Formalizing Axiomatic Systems for Propositional Logic in Isabelle/HOL

Asta Halkjær From, Agnes Moesgård Eschen, Jørgen Villadsen
Technical University of Denmark

CICM-14 2021-07-28

Isabelle/HOL

Proof assistant based on higher-order logic

Every proof is checked mechanically

Create *canonical reference documents* for:

- logics
- metatheory
- algorithms
- ...

Our Work

Formalize six axiomatic proof systems for propositional logic
Two flavours with different primitives:

- `System_W.thy` based on $\bot, \rightarrow$
- `System_R.thy` based on $\neg, \vee$

https://github.com/logic-tools/axiom

- Soundness and completeness
- Alternative and unnecessary axioms
- *Deriving formulas*

System	Source	Page [3]	Axioms
Axiomatics	Wajsberg 1937	159	$p \to (q \to p)$ $(p \to q) \to (q \to r) \to (p \to r)$ $((p \to q) \to p) \to p)$ $\perp \to p$
FW	Wajsberg 1939	163	$p \to (q \to p)$ $(p \to (q \to r)) \to (p \to q) \to (p \to r)$ $((p \to \perp) \to \perp) \to p)$
WL	Łukasiewicz 1948	159	$((p \to q) \to r) \to ((r \to p) \to (s \to p))$ $\perp \to p$
Axiomatics	Rasiowa 1949	157	$p \bigvee p \to p$ $p \to p \bigvee q$ $(p \to q) \to (r \bigvee p) \to (q \bigvee r)$
RB	Russell 1908, Bernays 1926	157	$p \bigvee p \to p$ $p \to p \bigvee q$ $p \bigvee q \to q \bigvee p$ $(p \to q) \to (r \bigvee p) \to (q \bigvee r)$
PM	Whitehead & Russell 1910	-	$p \bigvee p \to p$ $p \to q \bigvee p$ $p \bigvee q \to q \bigvee p$ $(p \bigvee (q \bigvee r)) \to (q \bigvee (p \bigvee r))$ $(p \to q) \to (r \bigvee p) \to (q \bigvee r)$

Specifying a Language

Encode syntax as objects in the metalogic

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

Interpret 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)›
```

Example use

```
definition ‹valid p ≡ ∀I. (I ⊨ p)›
```

Specifying a Proof System

Inductive predicate ⊢ built from:

- One rule (modus ponens)
- Four axiom schemas

```
     inductive Axiomatics (‹⊢›) where
MP   ‹⊢ q› if ‹⊢ p› and ‹⊢ (p → q)› |
Imp1 ‹⊢ (p → (q → p))› |
Tran ‹⊢ ((p → q) → ((q → r) → (p → r)))› |
Clas ‹⊢ (((p → q) → p) → p)› |
Expl ‹⊢ (⊥ → p)›
```

Wajsberg 1937

The Simplest of Derivations

Consider the following syntactic abbreviation

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

These unfold automatically

The `.` proof method applies current facts as rules

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

Here we are *using* the fact $\vdash (\bot \longrightarrow ?p)$, which unifies with the goal

Soundness

Important smoke test but tedious to check manually

Isabelle discharges all proof obligations automatically

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

Enabled by formalizing both proof system and semantics

Completeness

We build on existing work [F. 2020]

Henkin-style proof based on maximal consistent sets

Model construction uses certain derivations

Two main tasks:

- Adapt formalization to same syntactic fragment
  - Easy to do with abbreviations
- Derive key formulas
  - Difficulty depends on proof system

Deriving Formulas I

Church has been an excellent source for relevant formulas

Let Automatic Theorem Provers and SMT solvers do the work

```
lemma Chu1: ‹⊢ ((p → (p → q)) → (p → q))›
sledgehammer
```

```
Sledgehammering...
Proof found...
"e": Try this: by (meson Clas MP Tran) (3 ms)
"vampire": Try this: by (meson Axiomatics.simps) (39 ms)
"cvc4": The prover gave up
"z3": Timed out
```

Deriving Formulas II

What's in a proof?

```
lemma Chu1: ‹⊢ ((p → (p → q)) → (p → q))›
  by (meson Tran Clas MP)
```

Tran and Clas are two axioms of the proof system — MP is modus ponens

"meson implements Loveland's model elimination procedure" [isar-ref]

meson proves *the existence* of a derivation for us

Łukasiewicz's Shortest Axiom I

A single axiom for the implicational propositional calculus

```
inductive WL (‹≫›) where
  ‹≫ q› if ‹≫ p› and ‹≫ (p → q)› |
  ‹≫ (((p → q) → r) → ((r → p) → (s → p)))› |
  ‹≫ (⊥ → p)›
```

We can reuse Łukasiewicz's notation

```
abbreviation (input) C :: ‹form ⇒ form ⇒ form› (‹C _ _› [0, 0] 1) where
  ‹(C p q) ≡ (p → q)›
```

We formalize his derivation of the Wajsberg axioms

Łukasiewicz’s Shortest Axiom II

1 $C\,C\,C\,p\,q\,r\,C\,C\,r\,p\,C\,s\,p.$

$1\ p/C\,p\,q,\ q/r,\ r/C\,C\,r\,p\,C\,s\,p,\ s/r + C\,1 - 2.$

2 $C\,C\,C\,C\,r\,p\,C\,s\,p\,C\,p\,q\,C\,r\,C\,p\,q.$

$1\ p/C\,C\,r\,p\,C\,s\,p,\quad q/C\,p\,q,\quad r/C\,r\,C\,p\,q,\quad s/t + C\,2 - 3.$

3 $C\,C\,C\,r\,C\,p\,q\,C\,C\,r\,p\,C\,s\,p\,C\,t\,C\,C\,r\,p\,C\,s\,p.$

```
lemma l1: ‹⊫ (C C C p q r C C r p C s p)›
  using WL.intros(2) .

lemma l2: ‹⊫ (C C C C r p C s p C p q C r C p q)›
  using l1 by (meson WL.intros(1))

lemma l3: ‹⊫ (C C C r C p q C C r p C s p C t C C r p C s p)›
  using l1 l2 by (meson WL.intros(1))
```

Łukasiewicz's Shortest Axiom III

We have formalized the 29 lines directly as given by Łukasiewicz
Isabelle/HOL handles the instantiations

Easy to then show equivalence with the Wajsberg axioms

```
theorem equivalence: ‹≫ p ⟷ ⊢ p›
proof
  have *: ‹⊢ (((p → q) → r) → ((r → p) → (s → p)))› for p q r s
    using completeness by simp
  show ‹⊢ p› if ‹≫ p›
    using that by induct (auto simp: * intro: Axiomatics.intros)
  show ‹≫ p› if ‹⊢ p›
    using that by induct (auto simp: l27 l28 l29 intro: WL.intros)
qed
```

System	Source	Page [3]	Axioms
Axiomatics	Wajsberg 1937	159	$p \to (q \to p)$ $(p \to q) \to (q \to r) \to (p \to r)$ $((p \to q) \to p) \to p)$ $\bot \to p$
FW	Wajsberg 1939	163	$p \to (q \to p)$ $(p \to (q \to r)) \to (p \to q) \to (p \to r)$ $((p \to \bot) \to \bot) \to p)$
WL	Łukasiewicz 1948	159	$((p \to q) \to r) \to ((r \to p) \to (s \to p))$ $\bot \to p$
Axiomatics	Rasiowa 1949	157	$p \bigvee p \to p$ $p \to p \bigvee q$ $(p \to q) \to (r \bigvee p) \to (q \bigvee r)$
RB	Russell 1908, Bernays 1926	157	$p \bigvee p \to p$ $p \to p \bigvee q$ $p \bigvee q \to q \bigvee p$ $(p \to q) \to (r \bigvee p) \to (q \bigvee r)$
PM	Whitehead & Russell 1910	-	$p \bigvee p \to p$ $p \to q \bigvee p$ $p \bigvee q \to q \bigvee p$ $(p \bigvee (q \bigvee r)) \to (q \bigvee (p \bigvee r))$ $(p \to q) \to (r \bigvee p) \to (q \bigvee r)$

Prototyping I

Consider the fragment ¬, ⋁

```
datatype form = Pro nat | Neg form | Dis form form (infix ‹⋁› 0)

abbreviation Imp (infix ‹⟶› 0) where ‹(p ⟶ q) ≡ (Neg p ⋁ q)›
```

Rasiowa's axioms are natural for these primitives

```
      inductive Axiomatics (‹⊢›) where
MP     ‹⊢ q› if ‹⊢ p› and ‹⊢ (p ⟶ q)› |
Idem   ‹⊢ ((p ⋁ p) ⟶ p)› |
AddR   ‹⊢ (p ⟶ (p ⋁ q))› |
Swap   ‹⊢ ((p ⟶ q) ⟶ ((r ⋁ p) ⟶ (q ⋁ r)))›
```

Prototyping II

However, unclear how to derive a formula like this where *sledgehammer* fails

```
lemma Tran: ‹⊢ ((p → q) → ((q → r) → (p → r)))›
```

- Start deriving it manually?
- Try to imagine what formulas might help?

In either case, we can admit *stepping stones* to be derived later

```
lemma SwapAnte: ‹⊢ (((p ⋁ q) → r) → ((q ⋁ p) → r))›
sorry
```

Prototyping III

```
lemma Chu1: ‹⊢ ((p → (p → q)) → (p → q))›
  by (meson Tran Clas MP)

lemma Chu2: ‹⊢ (p → ((p → q) → q))›
  by (meson Chu1 Imp1 Tran MP)

lemma Chu3: ‹⊢ ((p → (q → r)) → (q → (p → r)))›
  by (meson Chu2 Tran MP)

lemma Chu4: ‹⊢ ((q → r) → ((p → q) → (p → r)))›
  by (meson Chu3 Tran MP)

lemma Imp2: ‹⊢ ((p → (q → r)) → ((p → q) → (p → r)))›
  by (meson Chu4 Chu1 Chu3 MP)

lemma Imp3: ‹⊢ (p → p)›
  by (meson Imp1 Tran Clas MP)

lemma Neg: ‹⊢ (((p → ⊥) → ⊥) → p)›
  by (meson Chu4 Clas Expl MP)
```

Alternative Axioms

Can we swap out axiom AddR for an AddL that weakens on the left instead?

```
proposition alternative_axiom: ‹⊢ (p → (p ⋁ q))› if ‹⋀p q. ⊢ (p → (q ⋁ p))›
  by (metis MP Idem Swap that)
```

Yes! We can derive AddR from *that*

We can also derive AddL using AddR

```
lemma AddL: ‹⊢ (p → (q ⋁ p))›
  by (metis MP Idem Swap AddR)
```

Unnecessary Axioms I

Russell 1908,
Bernays 1926

```
inductive RB (‹⊩›) where
  ‹⊩ q› if ‹⊩ p› and ‹⊩ (p → q)› |
  ‹⊩ ((p ⋁ p) → p)› |
  ‹⊩ (p → (q ⋁ p))› |
  ‹⊩ ((p ⋁ q) → (q ⋁ p))› |
  ‹⊩ ((p → q) → ((r ⋁ p) → (r ⋁ q)))›
```

Principia Mathematica
Whitehead and Russell 1910

```
inductive PM (‹≫›) where
  ‹≫ q› if ‹≫ p› and ‹≫ (p → q)› |
  ‹≫ ((p ⋁ p) → p)› |
  ‹≫ (p → (q ⋁ p))› |
  ‹≫ ((p ⋁ q) → (q ⋁ p))› |
  ‹≫ ((p ⋁ (q ⋁ r)) → (q ⋁ (p ⋁ r)))› |
  ‹≫ ((p → q) → ((r ⋁ p) → (r ⋁ q)))›
```

Unnecessary Axioms II

Easy to show that PM extends RB

```
proposition PM_extends_RB: ‹⊩ p ⟹ ≫ p›
  by (induct rule: RB.induct) (auto intro: PM.intros)
```

RB is equivalent to Rasiowa's complete axioms

```
theorem Axiomatics_RB: ‹⊢ p ⟷ ⊩ p›
proof
  show ‹⊢ p› if ‹⊩ p›
    using that by induct (use SubR Axiomatics.intros in meson)+
  show ‹⊩ p› if ‹⊢ p›
    using that by induct (use RB.intros in meson)+
qed
```

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

Abridged Bibliography

Łukasiewicz, J.: The shortest axiom of the implicational calculus of propositions. Proc. Royal Irish Acad. Sect. A: Math. Phys. Sci. 52, 25–33 (1948)

Church, A.: Introduction to Mathematical Logic. Princeton University Press, Princeton (1956)

Michaelis, J., Nipkow, T.: Formalized proof systems for propositional logic. TYPES 2017. LIPIcs, vol. 104 (2017), pp. 5:1–5:16.

From, A.H.: Formalizing Henkin-style completeness of an axiomatic system for propositional logic. ESSLLI Virtual Student Session (2020), pp. 1–12.

Wenzel, M.: The Isabelle/Isar Reference Manual. Part of the Isabelle distribution (2021).

Nipkow, T. et al. "Functional Algorithms, Verified!" (2021). https://functional-algorithms-verified.org/