Tableau-based theorem proving

December 8, 2023

Introduction

The method of analytic tableaux

- is a proof procedure for first order logic
- is complete: If a formula has a proof, the search will succeed
- if the formula has no proof, we might search forever (RE-problem)
- simple special case: propositional logic

One-sided Sequent calculus (Propositional)

Assume formulas are put in NNF.

$$\dfrac{}{\Gamma, \phi, \neg\phi \vdash}\ \text{Hyp}$$

$$\dfrac{\Gamma, \phi, \psi \vdash}{\Gamma, \phi \wedge \psi \vdash}\ \text{And}$$

$$\dfrac{\Gamma, \phi \vdash \qquad \Gamma, \psi \vdash}{\Gamma, \phi \vee \psi \vdash}\ \text{Or}$$

Example:

$$\dfrac{\dfrac{\dfrac{}{a, b, \neg a}\ \text{Hyp} \qquad \dfrac{}{a, b, \neg b}\ \text{Hyp}}{a, b, \neg a \vee \neg b}\ \text{Or}}{a \wedge b \wedge (\neg a \vee \neg b)}\ \text{And}$$

One-sided Sequent calculus (Propositional)

Assume formulas are put in NNF.

$$\frac{}{\Gamma, \phi, \neg\phi \vdash}\ \text{Hyp}$$

$$\frac{\Gamma, \phi, \psi \vdash}{\Gamma, \phi \wedge \psi \vdash}\ \text{And}$$

$$\frac{\Gamma, \phi \vdash \qquad \Gamma, \psi \vdash}{\Gamma, \phi \vee \psi \vdash}\ \text{Or}$$

Example:

$$\dfrac{\dfrac{\dfrac{}{a, b, \neg a}\ \text{Hyp} \qquad \dfrac{}{a, b, \neg b}\ \text{Hyp}}{a, b, \neg a \vee \neg b}\ \text{Or}}{a \wedge b \wedge (\neg a \vee \neg b)}\ \text{And}$$

- Very simple, complete, generalizable
- Very naive, not very efficient

One-sided Sequent calculus (First Order)

Assume formulas are put in NNF.

$$\frac{}{\Gamma, \phi, \neg\phi \vdash}\ \text{Hyp}$$

$$\frac{\Gamma, \phi, \psi \vdash}{\Gamma, \phi \wedge \psi \vdash}\ \text{And}$$

$$\frac{\Gamma, \phi \vdash \qquad \Gamma, \psi \vdash}{\Gamma, \phi \vee \psi \vdash}\ \text{Or}$$

$$\frac{\Gamma, \phi[x := y] \vdash}{\Gamma, \exists x.\phi}\ \text{Exists } (y \text{ a fresh variable})$$

$$\frac{\Gamma, \phi[x := t] \vdash}{\Gamma, \forall x.\phi}\ \text{Forall } (t \text{ an arbitrary term})$$

Instantiation: How to pick the good t

Strategy 1: try all possible terms, in order.
If the formula contains function symbol f^1 and g^2, and free variables x and y...

$$S_1 = \{x, y\}$$

$$S_2 = \{x, y, f(x), f(y) g(x, x), g(x, y), g(y, x), g(y, y)\}$$

$$S_n = S_1 \cup \{f(\_), g(\_, \_)\}[S_{n-1}]$$

$$|S_n| = \mathcal{O}(|S_{n-1}|^2)$$

Completely useless in practice

Instantiation: How to pick the good t

Strategy 2: delay choice of t.
Formally, add rule

$$\frac{\Gamma, \forall x.\phi, \phi[x := t] \vdash}{\Gamma, \forall x.\phi \vdash} \text{ Forall } (t \text{ an arbitrary term})$$

$$\frac{\Gamma, \forall x.\phi, \phi[x := x'] \vdash}{\Gamma, \forall x.\phi \vdash} \text{ Forall } (x' \text{ a metavariable})$$

$$\frac{\text{Apply } \sigma \text{ to quantifiers of } x', y', \ldots}{\Gamma, \phi, \neg\psi \vdash} \text{ Unification (with } \phi \text{ and } \psi \text{ having MGU } \sigma_{x',y'\ldots})$$

Instantiation: How to pick the good t

Problem: when a $\forall$ contains multiple branches. A unification for one branch does not necessarily work for the other, ex:

$$\neg P(a), \neg Q(b) \wedge \forall x. P(x) \vee Q(x)$$

When instantiating metavariables, it must propagate to other branches! Non local, non-functional.

Instantiation: How to pick the good t

Problem: when a $\forall$ contains multiple branches. A unification for one branch does not necessarily work for the other, ex:

$\neg P(a), \neg Q(b) \wedge \forall x. P(x) \vee Q(x)$

When instantiating metavariables, it must propagate to other branches! Non local, non-functional. Alternative: Don't instantiate metavariables, but apply Forall directly.

Instantiation: How to pick the good t

Last problem: When computing unification, ther are often multiple different unifiers that can close the branch. Which to pick?

- Good heuristic necessary for completeness
- Also necessary for reasonable performances in practice
- Sensible strategies: Prioritize unification $\{x_1 := t1, ..., x_n := t_n\}$, with lowest n, lowest height of t_i, where the quantified formulas corresponding to x_i are the smallest, etc.

Extensions

- We dealt with pure first order logic, but techniques exist to extend it to additional theories
-
    - Equality
    - Operations on sets
    - Presburger Arithmetic
- Tableau method is well suited to produce proofs, but tools based on the superposition method are typically better in competitions.
- A lot of complex additional techniques and optimization can be used.