Model-theoretic Conservative Extension for HOL with Ad-hoc Overloading

Arve Gengelbach[1] Johannes Åman Pohjola[2,3]
Tjark Weber[1]

[1]Uppsala University, Uppsala, Sweden

[2]CSIRO's Data61, Sydney, Australia

[3]University of New South Wales, Sydney, Australia

CakeML Seminar Series
September 22, 2020

UPPSALA
UNIVERSITET

Motivation

In a logical framework,
what does a new definition change?

Semantically, at most any term changes
that uses the new symbol.

Answer for new semantics and mechanised proofs.

Motivation

Example: $T = \{\mathsf{e}_{\mathsf{bool}} \equiv \mathtt{f}_{\mathsf{bool}}\}$

What does the new definition $\mathtt{f}_{\mathsf{bool}} \equiv \mathsf{True}$ change?

Extending T with the definition $\mathtt{f}_{\mathsf{bool}} \equiv \mathsf{True}$ affects which values $\mathsf{e}_{\mathsf{bool}}$ may take.
Anything not mentioning $\mathtt{f}_{\mathsf{bool}}$ is unaffected.

Motivation: Proof-theoretic Conservativity

Theory extension by definition is *proof-theoretic conservative* if for all theories T, and their extension by $\mathtt{upd}$,
for all formula φ (in the language of T) we have:

$$T \vdash \varphi \quad \Leftrightarrow \quad T \cup \{\mathtt{upd}\} \vdash \varphi$$

Implies a *model-theoretic conservativity* (if proof-calculus complete then equivalent).

Motivation: Model-theoretic Conservativity

Theory extension by definition is *model-theoretic conservative* if for all theories T, and their extension by $\mathtt{upd}$,
for any model $\mathcal{M}$ of T, there is a model $\mathcal{M}'$ of $T \cup \{\mathtt{upd}\}$ such that for all formula φ (in the language of T) we have:

$$\mathcal{M} \models \varphi \quad \Leftrightarrow \quad \mathcal{M}' \models \varphi$$

Contribution

- Formalise model-theoretic conservativity for HOL with overloading
- Replace monolithic model construction by an incremental one
- The dual proof-theoretic conservativity (as above) may hold. [1]

[1] Proven in a weaker form by Kunčar and Popescu.

Higher-Order Logic (HOL)

- Typed λ-calculus $\quad x_\sigma \mid c_\sigma \mid (s_{\sigma\to\tau}\ t_\sigma)_\tau \mid (\lambda x_\sigma.\, t_\tau)_{\sigma\to\tau}$
- Rank 1 polymorphism
- With built-in types $\quad \to, \mathsf{bool}$
  and a built-in constant $\quad =_{\alpha\to\alpha\to\mathsf{bool}}$
- We say *symbols* for types and constants

Definitions

- Overloaded *constant specification*
  Given witnesses, simultaneously introduce several constants satisfying a property.
  Example: $\mathsf{c}_{\mathbb{N}}, \mathsf{c}_{\mathsf{bool}}, \mathsf{d}_{\mathsf{bool}} \equiv 2, \mathsf{True}, \mathsf{False}$
  satisfying $\mathsf{c}_{\mathbb{N}} \leq 4 \wedge \mathsf{c}_{\mathsf{bool}} \neq \mathsf{d}_{\mathsf{bool}}$
- Type $\tau \equiv t_{\sigma \to \mathsf{bool}}$ meaning $\tau \subseteq \sigma$, where t holds with constants $\mathsf{abs}_{\tau \to \sigma}$ and $\mathsf{rep}_{\sigma \to \tau}$
  Example: $2\mathbb{N} \equiv \mathsf{even}_{\mathbb{N} \to \mathsf{bool}}$

We consider only non-overlapping definitions

Example of overlapping definitions:

$\mathsf{c}_{\alpha \times \mathsf{bool}}, \mathsf{c}_{\mathsf{bool} \times \alpha} \equiv t, t'$ have common instance $\mathsf{c}_{\mathsf{bool} \times \mathsf{bool}}$

Non-built-ins

We are interested in the non-built-in symbols:

- Top-level non-built-in types $\cdot^\bullet$

  $(\mathbb{N} \to \mathsf{bool})^\bullet = \{\mathbb{N}\}$

  $(\mathsf{map}_{(\alpha\to\beta)\to\alpha\ \mathsf{list}\to\beta\ \mathsf{list}})^\bullet = \{\alpha, \beta, \alpha\ \mathsf{list}, \beta\ \mathsf{list}\}$
- Non-built-in constant instances $\cdot^\circ$

  $(\mathsf{even} = (\lambda x_{\mathbb{N}}.\ x\ \mathsf{mod}\ 2 = 0))^\circ = \{\mathsf{even}, \mathsf{mod}, 2, 0\}$

Dependency Relation

[Kunčar & Popescu 2015, Åman Pohjola & Gengelbach 2020]

Track a definition's dependencies (to disallow cyclic definitions).

- $u \equiv t$, and $v \in t^\bullet \cup t^\circ$ then $u \rightsquigarrow v$
  Example: $2\mathbb{N} \rightsquigarrow \mathbb{N}$, $2\mathbb{N} \rightsquigarrow \mathsf{even}_{\mathbb{N}\to\mathsf{bool}}$
  $\mathsf{c}_{\mathbb{N}} \rightsquigarrow 2, \mathsf{c}_{\mathsf{bool}} \rightsquigarrow \mathsf{True}, \mathsf{d}_{\mathsf{bool}} \rightsquigarrow \mathsf{False}$
  (from $\mathsf{c}_{\mathbb{N}}, \mathsf{c}_{\mathsf{bool}}, \mathsf{d}_{\mathsf{bool}} \equiv 2, \mathsf{True}, \mathsf{False}$)
- $\mathsf{c}_\sigma \rightsquigarrow v$ for $v \in \sigma^\bullet$
  Example: $\mathsf{map}_{(\alpha\to\beta)\to\alpha\ \mathsf{list}\to\beta\ \mathsf{list}} \rightsquigarrow v$
  for $v \in \{\alpha,\ \beta,\ \alpha\ \mathsf{list},\ \beta\ \mathsf{list}\}$
- $(\sigma_1, \ldots, \sigma_n)k \rightsquigarrow \sigma_i$ for a type constructor k
  Reason/Example: $\tau \to \sigma \rightsquigarrow \tau$ and $\tau \to \sigma \rightsquigarrow \sigma$

Lazy Ground Semantics [Åman Pohjola & Gengelbach 2020]

- φ_{bool} satisfied w.r.t. $[\cdot]$ iff
  for all ground type substitutions ρ
  and all variable assignments ξ_ρ: $[\varphi]_{\xi_\rho} = \mathsf{true}$
  - Earlier semantics: $[\rho(\varphi)]_{\xi_\rho} = \mathsf{true}$
    Problem: Term variables x_α and x_{bool} are distinct,
    but immediatly applying ρ equates these.
  - Lazy semantics
    For $\rho(\sigma)$ ground type, have $[\mathsf{c}_\sigma]_{\xi_\rho} = [\mathsf{c}_{\rho(\sigma)}]$ and
    $[x_\sigma]_{\xi_\rho} = \xi_\rho(x_\sigma)$ such that $\xi_\rho(x_\sigma) \in [\rho(\sigma)]$
- $\mathcal{M} \models T$ iff every $\varphi_{\mathsf{bool}} \in T$ is satisfied w.r.t. $\mathcal{M}$.

Recipe to Model-theoretic Conservativity

For a definitional theory T, with a model $\mathcal{M} \models T$ we want a model of an extension of T by a new definition $u \equiv t$.

- Reuse interpretations from $\mathcal{M}$ that are unaffected by the new definition $u \equiv t$ (called F_u).
- Define interpretations for the symbols that are affected by the definition $u \equiv t$.

Symbol-independent Fragment [Gengelbach & Weber 2017]

The u-independent fragment[2] is

$$F_u := \mathsf{Symb} \setminus \{x | \exists u' \in u, \rho.\ x \rightsquigarrow^{\downarrow *} \rho(u')\}.$$

F_u contains all symbols that are not depending on any instance from u.

Example: $\mathsf{c}_\alpha \equiv \mathsf{d}_\alpha$, $\mathsf{d}_{\mathsf{bool}} \equiv \mathsf{True}$

$\mathsf{c}_{\mathsf{bool}}, \mathsf{d}_{\mathsf{bool}} \notin F_{\mathsf{d}_{\mathsf{bool}}}$ by $\mathsf{c}_{\mathsf{bool}} \rightsquigarrow^{\downarrow *} \mathsf{d}_{\mathsf{bool}}$

$\mathsf{c}_{\alpha\ \mathsf{list}} \in F_{\mathsf{d}_{\mathsf{bool}}}$ by $\mathsf{c}_{\alpha\ \mathsf{list}} \not\rightsquigarrow^{\downarrow *} \mathsf{d}_{\mathsf{bool}}$

[2] $\cdot^{\downarrow}$ type-substitutive closure, $\cdot^*$ reflexive-transitive closure

Model-theoretic Conservative Extension

Claim:
For a definitional theory $T \cup \{u \equiv t\}$ with $\mathcal{M} \models T$
there exists a model extension $\mathcal{M}' \models T \cup \{u \equiv t\}$
such that $\mathcal{M}$ and $\mathcal{M}'$ interpret terms built from F_u equally.

Example: $T = \{\mathsf{c}_\alpha \equiv \mathsf{d}_\alpha\}$, $T' = T \cup \{\mathsf{d}_{\mathsf{bool}} \equiv \mathsf{True}\}$.
Any model $\mathcal{M}$ for T has an extension $\mathcal{M}'$ for T'.
Might have $\mathcal{M}(\mathsf{c}_{\mathsf{bool}}) \neq \mathcal{M}'(\mathsf{c}_{\mathsf{bool}})$ because $\mathsf{c}_{\mathsf{bool}} \notin F_{\mathsf{d}_{\mathsf{bool}}}$.
But $\mathcal{M}(\mathsf{c}_{\alpha\ \mathsf{list}}) = \mathcal{M}'(\mathsf{c}_{\alpha\ \mathsf{list}})$.

Implementation: Mutually Recursive Model Construction

```
type_interpretation_ext ind upd T Δ Γ τ =
  if ~wellformed (T ∪ {upd})
  then One
  else if  (∀tm. upd ≠ NewAxiom tm)
    ∧ τ ∈ indep_frag_upd (T ∪ {upd}) upd
  then Δ τ
  else ... // as in Åman Pohjola & Gengelbach, LPAR 2020

term_interpretation_ext ind upd T Δ Γ c_τ = ...
```

Remaining claim

For a theory T with model $\mathcal{M}$ any axiom from T
is valid in the constructed model $\mathcal{M}'$ for $T \cup \{\mathtt{upd}\}$.

Assumption for constant specification

- For a constant specification
  $\mathsf{d}_{\mathsf{bool}}, \mathsf{e}_{\mathsf{bool}} \equiv \mathsf{False}, (\mathsf{c}_{\mathsf{bool}} \Rightarrow \mathsf{True})$ with axiom $\mathsf{d}_{\mathsf{bool}} \neq \mathsf{e}_{\mathsf{bool}}$
  updated with $\mathsf{c}_{\mathsf{bool}} = \mathsf{True}$, show $\mathcal{M}'(\mathsf{d}_{\mathsf{bool}} \neq \mathsf{e}_{\mathsf{bool}}) = \mathsf{true}$
- Here, $\mathsf{d}_{\mathsf{bool}} \not\rightsquigarrow \mathsf{c}_{\mathsf{bool}}$ and $\mathsf{e}_{\mathsf{bool}} \rightsquigarrow \mathsf{c}_{\mathsf{bool}}$,
  thus $\mathsf{d}_{\mathsf{bool}} \in F_{\mathsf{c}_{\mathsf{bool}}}$ and $\mathsf{e}_{\mathsf{bool}} \notin F_{\mathsf{c}_{\mathsf{bool}}}$.
- Knowing $\mathcal{M}(\mathsf{d}_{\mathsf{bool}}) = \mathcal{M}'(\mathsf{d}_{\mathsf{bool}})$ and $\mathcal{M}(\mathsf{d}_{\mathsf{bool}}) = \mathsf{false}$
  we can prove $\mathcal{M}'(\mathsf{d}_{\mathsf{bool}} \neq \mathsf{e}_{\mathsf{bool}}) = \mathsf{true}$

Model-theoretic Conservative Extension

Theorem:
For a definitional theory $T \cup \{u \equiv t\}$ with $\mathcal{M} \models T$
and if $\mathcal{M}(c) = \mathcal{M}(t')$ for any constant c with witness t' introduced by constant specification then
there exists a model extension $\mathcal{M}' \models T \cup \{u \equiv t\}$
such that $\mathcal{M}$ and $\mathcal{M}'$ interpret terms built from F_u equally.

By construction the additional assumption holds for $\mathcal{M}'$.

Corollary: Consistency
Any definitional theory has a model.

What does a new definition $u \equiv t$ change?

In a model, at most some symbols that are expressed
in terms of instances of the new defined symbol u.

$$\{x | \exists u' \in u, \rho.\ x \rightsquigarrow^{\downarrow *} \rho(u')\}$$

Semantically, definitions are *merely abbreviations*.
And we have formalised proof for it:

`https://code.cakeml.org/tree/master/candle/overloading`