SeCaV:
A Sequent Calculus Verifier in Isabelle/HOL

Asta Halkjær From, Frederik Krogsdal Jacobsen and Jørgen Villadsen

LSFA 2021

Overview

Introduction

SeCaV

Conclusion

Introduction

- Classical first-order logic is important to learn
- A sequent calculus can be used to teach formal deduction and show proof theory results
- Computer assistance helps students by providing immediate feedback
- We introduce the Sequent Calculus Verifier (SeCaV) as a simple system to support students when learning about first-order logic
- We have used the system in multiple courses

SeCaV
Syntax

$$
\begin{aligned}
tm \quad &::= \quad \text{Fun}\ n\ [tm] \\
&\quad\ \mid \quad \text{Var}\ n \\
fm \quad &::= \quad \text{Pre}\ n\ [tm] \\
&\quad\ \mid \quad \text{Imp}\ fm\ fm \\
&\quad\ \mid \quad \text{Dis}\ fm\ fm \\
&\quad\ \mid \quad \text{Con}\ fm\ fm \\
&\quad\ \mid \quad \text{Exi}\ fm \\
&\quad\ \mid \quad \text{Uni}\ fm \\
&\quad\ \mid \quad \text{Neg}\ fm
\end{aligned}
$$

SeCaV

A simple example - Isabelle/HOL

```
lemma ‹ ⊩
  [
    Dis (Pre 0 [Fun 0 [], Fun 1 []]) (Neg (Pre 0 [Fun 0 [], Fun 1 []]))
  ] ›
proof –
  from AlphaDis have ?thesis if ‹ ⊩
    [
      Pre 0 [Fun 0 [], Fun 1 []],
      Neg (Pre 0 [Fun 0 [], Fun 1 []])
    ] ›
    using that by simp
  with Basic show ?thesis
    by simp
qed
```

SeCaV

A simple example - SeCaV Unshortener

```
Dis p[a, b] (Neg p[a, b])

AlphaDis
  p[a, b]
  Neg p[a, b]
Basic
```

Semantics

- Semantics of connectives and quantifiers are defined by lifting them to the meta-logic of Isabelle/HOL
- Semantics are defined in terms of simple functions
- This allows students to easily understand the semantics

SeCaV

Proof rules I

$$\frac{\text{Neg}\ p \in z}{\Vdash p, z}\ \textsc{Basic}$$

$$\frac{\Vdash z \qquad z \subseteq y}{\Vdash y}\ \textsc{Ext}$$

$$\frac{\Vdash p, z}{\Vdash \text{Neg}\ (\text{Neg}\ p), z}\ \textsc{NegNeg}$$

$$\frac{\Vdash p, q, z}{\Vdash \text{Dis}\ p\ q, z}\ \textsc{AlphaDis}$$

$$\frac{\Vdash \text{Neg}\ p, q, z}{\Vdash \text{Imp}\ p\ q, z}\ \textsc{AlphaImp}$$

$$\frac{\Vdash \text{Neg}\ p, \text{Neg}\ q, z}{\Vdash \text{Neg}\ (\text{Con}\ p\ q), z}\ \textsc{AlphaCon}$$

SeCaV

Proof rules II

$$\frac{\Vdash p, z \qquad \Vdash q, z}{\Vdash \text{Con}\ p\ q, z}\ \textsc{BetaCon} \qquad \frac{\Vdash p, z \qquad \Vdash \text{Neg}\ q, z}{\Vdash \text{Neg}\ (\text{Imp}\ p\ q), z}\ \textsc{BetaImp}$$

$$\frac{\Vdash \text{Neg}\ p, z \qquad \Vdash \text{Neg}\ q, z}{\Vdash \text{Neg}\ (\text{Dis}\ p\ q), z}\ \textsc{BetaDis}$$

SeCaV

Proof rules III

$$\frac{\Vdash p\,[\mathrm{Var}\ 0/t], z}{\Vdash \mathrm{Exi}\ p, z}\ \textsc{GammaExi} \qquad \frac{\Vdash \mathrm{Neg}\ (p\,[\mathrm{Var}\ 0/t]), z}{\Vdash \mathrm{Neg}\ (\mathrm{Uni}\ p), z}\ \textsc{GammaUni}$$

$$\frac{\Vdash p\,[\mathrm{Var}\ 0/\mathrm{Fun}\ i\ []], z \qquad i \text{ fresh}}{\Vdash \mathrm{Uni}\ p, z}\ \textsc{DeltaUni}$$

$$\frac{\Vdash \mathrm{Neg}\ (p\,[\mathrm{Var}\ 0/\mathrm{Fun}\ i\ []]), z \qquad i \text{ fresh}}{\Vdash \mathrm{Neg}\ (\mathrm{Exi}\ p), z}\ \textsc{DeltaExi}$$

SeCaV
Proof rules IV

Substitution

- Variables are referred to using de Bruijn indices
- Substitution is implemented using basic functions – almost no prior experience required
- Each function can be called separately to understand each step
- This makes it easier for students to learn how de Bruijn indices work

Soundness and completeness

Formalized in Isabelle/HOL, which allows us to prove properties of the calculus directly

Soundness

- Simple proof by induction using a lemma about substitution

Completeness

- Based on existing work in the Archive of Formal Proofs
- Proof is by relating our calculus to an existing sequent calculus

SeCaV Unshortener I

- Online system to make writing proofs easier
- Includes warnings for wrong proofs, but proofs must be verified in Isabelle/HOL to be sure
- Available at secav.compute.dtu.dk

SeCaV

SeCaV Unshortener II

Sequent Calculus Verifier | Help and Input Examples | 1:1 | Copy Output to Clipboard | SeCaV Unshortener 1.0

```
(* A shortened proof *)

Imp p p

AlphaImp
  Neg p
  p
Ext
  p
  Neg p
Basic
```

```
proposition ‹p ⟶ p› by metis

text ‹
  Predicate numbers
    0 = p
  ›

lemma ‹⊩
  [
    Imp (Pre 0 []) (Pre 0 [])
  ]
  ›
proof -
  from AlphaImp have ?thesis if ‹⊩
    [
      Neg (Pre 0 []),
      Pre 0 []
    ]
    ›
    using that by simp
  with Ext have ?thesis if ‹⊩
    [
      Pre 0 [],
      Neg (Pre 0 [])
    ]
    ›
    using that by simp
  with Basic show ?thesis
    by simp
qed
```

SeCaV

SeCaV Unshortener III

Sequent Calculus Verifier | Help and Input Examples | 1:1 | Copy Output to Clipboard | SeCaV Unshortener 1.0

```
(* A shortened proof - with a mistake *)

Imp p p

AlphaImp
  Neg p
  p
Basic
```

```
(* Warning:
(Basic) Sequent not a tautology/Positive formula is not the first
*)
proposition ‹p ⟶ p› by metis

text ‹
  Predicate numbers
    0 = p
  ›

lemma ‹⊩
  [
    Imp (Pre 0 []) (Pre 0 [])
  ]
  ›
proof -
  from AlphaImp have ?thesis if ‹⊩
    [
      Neg (Pre 0 []),
      Pre 0 []
    ]
    ›
    using that by simp
  with Basic show ?thesis
    by simp
qed
```

DTU

Conclusion

- The Sequent Calculus Verifier allows students to experiment with formal proofs in a sequent calculus
- Students can understand how the entire system works because everything is implemented with simple functions
- The system can also be used to give students a taste of proofs of soundness and completeness