

Towards Formally Verified Theorem Provers

Ramana Kumar

Computer Laboratory, University of Cambridge

Twenty Years of the QED Manifesto
Vienna Summer of Logic, 2014

HOL Verified

Goals of the Project

An implementation of HOL Light

1. that runs existing developments (e.g. Flyspeck),
2. whose soundness, from the semantics of HOL to the semantics of the processor running the machine-code, is verified.
3. And which runs the verification of its own soundness!

Achievements to date

1. Implementation of HOL Light kernel in CakeML proved sound against HOL semantics.
2. Verified compiler for CakeML proved sound against machine-code semantics.

HOL Verified

Goals of the Project

An implementation of HOL Light

1. that runs existing developments (e.g. Flyspeck),
2. whose soundness, from the semantics of HOL to the semantics of the processor running the machine-code, is verified.
3. And which runs the verification of its own soundness!

Achievements to date

1. Implementation of HOL Light kernel in CakeML proved sound against HOL semantics.
2. Verified compiler for CakeML proved sound against machine-code semantics.

How did we get here?

Key Ideas

1. Simple, well-understood, but expressive logic.
2. LCF architecture, supporting substantial packages.
 - ▶ inductive datatypes,
 - ▶ recursive functions,
 - ▶ inductive relations,
 - ▶ rewriting,
 - ▶ **custom automation:**
3. Certifying translations between shallow and deep embeddings.
4. Bootstrapping by evaluation in the logic.

A reflection principle for HOL?

Compared to Milawa

- ▶ Larger gap between underlying logic (HOL) and executable model (CakeML)
- ▶ No explicit proofs: uses ephemeral proofs represented by the protected type of theorems

But it probably could be done!

Compared to Coq

- ▶ Computational reflection rule for HOL?

Worth investigation.

Tiling Trust (Very Preliminary!)

Another Kind of Reflection

- ▶ Certifying translation from HOL terms to deeply-embedded HOL terms.
- ▶ Ultimately, for each ϕ :

$$\begin{aligned} [] \vdash \forall n. \text{Pr}(\text{"}\forall l. \text{LCA}(l) \implies \phi(l, n)\text{"}) \\ \implies \forall l. \text{LCA}(l + 1) \implies \phi(l, n) \end{aligned}$$

- ▶ Instantiate $\phi(l, n)$ with “The l th printed theorem by prover with code n is true”.

Tiling Trust (Very Preliminary!)

More reflection than Milawa

- ▶ Spec for Milawa: “Every printed theorem is true (according to the semantics of Milawa logic)”
- ▶ Milawa method: Prove “Every printed theorem has a proof in the initial Milawa inference system”
- ▶ Less restrictive: Prove “Every printed theorem is true” directly, allowing extensions to the inference system.

Is Self-Verification Useful?

Run a Prover on a Proof of its soundness.

Suppose the Prover accepts the proof. What could this mean?

	Prover Sound	Prover Not Sound
Proof Valid	Good!	Impossible 1
Proof Not Valid	Impossible 2	Danger

Impossible 1 Valid proof that prover is sound, but prover not sound. (Modelling problem.)

Impossible 2 Prover sound, but accepts an invalid proof.

Danger A real possibility where we have learned nothing.

Is Self-Verification Useful?

Run a Prover on a Proof of its soundness.

Suppose the Prover accepts the proof. What could this mean?

	Prover Sound	Prover Not Sound
Proof Valid	Good!	Impossible 1
Proof Not Valid	Impossible 2	Danger

Impossible 1 Valid proof that prover is sound, but prover not sound. (Modelling problem.)

Impossible 2 Prover sound, but accepts an invalid proof.

Danger A real possibility where we have learned nothing.

Is Self-Verification Useful?

Run a Prover on a Proof of its soundness.

Suppose the Prover accepts the proof. What could this mean?

	Prover Sound	Prover Not Sound
Proof Valid	Good!	Impossible 1
Proof Not Valid	Impossible 2	Danger

Impossible 1 Valid proof that prover is sound, but prover not sound. (Modelling problem.)

Impossible 2 Prover sound, but accepts an invalid proof.

Danger A real possibility where we have learned nothing.

Need to Trust Something

What is missing is independent evidence that either the proof is valid or the prover is sound. But if we had either of those, why bother with self-verification?

Formal Proof as Evidence

- ▶ Ultimately verification is just one kind of evidence affecting the probability that something is true.
- ▶ Self-verification is a stress test for a theorem prover in multiple ways.

Need to Trust Something

What is missing is independent evidence that either the proof is valid or the prover is sound. But if we had either of those, why bother with self-verification?

Formal Proof as Evidence

- ▶ Ultimately verification is just one kind of evidence affecting the probability that something is true.
- ▶ Self-verification is a stress test for a theorem prover in multiple ways.

Relevance to QED?

- ▶ Interoperability is still the main obstacle to QED.
- ▶ Applying formal methods to our formal methods to gain some control?