Translating Higher-Order to Higher-Order

Chad E. Brown Thibault Gauthier Josef Urban

Czech Technical University in Prague

April 2019

Outline

Introduction

- ATPs are for different logics

Introduction

- ATPs are for different logics
  - First-Order
  - Higher-Order (with choice)

Introduction

- ATPs are for different logics
  - First-Order
    - FOF: untyped/one type
    - TF0: many types
    - TF1: many types with type variables (polymorphism)
  - Higher-Order (with choice)
    - TH0: function types but no type variables
    - TH1: function types and type variables

Introduction

- ATPs are for different logics
  - First-Order
    - FOF: untyped/one type
    - TF0: many types
    - TF1: many types with type variables (polymorphism)
  - Higher-Order (with choice)
    - TH0: function types but no type variables
    - TH1: function types and type variables
- HOL4 (ITP): Higher-Order + choice + infinity + polymorphism + type definitions

Introduction

- Goal: Translate HOL4 library into ATP problems with a variety of ATP representations.
- Proposed Competition allows FO ATPs (e.g., E) to compete against HO ATPs (e.g., LEO-III)
- GRUNGE

Grand **Un**ified Lar**ge** Theory Benchmarks

Introduction

- Goal: Translate HOL4 library into ATP problems with a variety of ATP representations.
- Proposed Competition allows FO ATPs (e.g., E) to compete against HO ATPs (e.g., LEO-III)
- GRUNGE

  Grand **Un**ified Lar**ge** Theory Benchmarks
- Today:
- HOL4 Logic and Library
- Families of Translations
- Example and Preliminary Results

Outline

HOL4 Logic (Types)

- o propositions/booleans
- ι infinite base type
- $\sigma \to \tau$ function types
- $\delta(\sigma_1 \cdots \sigma_n)$ defined types, e.g., real or list σ.
  defined by giving a provably nonempty predicate over an existing type

Types are nonempty.

HOL4 Logic (Terms)

- x variable
- $c(\sigma_1, \ldots, \sigma_n)$ primitive or defined constant
  - Primitive Choice(σ) of type $(\sigma \to o) \to \sigma$
  - Primitive Forall(σ) of type $(\sigma \to o) \to o$
  - Defined Exists(σ) of type $(\sigma \to o) \to o$
- $(s\ t)$ application
- $(\lambda x.t)$ abstraction

Propositions are terms of type o.

HOL4 Standard Library

- 15733 propositions
- 8 axioms
- 2294 definitions
- 13431 theorems

12140 of the theorems give benchmarks of the form:

> "Given certain chosen types, constants and previous propositions, prove the theorem."

"Bushy" problems in LTB terminology.

Notion of Translation

Defined types depend on terms, so the translations of types and terms must be mutually recursive. In addition, propositions are treated as special.
3 recursive procedures:

- Types $\mapsto$ TPTP types or terms
- Terms $\mapsto$ TPTP terms
- Propositions $\mapsto$ TPTP formulas

Outline

Two Kinds of Translations

I Building on Known Translations (HOLyHammer, Sledgehammer)
- Tag terms with types, Lambda lifting
- Modifications:
- More context independent
- Add axioms to gain more proofs (S,K,I)
- Use or embed polymorphic types instead of monomorphizing
- Make use of multiple sorts in the cases other than FOF

II Set Theory Semantic Motivations
- Types map to nonempty sets
- Terms map to sets
- Guard quantifiers with set membership
- Propositions map to set theoretic formulas

Higher-Order Set Theory

- ι base type of sets
- $\in : \iota \to \iota \to o$ set membership
- Some unsurprising axioms

Higher-Order Set Theory

- ι base type of sets
- $\in: \iota \to \iota \to o$ set membership
- Some unsurprising axioms
- ν type of nonempty sets
- $\in: \iota \to \nu \to o$
- $\Rightarrow: \nu \to \nu \to \nu$ for sets of functions
- $\mathrm{ap} : \iota \to \iota \to \iota$ set application
- $\mathrm{lam} : \nu \to (\iota \to \iota) \to \iota$ set level abstraction.
  $\mathrm{lam}\ X\ (\lambda x.t)$ represents the set theoretic function f such that $f(x) = t$ for $x \in X$.
- "Typing" style axioms; β rule

Tags vs. Guards

When translating $\forall x : \sigma.\varphi$:

- The I-translations use tags when necessary, translating as $\forall x. \ldots$ and using $\mathrm{tp}(x, \hat{\sigma})$ instead of x to ensure the occurence of x has type σ.
- The II-translations use set membership guards, translating as $\forall x.x \in \hat{\sigma} \to \cdots$
- For both I and II (except the FOF cases) for some monomorphic types new base types μ will be declared along with $i_\mu : \mu \to \iota$ and $j_\mu : \iota \to \mu$ satisfying appropriate properties.
- When special μ types are used, guards and tags can be avoided.
- TF1-I and TH1-I can map HOL4 types to TPTP types, since TF1 and TH1 support polymorphism.

Translation of Lambdas

- In the TH0-II case, λ-abstractions are translated using the set level lam operator.
- In other cases, λ-abstractions are translated using λ-lifting. A new function f is declared and is defined to behave in accordance with the body of the λ-abstraction.

Outline

Example (SUM CASES)

$$\forall s\, P : \alpha \to o. \forall f g : \alpha \to \rho. \text{FINITE } s \to$$
$$\text{sum } s\ (\lambda x. \text{if } Px \text{ then } fx \text{ else } gx)$$
$$= \text{sum } \{x | x \in s \wedge Px\}\ f + \text{sum } \{x | x \in s \wedge \neg Px\}\ g$$

- ρ is HOL4 type of reals
- $+$ is addition on reals
- α is an implicitly quantified type variable

Main Lemma:

$$\forall o : \beta \to \beta \to \beta. \text{monoidal } o \to$$
$$\forall s\, P : \alpha \to o. \forall f g : \alpha \to \beta. \text{FINITE } s \to$$
$$\text{iterate } o\ s\ (\lambda x. \text{if } Px \text{ then } fx \text{ else } gx)$$
$$= o\ (\text{iterate } o\ \{x | x \in s \wedge Px\}\ f)$$
$$(\text{iterate } o\ \{x | x \in s \wedge \neg Px\}\ g).$$

Example (TH0-II representation)

Focus on:

$$\lambda x.\text{if } Px \text{ then } fx \text{ else } gx$$

The TH0-II version translates this as:

$$\text{lam } A\ (\lambda x.\text{ap } (\text{ap } (\text{ap } (\text{COND } \hat{\rho})\ (Px))\ (fx))\ (gx))$$

- where COND : $\nu \to \iota$ is (polymorphic) if-then-else,
- $\hat{\rho} : \nu$ corresponds to the HOL4 type of reals and
- $A : \nu$ is the TH0 variable corresponding to the type variable α.

The corresponding λ-abstraction in the lemma translates as

$$\text{lam } A\ (\lambda x.\text{ap } (\text{ap } (\text{ap } (\text{COND } B)\ (Px))\ (fx))\ (gx))$$

Theorem provers can easily match these. Satallax can prove the example in just over 2 minutes.

Example (most other representations)

- For the representations using lambda lifting, two new functions f and g are defined for the two λ-abstractions.
- In order to prove the theorem, the ATP would need to prove a relationship between f and g to use the lemma.
- E could not prove the TF0 representations.

Outline

ATP Results

- 19 ATPs
- 60s timeout
- Call each ATP on each representation it supports

ATP Results

System	TH1-I	TH0-I	TH0-II	TF1-I	TF0-I	TF0-II	FOF-I	FOF-II	**Union**
agsyHOL		1374	1187						1605
Beagle					2007	2047			2531
cocATP		899	599						1000
CSE_E							4251	3102	4480
CVC4					4851	3991			5252
E					4277	3622	4618	3844	5118
HOLyHammer	5059								5059
iProver							2778	2894	3355
iProverMo'					2435	1639			2699
LEO-II		2579	1923						3213
Leo-III	6668	5018	3485	3458	4032	3421			7062
Metis							2353	474	2356
Princess					3646	2138			3849
Prover9							2894	1742	3128
Satallax		2207	1292						2494
SPASS							2850	3349	3821
Vampire					4837	4693	4008	4928	5929
Zipperp'n		2252	2161	3771	3099	2576			4203
Union	6824	5209	3771	4608	5732	5073	5165	5108	**7377**

- Overall 7377 (61%) solved.
- Leo-III wins using TH1-I representation (6668, 51%)
- TH0-II was harder than TH0-I (Leo-III wins both)

ATP Results

System	TH1-I	TH0-I	TH0-II	TF1-I	TF0-I	TF0-II	FOF-I	FOF-II	**Union**
agsyHOL		1374	1187						1605
Beagle					2007	2047			2531
cocATP		899	599						1000
CSE_E							4251	3102	4480
CVC4					4851	3991			5252
E					4277	3622	4618	3844	5118
HOLyHammer	5059								5059
iProver							2778	2894	3355
iProverMo'					2435	1639			2699
LEO-II		2579	1923						3213
Leo-III	6668	5018	3485	3458	4032	3421			7062
Metis							2353	474	2356
Princess					3646	2138			3849
Prover9							2894	1742	3128
Satallax		2207	1292						2494
SPASS							2850	3349	3821
Vampire					4837	4693	4008	4928	5929
Zipperp'n		2252	2161	3771	3099	2576			4203
Union	6824	5209	3771	4608	5732	5073	5165	5108	**7377**

- CVC4 wins TF0-I, Vampire wins TF0-II
- E wins FOF-I, Vampire wins FOF-II

Outline

Conclusion

- 12140 HOL4 theorems became $12140 \cdot 8$ ATP problems
- GRUNGE

  Grand **U**nified **N**Large Theory Benchmarks
- Provers for different logics can compete on the same problems.
- Leo-III has significantly improved.
- LTB competition in Brazil will determine official(ish) winner.