Translating a Satallax Refutation to a Tableau Refutation Encoded in Coq

Bachelor Seminar - first talk

Andreas Teucke

Advisor: Chad Brown
Supervisor: Gert Smolka

Department of Computer Science
Saarland University

October 22, 2010

Outline

SAARLAND UNIVERSITY

A Cut-free Tableau Calculus for Simple Type Theory $\mathcal{T}$

- 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)

SAARLAND UNIVERSITY

Some Tableau Rules from $\mathcal{T}$

$$\mathcal{T}_\vee \quad \frac{s \vee t}{s \mid t} \qquad\qquad \mathcal{T}_\wedge \quad \frac{(s \wedge t)}{s, t}$$

$$\mathcal{T}_\forall \quad \frac{\forall x.s}{s^x_y} \quad y \in \mathcal{U} \qquad\qquad \mathcal{T}_\exists \quad \frac{\exists x.s}{s^x_y} \quad y \in \mathcal{V} \text{ fresh}$$

$$\mathcal{T}_{MAT} \quad \frac{\delta s, \neg \delta t}{s \neq t}$$

SAARLAND UNIVERSITY

Automated Theorem Prover Satallax

- Written by Chad E. Brown as a theorem prover using the tableau calculus
- It reduces HO problems to a sequence of SAT problems, which are solved by Minisat
- In case Minisat returns unsatisfiable, the initial problem is refutable
- Chad E. Brown:" Reducing Theorem Proving to a Sequence of SAT Problems" (September 10, 2010)

SAARLAND UNIVERSITY

The Output of Satallax

called a Satallax Refutation

- Returns an unsatisfiable set of clauses C_Σ
- Using PicoSat the set is reduced to its unsatisfiable core
- A clause c is a finite set of formulas $c = \{s_1, \ldots, s_n\}$ thought of disjunctively
- The initial (unit) clauses correspond to the assumptions in the original branch
- All other clauses correspond to rules in the calculus

SAARLAND UNIVERSITY

Step 1: A Finite Tableau Calculus $\mathcal{T}_\Sigma$

- each clause c defines a set F_c of allowed formulas, a set of steps $\mathcal{T}_{c,F}$ restricted on formulas in F and a set Δ_c, that tells, whether a step can be applied
- e.g. for $c = \{\ \overline{s \vee t},\ s,\ t\}$
  $F_c = \{\ s \vee t,\ \neg(s \vee t),\ s,\ t\}$
  $\mathcal{T}_{c,F} = \{< A, A \cup \{s\}, A \cup \{t\} > | \{s \vee t\} \subseteq A \subseteq F\} \subseteq \mathcal{T}_\vee$
  $\Delta_c = \{s \vee t\}$
- $F = \bigcup_{c \in C} F_c$ and $\mathcal{T}_\Sigma = \bigcup_{c \in C} \mathcal{T}_{c,F}$

Step 2: Searching for a Refutation

- Start with only the initial branch in the tableau
- `While` there is an open branch A in the tableau `do`
  - Choose $c \in C_\Sigma$ such that $A \cap c = \emptyset$ (c is not satisfied by A).
  - Such a clause exists, because C_Σ is strongly unsatisfiable.
  - Apply $\mathcal{T}_c$ and replace A by the new branches in the tableau.
- This terminates, because every branch in the tableau would eventually be a C_Σ-branch

SAARLAND UNIVERSITY

Step 2: Searching for a Refutation

- e.g. $c = \{\ \overline{s \vee t},\ s,\ t\}$:
- Case 1 $\{s \vee t\} \subseteq A$: apply $< A, A \cup \{s\}, A \cup \{t\} > \in \mathcal{T}_\Sigma$ add $A \cup \{s\}$ and $A \cup \{t\}$.
- Case 2 $\{s \vee t\} \not\subseteq A$: apply Cut on $s \vee t$ add $A \cup \{s \vee t, s\}$, $A \cup \{s \vee t, t\}$ and $A \cup \{\neg(s \vee t)\}$.

SAARLAND UNIVERSITY

What is Cut and Why We don't Want it

- Cut as a tableau rule $\mathcal{T}_{cut}$ $\frac{}{s \mid \neg s}$ is not in the `cut-free` tableau calculus $\mathcal{T}$
- Therefore we know that there is a refutation without Cut
- Can we always choose c such that we are in case 1?

A Surprising Example

- initial branch $A = \{\{\delta s \vee \delta t\}, \{\neg\delta u \vee \neg\delta t\}, \{s = t\}, \{t = u\}\}$
- could result in $C_\Sigma =$

$$
\begin{array}{l}
\delta s \vee \delta t \\
\neg\delta u \vee \neg\delta t \\
s = t \\
t = u \\
\hline
\end{array}
$$

$$
\begin{array}{lll}
\overline{\delta s \vee \delta t} \sqcup \delta s \sqcup \delta t & & \leftarrow \mathcal{T}_\vee \\
\overline{\neg\delta u \vee \neg\delta t} \sqcup \neg\delta u \sqcup \neg\delta t & & \leftarrow \mathcal{T}_\vee \\
\overline{\delta s} \sqcup \overline{\neg\delta t} \sqcup s \neq t & & \leftarrow \mathcal{T}_{MAT} \\
\overline{\delta t} \sqcup \overline{\neg\delta u} \sqcup t \neq u & & \leftarrow \mathcal{T}_{MAT} .
\end{array}
$$

SAARLAND UNIVERSITY

A Surprising Example

Trying to refute this in $\mathcal{T}_\Sigma$ without Cut ...

$$\begin{array}{c}
\delta s \lor \delta t \\
\neg\delta u \lor \neg\delta t \\
s = t \\
t = u
\end{array}$$

$$\begin{array}{c|c}
\begin{array}{c}
\delta s \\
\begin{array}{c|c}
\neg\delta u & \begin{array}{c} \neg\delta t \\ s \neq t \end{array}
\end{array}
\end{array}
&
\begin{array}{c}
\delta t \\
\begin{array}{c|c}
\begin{array}{c} \neg\delta u \\ t \neq u \end{array} & \neg\delta t
\end{array}
\end{array}
\end{array}$$

... we get stuck.

A Surprising Example

But with a Cut on δt we can complete the refutation.

$$\delta s \lor \delta t$$
$$\neg\delta u \lor \neg\delta t$$
$$s = t$$
$$t = u$$

- δs
  - $\neg\delta u$
    - δt, $t \neq u$
    - $\neg\delta t$, $s \neq t$
  - $\neg\delta t$, $s \neq t$
- δt
  - $\neg\delta u$, $t \neq u$
  - $\neg\delta t$

SAARLAND UNIVERSITY

Making a Compromise

Conclusion :

- $\mathcal{T}_\Sigma$ isn't complete without Cut
- As a solution certain Cuts will be allowed in $\mathcal{T}_\Sigma$
- $\mathcal{T}_\Sigma \cup \{< A, A \cup \{s\}, A \cup \{\neg s\} > | \exists\, c \in C_\Sigma, s \in \Delta_c\}$

Summary

- Search for a refutation in a finite calculus provided by Satallax
- This calculus won't be cut-free

- Outlook
    - Further restricting the use of Cuts
    - Dealing with freshness
      F. Pfenning: "Analytic and non-analytic proofs" (1984).

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"

SAARLAND UNIVERSITY

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.

SAARLAND UNIVERSITY

Some Definitions

Definition

A is a C -branch if $A \subseteq F_C$, A is open, and $\forall c \in C, A \cap c \neq \emptyset$.

Definition

A set of clauses C is strongly unsatisfiable, if there are no C-branches.

SAARLAND UNIVERSITY

Definition rule-clauses

- Definition And-rule : for $c = \{\ \overline{s \wedge t},\ s\} \ \vee\ c = \{\ \overline{s \wedge t},\ t\}$
  $F_c = \{\ \overline{s \wedge t},\ s \wedge t,\ s,\ t\}$, $\Delta_c = \{s \wedge t\}$ and
  $\mathcal{T}_{c,F} = \{< A, A \cup \{s, t\} > | \{s \wedge t\} \subseteq A \subseteq F\} \subseteq \mathcal{T}_\wedge$
- Definition Forall-rule : for $c = \{\ \overline{\forall x.s},\ s^x_y\ \}$
  $F_c = \{\ \overline{\forall x.s},\ , \forall x.s,\ s^x_y\ \}$, $\Delta_c = \{\forall x.s\}$ and
  $\mathcal{T}_{c,F} = \{< A, A \cup \{s^x_y\} > | \{\forall x.s\} \subseteq A \subseteq F\} \subseteq \mathcal{T}_\forall$
- Definition Exists-rule : for $c = \{\ \overline{\exists x.s},\ s^x_y\ \}$
  $F_c = \{\ \overline{\exists x.s},\ \exists x.s,\ s^x_y\ \}$, $\Delta_c = \{\exists x.s\}$ and
  $\mathcal{T}_{c,F} = \{< A, A \cup \{s^x_y\} > | \{\exists x.s\} \subseteq A \subseteq F$
  $\wedge\ y\ is\ fresh\ in\ A\} \subseteq \mathcal{T}_\exists$
  In this case we say c selects y.

SAARLAND UNIVERSITY

A Surprising Example - cut-free refutation

We would have to use $\mathcal{T}_{MAT}$ on δs and $\neg\delta u$ and $\mathcal{T}_{CON}$ on $s = t$ and $s \neq u$ to complete the refutation.

$$\delta s \lor \delta t$$
$$\neg\delta u \lor \neg\delta t$$
$$s = t$$
$$t = u$$
$$\begin{array}{c|c} \delta s & \delta t \\ \begin{array}{c|c} \neg\delta u & \neg\delta t \\ s \neq u & s \neq t \\ \begin{array}{c|c} s \neq s & t \neq u \end{array} & \end{array} & \begin{array}{c|c} \neg\delta u & \neg\delta t \\ t \neq u & \end{array} \end{array}$$

SAARLAND UNIVERSITY

How Freshness Adds to my Troubles

$$\mathcal{T}_\forall \; \frac{\forall x.s}{s^x_y} \; y \in \mathcal{U} \qquad\qquad \mathcal{T}_\exists \; \frac{\exists x.s}{s^x_y} \; y \in \mathcal{V} \text{ fresh}$$

- As the variables are already chosen by Satallax, if we choose a c which selects a variable x, x will need to be still fresh in A
- Therefore which c is chosen has to be restricted

SAARLAND UNIVERSITY

A Solution - The Strict Partial Order $<_C$

Definition

The strict partial order $<_C$ on clauses in C is the transitive closure of $<^0_C$, where $\forall c_1, c_2 \in C$, $c_1 <^0_C c_2 \to \exists$ variable x, c_1 selects $x \wedge x$ is free in c_2.

- The initial list of clauses Satallax produces is a linearisation of $<_{C_\Sigma}$
- The c has to be chosen as a minimum in the set of clauses unsatisfied by A

SAARLAND UNIVERSITY

Another Example

- the initial branch
  $A = \{\ \{\ \forall xy.\neg r\ x\ y\},\ \{(\forall x.r\ x\ x) \vee \exists x.r\ x\ x\}\ \}$
- could result in $C_\Sigma =$
  $\forall xy.\neg r\ x\ y$
  $(\forall x.r\ x\ x) \vee \exists x.r\ x\ x$

$$\overline{(\forall x.r\ x\ x) \vee \exists x.r\ x\ x} \sqcup \forall x.r\ x\ x \sqcup \exists x.r\ x\ x \quad \leftarrow \mathcal{T}_\vee$$

$$\overline{\exists x.r\ x\ x} \sqcup r\ x\ x \quad \leftarrow \mathcal{T}_\exists$$

$$\overline{\forall x.r\ x\ x} \sqcup r\ x\ x \quad \leftarrow \mathcal{T}_\forall$$

$$\overline{\forall xy.\neg r\ x\ y} \sqcup \forall y.\neg r\ x\ y \quad \leftarrow \mathcal{T}_\forall$$

$$\overline{\forall y.\neg r\ x\ y} \sqcup \neg r\ x\ x \quad \leftarrow \mathcal{T}_\forall$$

SAARLAND UNIVERSITY

Another Example

With a Cut on $\exists x. r\ x\ x$ we can again complete the refutation

$$\forall xy. \neg r\ x\ y$$

$$(\forall x. r\ x\ x) \vee \exists x. r\ x\ x$$

$\forall x. r\ x\ x$		$\exists x. r\ x\ x$
$\neg \exists x. r\ x\ x$	$\exists x. r\ x\ x$	$r\ x\ x$
$r\ x\ x$	$r\ x\ x$	$\forall y. \neg r\ x\ y$
$\forall y. \neg r\ x\ y$	$\forall y. \neg r\ x\ y$	$\neg r\ x\ x$
$\neg r\ x\ x$	$\neg r\ x\ x$	

SAARLAND UNIVERSITY

Another Example - desired solution

But we actually would like to have ...

$$
\begin{array}{c}
\forall xy.\neg r\ x\ y \\
(\forall x.r\ x\ x) \vee \exists x.r\ x\ x \\
\begin{array}{c|c}
\forall x.r\ x\ x & \exists x.r\ x\ x \\
r\ x\ x & r\ x\ x \\
\forall y.\neg r\ x\ y & \forall y.\neg r\ x\ y \\
\neg r\ x\ x & \neg r\ x\ x
\end{array}
\end{array}
$$