A Generic Tableau Prover and Its Integration with Isabelle

Lawrence C. Paulson

Computer Laboratory

University of Cambridge

Overview of Isabelle

- a generic interactive prover for FOL, set theory, HOL, . . .
- Prolog influence: resolution of generalized Horn clauses

Existing classical reasoner (`Fast_tac`)

- tableau methods
- generic: accepts supplied rules
- runs on Isabelle's Prolog engine (trivial integration)

Objectives for the New Tactic

- Genericity: no restriction to predicate logic
- Power: quantifier duplication, transitivity reasoning . . .
- Speed: perhaps 10–20 seconds for interactive use
- Compatibility with Isabelle's existing tools (`Fast_tac`)

Why Write a New Tableau Prover?

Q. Why not rewrite with $A \subseteq B \iff \forall x\,(x \in A \to x \in B)$?

A. Destroys legibility

A. Not always possible: inductive definitions

Q. Why not just call Otter, SETHEO or LeanTaP?

A. We need higher-order syntax

Typical Generic Tableau Rules

type α	type γ/β	type δ/α
$\dfrac{t \in A \cap B}{\begin{array}{c} t \in A \\ t \in B \end{array}}$	$\dfrac{A \subseteq B}{\neg(?x \in A) \mid ?x \in B}$	$\dfrac{\neg(A \subseteq B)}{\begin{array}{c} \mathbf{s} \in A \\ \neg(\mathbf{s} \in B) \end{array}}$

Complications from genericity:

- overloading — store some type info
- variable instantiation — heuristic limits
- recursive rules — ad-hoc checks

Prover Architecture

Free-variable tableau with iterative deepening (leanTaP)

Term data structure: no types; variables as pointers

Basic heuristics

- discrimination nets
- search-space pruning
- delayed use of unsafe rules (γ-rules)
- suppressing needless duplication

Integration I: Translating Isabelle Rules

- multiple goal formulas via negation
- dual Skolemization $\Rightarrow$ standard Skolemization
- simplification of higher-order conclusions (η-contraction)
- limitations on function variables
- type translation for overloading

Integration II: Translating Tableau Proofs

Isabelle checks the proof—often the slowest phase

- direct correspondence from proof steps to Isabelle tactics
- failure might be caused by
  - breakdown of the correspondence
  - type complications
- recomputation of unifiers
- fancy tricks not possible (e.g. liberalized δ-rule)

Results & Limitations

Good performance on first-order benchmarks e.g. Pelletier's

Mostly compatible with `fast_tac`; can be 10 times faster

- and proves more theorems
- but slower for some 'obvious' problems

Set theory challenge:

$$(\forall x, y \in S\ x \subseteq y) \rightarrow \exists z\, S \subseteq \{z\}$$

Conclusions

- the first tableau prover with higher-order syntax?
- the first tableau prover for ZF, HOL, inductive definitions, . . . ?
- has almost replaced `fast_tac`
- a good example of integration in daily use