

Mechanizing the Metatheory of Sledgehammer

Jasmin Blanchette and Andrei Popescu

Fakultät für Informatik
Technische Universität München

Overview

- Sledgehammer
- Sledgehammer's Metatheory
- Mechanization

Overview

- Sledgehammer
- Sledgehammer's Metatheory
- Formal Verification

Overview

- Sledgehammer
- Sledgehammer's Metatheory
- Formal Verification

Sledgehammer

- Isabelle: **interactive** theorem prover
- Sledgehammer: connects Isabelle with **automatic** theorem provers (ATPs)

Lemma: $\forall x. \exists y. P(x, y)$

Proof (with Isabelle's structured proof language):

Fix x

Obtain z where $Q(x, z)$ using $\langle \text{facts} \rangle$ by $\langle \text{method} \rangle$

From **this** obtain y where $P(x, y)$ **call sledgehammer**

qed

Sledgehammer's Pipeline

φ

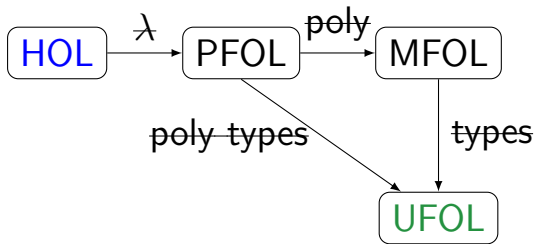
- Start with (typed) HOL goal φ
- Gather set of relevant facts F
- Encode in (untyped) FOL
 λ 's go away, types go away
- Feed to ATPs
- If one succeeds, with $E \subseteq F$
- Reprove with Isabelle's internal ATP, Metis
- Why reprove? Isabelle trusts no one but herself

Overview

- Sledgehammer
- Sledgehammer's Metatheory
- Formal Verification

Sledgehammer's Metatheory

Higher Order Logic



Are the encodings sound? complete? efficient?
All these are important practical questions!

Sledgehammer's Metatheory

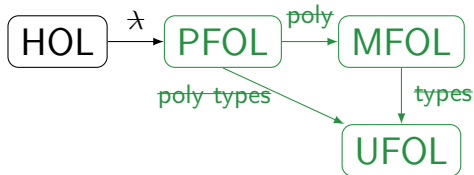
- **Completeness:** The original Isabelle goal provable implies FOL encoding also provable
↳ The ATPs have a chance to prove it
- **Soundness:** FOL encoding provable implies original Isabelle goal also provable
↳ Isabelle can reconstruct the proof
- **Efficiency**
 - When erasing types, extra structure is added in compensation: tag or guard decoration
 - Decoration should be lightweight, yet preserve soundness and completeness
↳ Smaller clutter means better chance for ATPs

It's all for the user 😊

Overview

- Sledgehammer
- Sledgehammer's Metatheory
- Formal Verification

Formal Verification



Blanchette et al., TACAS 2013:

Encoding Monomorphic and Polymorphic Types
(Soundly, Completely, and Efficiently)

A variety of encodings

- based on tags or guards
- full, lightweight, featherweight
- factor in infiniteness knowledge

FroCoS 2013: [Formal verification by Isabelle herself](#)

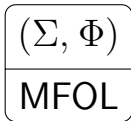
Why Formal Verification

- Blanchette has finetuned the encodings making them as clutter-free as possible
He can sleep better: they are still sound!
- General-purpose formalization of MFOL results: completeness, clausification, Löw-Skolem
- Test/illustrate new Isabelle features
 - Theory of syntax with bindings
 - Codatatypes

Encodings

Σ : MFOL signature

Φ : set of Σ -formulas



M

$N^\$$



$M^\#$

N

Encodings

Example unary guard predicates for each erased sort

(Σ, Φ)
MFOL

$(\Sigma^\#, \Phi^\#)$
UFOL

$\Sigma : (\{\text{elem}, \text{list}\}, \{\text{nil}, \text{cons}\}, \emptyset)$

$\Phi : \text{nil} \neq \text{cons}(x, xs)$

$\Sigma^\# : (\{\text{nil}, \text{cons}\}, \{g_{\text{elem}}, g_{\text{list}}\})$

$\text{Ax}_\Phi : g_{\text{list}}(\text{nil}), g_{\text{list}}(\text{cons}(x, xs))$

$\Phi^\# : \text{Ax}_\Phi$ together with

$g_{\text{elem}}(x) \wedge g_{\text{list}}(xs) \implies \text{nil} \neq \text{cons}(x, xs)$

Encodings

Example unary guard predicates for each erased sort

(Σ, Φ)
MFOL

$(\Sigma^\#, \Phi^\#)$
UFOL

$\Sigma : (\{\text{elem}, \text{list}\}, \{\text{nil}, \text{cons}\}, \emptyset)$

$\Phi : \text{nil} \neq \text{cons}(x, xs)$

$\Sigma^\# : (\{\text{nil}, \text{cons}\}, \{g_{\text{elem}}, g_{\text{list}}\})$

$Ax_\Phi : \cancel{g_{\text{list}}(\text{nil})} \wedge \cancel{g_{\text{list}}(\text{cons}(x, xs))}$

$\Phi^\# : Ax_\Phi$ together with

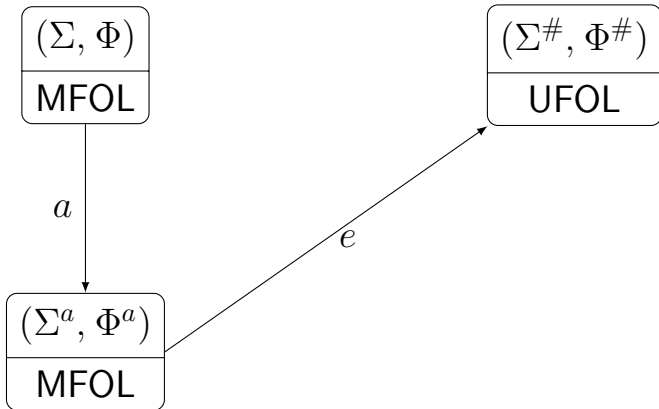
$\cancel{g_{\text{elem}}(x)} \wedge \cancel{g_{\text{list}}(xs)} \implies \text{nil} \neq \text{cons}(x, xs)$

Encodings

Methodology: add decorations first, erase second

$$\# = e \circ a$$

Sufficient: make Φ^a monotonic



Monotonicity

- Claessen et al., CADE 2011:
Sort It Out with Monotonicity
- Φ monotonic: If Φ has a model, then it also has a larger model with all type interpretations of equal size
 \mapsto types in Φ can be safely erased
- $\text{nil} \neq \text{cons}(x, xs)$ already monotonic
- Syntactic criteria

Our Formalization

- MFOL and UFOL
 - basics: formulas, models, satisfaction, substitution
 - classic results: completeness, compactness, Löwenheim-Skolem, clausification
- Type encodings from Blanchette et al.
- Monotonicity criteria from Claessen et al.
- Soundness and completeness

Our Formalization

Locales: Fix types, constants and assumptions

Problems = Locales Encodings = Morphisms

$\text{MFOL_Prob} <_a \text{MFOL_Mono_Prob} <_e \text{UFOL_Prob}$

- MFOL and UFOL
 - basics: formulas, models, satisfaction, substitution
 - classic results: completeness, compactness, Löwenheim-Skolem, classification
- **Type encodings** from Blanchette et al.
- Monotonicity criteria from Claessen et al.
- **Soundness and completeness**

Our Formalization

Can focus on at most countable models

Use a fixed countable HOL type

Monotonicity analysis becomes manageable

- MFOL and UFOL
 - basics: formulas, models, satisfaction, substitution
 - classic results: completeness, compactness, Löwenheim-Skolem, clausification
- Type encodings from Blanchette et al.
- Monotonicity criteria from Claessen et al.
- Soundness and completeness

Our Formalization

- MFOL and UFOL
 - basics: formulas, models, satisfaction, substitution
 - classic results: completeness, compactness, Löwenheim-Skolem, clausification
- Type encodings from Blanchette et al.
- Monotonicity criteria from Claessen et al.
- Soundness and completeness

Our Formalization

- MFOL and UFOL
 - **bureaucracy**
 - classic results: completeness, compactness, Löwenheim-Skolem, clausification
- Type encodings from Blanchette et al.
- Monotonicity criteria from Claessen et al.
- Soundness and completeness

Our Formalization

New Isabelle theory of syntax with bindings

- MFOL and UFOL
 - **eaucracy**
 - classic results: completeness, compactness, Löwenheim-Skolem, clausification
- Type encodings from Blanchette et al.
- Monotonicity criteria from Claessen et al.
- Soundness and completeness

Our Formalization

New Isabelle theory of syntax with bindings
Recursors that interact well with substitution

- MFOL and UFOL
 - **acy**
 - classic results: completeness, compactness, Löwenheim-Skolem, clausification
- Type encodings from Blanchette et al.
- Monotonicity criteria from Claessen et al.
- Soundness and completeness

Our Formalization

New Isabelle theory of syntax with bindings
Recursors that interact well with substitution
Optimized for semantic interpretations of syntax

- MFOL and UFOL
 - classic results: completeness, compactness, Löwenheim-Skolem, classification
- Type encodings from Blanchette et al.
- Monotonicity criteria from Claessen et al.
- Soundness and completeness

Our Formalization

Formalized tableau-based proof a la Beth/Hintikka
Infinite proof trees captured as a codatatype
Used Isabelle's codatatype package

- MFOL and UFOL
 - basics: formulas, models, satisfaction, substitution
 - classic results: **completeness**, compactness, Löwenheim-Skolem, clausification
- Type encodings from Blanchette et al.
- Monotonicity criteria from Claessen et al.
- Soundness and completeness

Conclusions

- Have put Isabelle to work in order to verify its own communication with ATPs
- Based on this, Isabelle might learn (how) to trust the ATPs

Mechanizing the Metatheory of Sledgehammer

Jasmin Blanchette and Andrei Popescu

Fakultät für Informatik
Technische Universität München