}(\neg\phi) \text{ closes} \\ \text{Tab}(\psi) \text{ closes} \end{array}\right\} \Rightarrow \left\{\begin{array}{l} \models \phi \\ \neg\phi \text{ unsat} \\ \psi \text{ unsat} \end{array}\right.}{\text{Tab}(\psi) \text{ never closes} \Leftarrow \psi \text{ is sat}}$$

Soundness of Propositional Tableau Algorithm

If ϕ is satisfiable then tableau for ϕ cannot close, so algorithm outputs SATISFIABLE eventually.

Proof.

Assume $v(\phi) = \top$. Prove by induction on number n of iterations of while statement that there is $\Sigma \in \mathsf{Tab}$ where $\theta \in \Sigma \to \mathsf{v}(\theta) = \top$.
True for n = 0 by initialisation and assumption.
Assume after n iterations that there is $\Sigma \in \mathsf{Tab}$ such that $\theta \in \Sigma \to \mathsf{v}(\theta) = \top$.
If Σ is dequeued and $\psi \in \Sigma$ is picked we have $\mathsf{v}(\psi) = \top$ (by IH).
If ψ is an α then $\mathsf{v}(\alpha_1) = \mathsf{v}(\alpha_2) = \top$ so $\theta \in \Sigma[\alpha/\{\alpha_1, \alpha_2\}] \to \mathsf{v}(\theta) = \top$ still true.
If ψ is a β then either $\mathsf{v}(\beta_1) = \top$ or $\mathsf{v}(\beta_2) = \top$, so either $\theta \in \Sigma_1 \to \mathsf{v}(\theta) = \top$ or $\theta \in \Sigma_2 \to \mathsf{v}(\theta) = \top$.
Result follows by induction over n.

$\square$

Soundness of Predicate Tableau Algorithm

Assume ϕ has a model.
Prove by induction on number of iterations of while loop that there is $\Sigma \in \mathsf{Tab}$ and $\mathsf{S} \models_{\mathsf{A}} \Sigma$ (some S, A).
True for $\mathsf{n} = 0$ by assumption.
Assume $\Sigma \in \mathsf{Tab}$ and $\mathsf{S} \models_{\mathsf{A}} \Sigma$.
When Σ is dequeued and ψ is picked, cases where ψ is α or β are same as propositional case, no change to S, A needed.
If ψ is δ, say $\psi = \exists \mathsf{x} \theta(\mathsf{x})$ then $\mathsf{S} \models_{\mathsf{A}} \exists \mathsf{x} \theta(\mathsf{x})$ (by IH).
So there is $\mathsf{s} \in \mathsf{S}$ such that $\mathsf{S}_{\mathsf{A}[\mathsf{x} \to \mathsf{s}]} \models \theta(\mathsf{x})$.
$\exists \mathsf{x} \theta(\mathsf{x})$ gets replaced by $\theta(\mathsf{c})$ in Σ (new constant c).
Let S' be same as S except $\mathsf{I}(\mathsf{c}) = \mathsf{s}$. Then $\mathsf{S}' \models_{\mathsf{A}} \Sigma[\exists \mathsf{x} \theta(\mathsf{x}) / \theta(\mathsf{c})]$.
If ψ is a γ, say $\psi = \forall \mathsf{x} \theta(\mathsf{x})$ then $\mathsf{S} \models_{\mathsf{A}} \forall \mathsf{x} \theta(\mathsf{x})$ (I.H.)
Follows that $\mathsf{S} \models_{\mathsf{A}} \theta(\mathsf{t})$ for any closed term t.
$\mathsf{S} \models_{\mathsf{A}} \Sigma[\forall \mathsf{x} \theta(\mathsf{x}) / \theta(\mathsf{t})]$ still true (no need to change S, A in this case).

Tableau proof from hypothesis

As with axiomatic proofs, you can use tableau to prove from hypotheses.

Tableau proof from hypothesis

As with axiomatic proofs, you can use tableau to prove from hypotheses.

Let Γ be a set of hypotheses, let ϕ be a formula.

Tableau proof from hypothesis

As with axiomatic proofs, you can use tableau to prove from hypotheses.

Let Γ be a set of hypotheses, let ϕ be a formula.

For tableau as list of theories, initialise as $[\Gamma \cup \{\neg\phi\}]$.

If list becomes empty, write $\Gamma \vdash \phi$

Tableau proof from hypothesis

As with axiomatic proofs, you can use tableau to prove from hypotheses.

Let Γ be a set of hypotheses, let ϕ be a formula.

For tableau as list of theories, initialise as $[\Gamma \cup \{\neg\phi\}]$.

If list becomes empty, write $\Gamma \vdash \phi$

For tableau as tree, start with $\neg\phi$ at root.

Expand as before, but add new rule — at any stage you can pick $\gamma \in \Gamma$ and add node labelled γ at any leaf.

If tableau eventually closes, we may write

$$\Gamma \vdash \phi$$

Tableau proof from hypothesis

As with axiomatic proofs, you can use tableau to prove from hypotheses.
Let Γ be a set of hypotheses, let ϕ be a formula.
For tableau as list of theories, initialise as $[\Gamma \cup \{\neg\phi\}]$.
If list becomes empty, write $\Gamma \vdash \phi$
For tableau as tree, start with $\neg\phi$ at root.
Expand as before, but add new rule — at any stage you can pick $\gamma \in \Gamma$ and add node labelled γ at any leaf.
If tableau eventually closes, we may write

$$\Gamma \vdash \phi$$

If Γ is a finite set, say $\{\gamma_0, \gamma_1 \ldots, \gamma_{n-1}\}$ you may as well just start with $\neg\phi, \gamma_0, \ldots, \gamma_{n-1}$ in a single branch of new tableau and try to close it.

Ancestors

If $\Sigma \in \mathsf{Tab}$ is dequeued and Σ_1 (Σ_2) are enqueued, then Σ is parent of Σ_1 (Σ_2)

$$\begin{aligned} \mathsf{P}(\Sigma) &= \Sigma' \text{ if } \Sigma' \text{ is the parent of } \Sigma \\ \mathsf{P}^0(\Sigma) &= \Sigma \\ \mathsf{P}^{n+1}(\Sigma) &= \mathsf{P}(\mathsf{P}^n(\Sigma)) \end{aligned}$$

Say Σ is ancestor of Σ' if there is $n \geq 0$ and $\mathsf{P}^n(\Sigma') = \Sigma$.
Initial tableau $[\{\phi\}]$ and $\{\phi\}$ is ancestor of every theory in the tableau.

Completeness for Tableaus

$$\left.\begin{array}{l} \vdash \phi \\ \text{Tab}(\neg\phi) \text{ closes} \\ \text{Tab}(\psi) \text{ closes} \end{array}\right\} \Leftarrow \left\{ \begin{array}{l} \models \phi \\ \neg\phi \text{ unsat} \\ \psi \text{ unsat} \end{array}\right.$$

Completeness for Tableaus

$$\dfrac{\left.\begin{array}{l} \vdash \phi \\ \text{Tab}(\neg\phi) \text{ closes} \\ \text{Tab}(\psi) \text{ closes} \end{array}\right\} \Leftarrow \left\{\begin{array}{l} \models \phi \\ \neg\phi \text{ unsat} \\ \psi \text{ unsat} \end{array}\right.}{\text{Tab}(\psi) \text{ never closes } \Rightarrow \psi \text{ is sat}}$$

Completeness for Tableaus

$$\dfrac{\left.\begin{array}{l}\vdash \phi \\ \text{Tab}(\neg\phi) \text{ closes} \\ \text{Tab}(\psi) \text{ closes}\end{array}\right\} \Leftarrow \left\{\begin{array}{l}\models \phi \\ \neg\phi \text{ unsat} \\ \psi \text{ unsat}\end{array}\right.}{\text{Tab}(\psi) \text{ never closes \ (with fair schedule)} \Rightarrow \psi \text{ is sat}}$$

Completeness of Propositional Tableau Algorithm

Start tableau with ϕ. If algorithm outputs SATISFIABLE then there is $v : \text{Props} \to \{\top, \bot\}$ such that $v(\phi) = \top$, so ϕ is satisfiable.

Proof.

Output SATISFIABLE $\Rightarrow \Sigma \in$ Tab dequeued $\wedge \text{Exp}(\Sigma) \wedge \neg C(\Sigma)$
Define v by $v(p) = \top \iff p \in \Sigma$.
We have $\phi \in \Sigma \Rightarrow v(\phi) = \top$
We prove, if $(\phi \in \Sigma' \Rightarrow v(\phi) = \top)$ then $(\phi \in P(\Sigma') \Rightarrow v(\phi) = \top)$.
For this, one formula is replaced in $P(\Sigma')$ by expansion formula(s) in Σ'.
If it is an α, then $\alpha_1, \alpha_2 \in \Sigma'$, so $v(\alpha_1) = v(\alpha_2) = \top$, hence $v(\alpha) = \top$
If it is a β, then either $\beta_1 \in \Sigma'$ or $\beta_2 \in \Sigma'$, so either $v(\beta_1) = \top$ or $v(\beta_2) = \top$, either way $v(\beta) = \top$.
Other formulas in $P(\Sigma')$ don't change in Σ'.
It follows (by induction) that all formulas in the original ancestor theory are true under v.
Hence $v(\phi) = \top$, so ϕ is satisfiable.

$\square$

Herbrand Structures

A closed term t is built up from constants and function symbols only — no variables.

A Herbrand structure $H = (D, I)$ has

Domain

$$D = \{\text{closed terms}\}$$

Interpretation $I = (I_c, I_f, I_p)$.

$$\begin{array}{rcl} I_c(c) & = & c \\ I_f(f^n) : (d_1, \ldots, d_n) & \mapsto & f^n(d_1, \ldots, d_n) \end{array}$$

I_p can be chosen freely.

It follows, for any closed term t, that

$$[t]^{H,A} = t$$

Herbrand Theorem

Let L be a language with ∞ many constant symbols (and no equality predicate in this version of the theorem).
If ϕ is satisfiable (i.e. $S \models_A \phi$, some S some A) then ϕ is satisfiable in a Herbrand model H, i.e. $H \models_A \phi$ (some A).

Fairness

Suppose you have several (countably many) processes $P_1, P_2, \ldots, P_k, \ldots$ and each of them is waiting for some input. It might be that when you give P_i some input, it creates a new process P_{k+1}, so the list can grow, but it will always be countable. In what order should you supply inputs to the various processes? You could simply supply input to P_1 again and again, but that would be unfair to all the other processes. In a fair schedule, if any process P_i is waiting for input at time t then eventually (at some time $t' > t$) P_i will get some input. If a process is always waiting for input, then it will get input infinitely often. Since the total number of requests for input is countable, it is possible to find a fair schedule.

Rank

$$\begin{aligned}
\mathsf{Rk}(\mathsf{P}(\mathsf{t}_0,\ldots,\mathsf{t}_{\mathsf{k}-1})) &= 1 \\
\mathsf{Rk}(\neg\phi) &= 1 + \mathsf{Rk}(\phi) \\
\mathsf{Rk}((\phi \circ \psi)) &= 1 + \mathsf{Rk}(\phi) + \mathsf{Rk}(\psi) \\
\mathsf{Rk}(\exists \mathsf{x}\phi) &= 1 + \mathsf{Rk}(\phi) \\
\mathsf{Rk}(\forall \mathsf{x}\phi) &= 1 + \mathsf{Rk}(\phi)
\end{aligned}$$

So $\mathsf{Rk}(\phi)$ is number of nodes in parse tree.

Formula Induction

Base case Prove $\mathsf{Rk}(\phi) \leq 2 \Rightarrow \pi(\phi)$

Induction Hypothesis Assume (for some $\mathsf{k} \geq 2$) that

$$\mathsf{Rk}(\phi) \leq \mathsf{k} \Rightarrow \pi(\phi)$$

Induction Step Prove that $\mathsf{Rk}(\phi) \leq \mathsf{k} + 1 \Rightarrow \pi(\phi)$.

If you prove the base case, and prove the induction step using induction hypothesis, then you can conclude

$$\pi(\phi)$$

for all FO formulas ϕ.

Completeness of algorithm for predicate tableau

If tableau for ϕ never closes and expanded by a fair schedule then ϕ is satisfiable.

Proof: If tableau never closes, by König's tree lemma there is a sequence $\Sigma_0, \Sigma_1, \Sigma_2, \ldots \in \mathsf{Tab}$ where $\Sigma_n = P(\Sigma_{n+1})$. Let $\Sigma = \bigcup_{n<\infty} \Sigma_n$.
Since fair schedule was used,

$$\alpha \in \Sigma \Rightarrow \alpha_1 \in \Sigma \text{ and } \alpha_2 \in \Sigma$$

$$\beta \in \Sigma \Rightarrow \beta_1 \in \Sigma \text{ or } \beta_2 \in \Sigma$$

$$\exists x \theta(x) \in \Sigma \Rightarrow \theta(c) \in \Sigma (\text{some } c)$$

$$\neg\forall x \theta(x) \in \Sigma \Rightarrow \neg\theta(c) \in \Sigma \ (\text{some } c)$$

$$\forall x \theta(x) \in \Sigma \Rightarrow \theta(t) \in \Sigma \ (\text{all closed terms } t)$$

$$\neg\exists x \theta(x) \Rightarrow$$

Completeness, continued

Let H be Herbrandt structure, base $\{\text{closed terms of } \Sigma\}$, $\mathsf{I}(\mathsf{t}) = \mathsf{t}$ (closed term t) and
$(\mathsf{t}_0, \mathsf{t}_1, \ldots, \mathsf{t}_{\mathsf{k}-1}) \in \mathsf{I}(\mathsf{R}^\mathsf{n}) \iff \mathsf{R}^\mathsf{n}(\mathsf{t}_0, \ldots, \mathsf{t}_{\mathsf{n}-1}) \in \Sigma$.
Note that rank of expansion formulas is strictly less than rank of formula.
Let θ be any sentence. Prove by induction on $\mathsf{Rk}(\theta)$ that

$$\theta \in \Sigma \Rightarrow \mathsf{H} \models \theta.$$

Hence $\mathsf{H} \models \phi$.

Equality Rules

$$\frac{\begin{array}{c}A(t)\\ t = s\end{array}}{A(s)} \qquad \frac{\begin{array}{c}A(t)\\ s = t\end{array}}{A(s)} \qquad \frac{\neg(t = t)}{x}$$

Example $s = t \vdash t = s$

1. $s = t$ (hypothesis)
2. $\neg(t = s)$ (to prove $t = s$)

Example $s = t \vdash t = s$

1. $s = t$ (hypothesis)
2. $\neg(t = s)$ (to prove $t = s$)
3. $\neg(s = s)$ (from (2), equality rule)

Example $s = t \vdash t = s$

1. $s = t$ (hypothesis)
2. $\neg(t = s)$ (to prove $t = s$)
3. $\neg(s = s)$ (from (2), equality rule)
4. — closed, by 3rd equality rule

Tableau Summary

- Tableau method is sound and complete for first order logic (this is, essentially, Gödel's completeness theorem).
- If ϕ is not satisfiable its tableau will close finitely, provided a fair sequence is used (completeness).
- If ϕ is satisfiable its tableau will never close (soundness).
- But a tableau construction may never terminate.