On the Use of Isabelle/HOL for Formalizing New Concise Axiomatic Systems for Classical Propositional Logic

Asta Halkjær From & Jørgen Villadsen

Technical University of Denmark

TYPES 2021-06-16

Introduction

We have recently (*)
formalized soundness and
completeness theorems
for a number of
axiomatic systems for
classical propositional logic
and build on this work here

(*) Asta Halkjær From, Agnes Moesgård Eschen & Jørgen Villadsen
Conference on Intelligent Computer Mathematics (CICM 2021)

Wajsberg 1937	$p \to (q \to p)$ $(p \to q) \to (q \to r) \to (p \to r)$ $((p \to q) \to p) \to p)$ $\bot \to p$
Wajsberg 1939	$p \to (q \to p)$ $(p \to (q \to r)) \to (p \to q) \to (p \to r)$ $((p \to \bot) \to \bot) \to p)$
Łukasiewicz 1948	$((p \to q) \to r) \to ((r \to p) \to (s \to p))$ $\bot \to p$

Tarski showed in 1925 that the pure implicational calculus could be based on a single axiom, but a series of improvements by Wajsberg and Łukasiewicz led to the latter's discovering in 1936 that the formula CCCpqrCCrpCsp could serve as single axiom and that no shorter axiom would suffice, though the publication of this result had to wait until 1948.

Main Goal

From a teaching perspective, the proof assistant Isabelle/HOL provides an environment for students where they can confidently explore variations of the axiomatic systems.

We describe novel axiomatic systems for classical propositional logic: one based on the K and S combinators and elimination rules and one on transitivity of implication, explosion and rules for disjunction.

We show how Isabelle/HOL helps investigate such systems.

Our present work can be extended in several research directions.

We benefit from the simple types and the automation in Isabelle/HOL.

Specifying a Language

Encode syntax as objects in the metalogic

```
datatype form = Falsity (‹⊥›) | Pro nat | Imp form form (infix ‹→› 0)
```

Give meaning by interpreting it into the metalogic

```
primrec semantics (infix ‹⊨› 0) where
  ‹(I ⊨ ⊥) = False› |
  ‹(I ⊨ (Pro n)) = I n› |
  ‹(I ⊨ (p → q)) = (if I ⊨ p then I ⊨ q else True)›
```

Specifying a Proof System

Inductive predicate ⊢ built from:

- One rule (modus ponens)
- Four axiom schemas

```
inductive Axiomatics (‹⊢›) where
  ‹⊢ q› if ‹⊢ p› and ‹⊢ (p → q)› |
  ‹⊢ (p → (q → p))› |
  ‹⊢ ((p → q) → ((q → r) → (p → r)))› |
  ‹⊢ (((p → q) → p) → p)› |
  ‹⊢ (⊥ → p)›
```

Syntactic abbreviations unfold automatically

```
abbreviation Truth (‹⊤›) where ‹⊤ ≡ (⊥ → ⊥)›
```

The . proof method applies current facts as rules

```
theorem ‹⊢ ⊤› using Axiomatics.intros(5) .
```

Soundness and Completeness

Enabled by formalizing both proof system and semantics

Soundness

```
theorem soundness: ‹⊢ p ⟹ I ⊨ p›
  by (induct rule: Axiomatics.induct) auto
```

Completeness

A few hundred lines of proof (so-called Henkin style)

Deriving certain formulas in classical propositional logic is key

Classical Propositional Logic – New Axioms

The TYPES1 Theory

The first two axioms correspond to the combinators K and S (involving only implication), and the second two axioms eliminate the negation and disjunction operators, respectively.

inductive H :: ‹$form \Rightarrow bool$› (‹$\vdash$ -› [50] 50) **where**
  ‹$\vdash p \Longrightarrow \vdash (p \longrightarrow q) \Longrightarrow \vdash q$› |
  ‹$\vdash (p \longrightarrow q \longrightarrow p)$› |
  ‹$\vdash ((p \longrightarrow q \longrightarrow r) \longrightarrow (p \longrightarrow q) \longrightarrow p \longrightarrow r)$› |
  ‹$\vdash (p \longrightarrow \neg\, p \longrightarrow q)$› |
  ‹$\vdash ((p \longrightarrow r) \longrightarrow (q \longrightarrow r) \longrightarrow p \vee q \longrightarrow r)$›

The last axiom is classical since disjunction is an abbreviation: $p \vee q \equiv \neg\, p \longrightarrow q$. With Isabelle/HOL we quickly prove other formulas derivable (also using Sledgehammer):

proposition $*$: ‹$\vdash (\neg\,\neg\, p \longrightarrow p)$›
  by (*metis H.intros*)

This formula is the key lemma in the completeness proof for the axiomatic system TYPES1.

Takeaways

- Experiment with derivations at a high level
    - Quickly derive a range of formulas
    - Or mark them with sorry
- Let the proof assistant handle the details
    - Finding relevant axioms
    - Instantiating rules
- Every definition is given in precise language
- Every result is mechanically checked
- Verify and build on historical results

Full List of Isabelle/HOL Theories

System_W	667 lines	Browse	Download
System_R	629 lines	Browse	Download
W0	23 lines	Browse	Download
TYPES1	30 lines	Browse	Download
TYPES2	34 lines	Browse	Download

View generated theory dependencies
View generated document

https://people.compute.dtu.dk/ahfrom/types2021/

Conclusions

Our work is part of the IsaFoL project, Isabelle Formalization of Logic, aiming to support modern research in automated reasoning as well as courses using proof assistants

Future Work – Modal and First-Order Logics

Formalized Soundness and Completeness of Epistemic Logic (Extended Abstract)
Asta Halkjær From, Alexander Birch Jensen & Jørgen Villadsen
Workshop on Logical Aspects in Multi-Agent Systems and Strategic Reasoning 2021

Programming and Verifying a Declarative First-Order Prover in Isabelle/HOL
Alexander Birch Jensen, John Bruntse Larsen, Anders Schlichtkrull & Jørgen Villadsen
AI Communications 31:281-299 2018