Translating a Satallax Refutation to a Tableau Refutation Encoded in Coq

Bachelor Seminar - proposal talk

Andreas Teucke

Advisor: Chad Brown
Supervisor: Gert Smolka

Department of Computer Science
Saarland University

January 7, 2011

Presentation of the Goal

- Higher-order problem given to Satallax
- Satallax normalizes the problem
  and turns it into a sequence of Sat-problems
- Most Sat-solvers don't provide proofs for unsatisfiability
- Goal: Extract a higher-order proof,
  where one can easily check correctness
- Solution: A tableau refutation
  encoded as a Coq Proof Script

Outline

1. Introduction
   - Goal
   - Summary of the first talk
2. Simple Proof
   - Definitions
   - Proof
3. Implementation
   - Search
   - Translation
   - Coq

Satallax

- Satallax is an automated higher-order theorem prover
- It reduces a problem to a sequence of SAT problems
- If the SAT problem is unsatisfiable,
  the HO problem is refutable
- The clauses correspond to rules in the tableau calculus

The Idea

- While showing unsatisfiability,
  Minisat indirectly refutes the problem ...
- ...only using the formulas and tableau steps corresponding to the literals and clauses
- Refuting with this finite tableau calculus terminates and requires no backtracking

Obstacles

- Analytic cut is in some cases required
- The $\exists$ rule can't introduce arbitrary fresh names, but an acyclic relation can assure soundness

Outline

SAARLAND UNIVERSITY

Theorem

If we have an abstract refutation for some problem A
- as a result from Satallax -,
then A is refutable

Definitions

Definition (abstract refutation (F,S))

Let A be an open branch, F a finite set of formulas and S a function from variables to terms.
Then we call (F, S) an abstract refutation of A, if

1. $<_S$ is acyclic
2. For every $x \in dom\ S$, x is not free in A
3. For every full expansion B, either
   B is refutable in $\mathcal{T}$ in one step or
   there is an $x \in dom\ S$ such that $\exists t \in B$ and $\neg[tx] \in B$
   where $t = S(x)$

SAARLAND UNIVERSITY

Definitions

Definition (full expansion)

Open branch A and formula-set F.
B is a full expansion of A,
if $A \subseteq B \subseteq F$, B is open and $\forall s \in F,\ s \in B$ or $B \cup \{s\}$ is closed.

Definition (relation $<_S$)

For a function from variables to terms S,
$<_S$ is the binary relation on variables in *dom* S
where for every $x, y \in dom\ S, x <_S y \Leftrightarrow\ x$ is free in $S(y)$.

Lemma

Lemma

$(F, \emptyset)$ *abstract refutation of* $A \Rightarrow A$ *refutable in* $\mathcal{T}$

Proof.

Induction on distance of A from a full expansion
Base: A is a full expansion $\Rightarrow$ A is refutable in one step.
Step: Apply Cut on some $t \notin A$
and use I.H. on A, t and $A, \neg t$. □

Theorem

Theorem

(F, S) abstract refutation of $A \Rightarrow A$ refutable in $\mathcal{T}$

Proof.

Induction on the size of *dom S*
Base: S is empty $\Rightarrow$ apply Lemma.
Step: Apply Cut and $\exists$ rules on $\exists t$,
where $t = S(x)$ of a $<_S$-minimal x
and use I.H. on $A, \exists t, [tx]$ and $A, \neg\exists t$ with (F, S^{-x}),
where S^{-x} does not contain x. □

Connection between abstract refutation and Satallax

abstract refutation	$\leftrightarrow$	unsatisfiable set of clauses
F	$\leftrightarrow$	set of all literals
S	$\leftrightarrow$	log of existential witnesses
full expansion	$\leftrightarrow$	model
refutation step	$\leftrightarrow$	clause

As every model has at least one unsatisfied clause,
every full expansion is refutable in one step,
where S replaces the freshness condition for the $\exists$ rule.

Outline

SAARLAND UNIVERSITY

3 modules

1. Construction of a refutation for the normalized problem.
2. Translation to a refutation for the original problem.
3. Outputting the refutation encoded as a Coq Proof Script.

Search

Recursive search divided into two parts:

OR-search	AND-search
Input: branch B	Input: branch B, rule t
if B closed then done	apply t on B
else choose a tableau rule t	for every subbranch B'
and call AND-search(B,t)	call OR-search(B')

Start with OR-search(A)

Translation

Satallax rewrites input and normalizes intermediate results:

- Logical constants are reduced to $\bot, \rightarrow, \forall$ and $=$
  e.g. $\exists x.s$ rewritten as $\neg\forall x.\neg s$
- Double negations are removed
- η-reduction
  $\lambda x.f\ x$ normalized to f

Can be applied anywhere in formulas

Translation

1. Problem:
Normalizations have to be translated
into explicit rewrites for Coq.

2. Problem:
The solution should refute the original problem.
Apply matching tableau rules instead of rewriting the problem.

An example

normalized problem

$$\forall x. \neg p\ x\ a$$
$$\forall x. p\ a\ x$$
$$\mathcal{T}_\forall \quad \neg p\ a\ a$$
$$\mathcal{T}_\forall \quad p\ a\ a$$
$$\lightning$$

original problem

$$\neg(\exists x. p\ x\ a)$$
$$\neg(\exists x. \neg p\ a\ x)$$
$$\mathcal{T}_{\neg\exists} \quad \neg p\ a\ a$$
$$\mathcal{T}_{\neg\exists} \quad \neg(\neg p\ a\ a)$$
$$\lightning$$

Proof Script

Definition of special tactics for tableau rules and rewrite

Creating names for bound variables and hypotheses

Upcoming

- Heuristic for choosing tableau rules
- Learning solved refutations of subbranches
- Proof Script module

References I

C. Brown, G. Smolka
"Analytic Tableaux for Simple Type Theory and its First-Order Fragment" (2010).

J. Backes, C. Brown
"Analytic Tableaux for Higher-Order Logic with Choice" (2010)

C. Brown
" Reducing Theorem Proving to a Sequence of SAT Problems" (September 10, 2010)

N. Eén, N. Sörensson
" An Extensible SAT-solver"

References II

F. Pfenning
" Analytic and non-analytic proofs"
In R.E. Shostak, editor, Proceedings of the 7th Conference on Automated Deduction, pages 394-413, Napa, California, May 1984. Springer-Verlag LNCS 170.