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
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: \[ T = \{ e_{\text{bool}} \equiv f_{\text{bool}} \} \]

What does the new definition \( f_{\text{bool}} \equiv \text{True} \) change?

Extending \( T \) with the definition \( f_{\text{bool}} \equiv \text{True} \) affects which values \( e_{\text{bool}} \) may take.
Anything not mentioning \( f_{\text{bool}} \) is unaffected.
Motivation: Proof-theoretic Conservativity

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

$$T \vdash \varphi \iff T \cup \{\text{upd}\} \vdash \varphi$$

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 $\text{upd}$, for any model $\mathcal{M}$ of $T$, there is a model $\mathcal{M}'$ of $T \cup \{ \text{upd} \}$ such that for all formula $\varphi$ (in the language of $T$) we have:

$$\mathcal{M} \models \varphi \iff \mathcal{M}' \models \varphi$$
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 $\lambda$-calculus
  $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 $\rightarrow$, $\text{bool}$
  and a built-in constant $=_{\alpha \to_\alpha \to \text{bool}}$

- We say symbols for types and constants
Definitions

- Overloaded constant specification
  Given witnesses, simultaneously introduce several constants satisfying a property.
  Example: $c_\mathbb{N}, c_{\text{bool}}, d_{\text{bool}} \equiv 2, \text{True}, \text{False}$
  satisfying $c_\mathbb{N} \leq 4 \land c_{\text{bool}} \neq d_{\text{bool}}$

- Type $\tau \equiv t_{\sigma \rightarrow \text{bool}}$ meaning $\tau \subseteq \sigma$, where $t$ holds with constants abs$_{\tau \rightarrow \sigma}$ and rep$_{\sigma \rightarrow \tau}$
  Example: $2\mathbb{N} \equiv \text{even}_\mathbb{N \rightarrow \text{bool}}$

We consider only non-overlapping definitions
Example of overlapping definitions:
$c_{\alpha \times \text{bool}}, c_{\text{bool} \times \alpha} \equiv t, t'$ have common instance $c_{\text{bool} \times \text{bool}}$
We are interested in the non-built-in symbols:

- **Top-level non-built-in types**: 
  
  \[ (\mathbb{N} \to \text{bool})^\bullet = \{\mathbb{N}\} \]
  
  \[ (\text{map}(\alpha \to \beta) \to \alpha \text{ list} \to \beta \text{ list})^\bullet = \{\alpha, \beta, \alpha \text{ list}, \beta \text{ list}\} \]

- **Non-built-in constant instances**: 
  
  \[ (\text{even} = (\lambda x_{\mathbb{N}}. x \mod 2 = 0))^\circ = \{\text{even}, \text{mod}, 2, 0\} \]
Track a definition’s dependencies (to disallow cyclic definitions).

- $u \equiv t$, and $v \in t^\bullet \cup t^\circ$ then $u \leadsto v$
  
  Example: $2\mathbb{N} \leadsto \mathbb{N}$, $2\mathbb{N} \leadsto \text{even}_{\mathbb{N} \rightarrow \text{bool}}$
  
  $c_{\mathbb{N}} \leadsto 2$, $c_{\text{bool}} \leadsto \text{True}$, $d_{\text{bool}} \leadsto \text{False}$
  
  (from $c_{\mathbb{N}}$, $c_{\text{bool}}$, $d_{\text{bool}} \equiv 2$, True, False)

- $c_{\sigma} \leadsto v$ for $v \in \sigma^\bullet$
  
  Example: $\text{map}(\alpha \rightarrow \beta) \rightarrow \alpha \text{list} \rightarrow \beta \text{list} \leadsto v$
  
  for $v \in \{\alpha, \beta, \alpha \text{list}, \beta \text{list}\}$

- $(\sigma_1, \ldots, \sigma_n)k \leadsto \sigma_i$ for a type constructor $k$
  
  Reason/Example: $\tau \rightarrow \sigma \leadsto \tau$ and $\tau \rightarrow \sigma \leadsto \sigma$
Lazy Ground Semantics

\[ \varphi_{\text{bool}} \text{ satisfied w.r.t. } \lfloor \cdot \rfloor \text{ iff} \]
for all ground type substitutions \( \rho \)
and all variable assignments \( \xi_\rho : \lfloor \varphi \rfloor_\rho = \text{true} \)

Earlier semantics: \( \lfloor \rho(\varphi) \rfloor_\rho = \text{true} \)
Problem: Term variables \( x_\alpha \) and \( x_{\text{bool}} \) are distinct,
but immediately applying \( \rho \) equates these.

Lazy semantics
For \( \rho(\sigma) \) ground type, have \( \lfloor c_\sigma \rfloor_\rho = \lfloor c_{\rho(\sigma)} \rfloor \) and
\( \lfloor x_\sigma \rfloor_\rho = \xi_\rho(x_\sigma) \) such that \( \xi_\rho(x_\sigma) \in \lfloor \rho(\sigma) \rfloor \)

\( \mathcal{M} \models T \text{ iff every } \varphi_{\text{bool}} \in T \text{ is satisfied w.r.t. } \mathcal{M}. \)
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 $u \equiv t$. 

A. Gengelbach, J. Åman Pohjola, T. Weber : Model-theoretic Conservative Extension for HOL with Ad-hoc Overloading
The $u$-independent fragment\(^2\) is

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

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

**Example:** $c_\alpha \equiv d_\alpha, \ d_{\text{bool}} \equiv \text{True}$

- $c_{\text{bool}}, d_{\text{bool}} \not\in F_{d_{\text{bool}}}$ by $c_{\text{bool}} \rightsquigarrow^\downarrow^* d_{\text{bool}}$
- $c_\alpha \text{ list} \in F_{d_{\text{bool}}}$ by $c_\alpha \text{ list} \rightsquigarrow^\downarrow^* d_{\text{bool}}$

\(^2\) \text{type-substitutitive closure, } \cdot^* \text{ 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: $T = \{c_\alpha \equiv d_\alpha\}$, $T' = T \cup \{d_{\text{bool}} \equiv \text{True}\}$.
Any model $\mathcal{M}$ for $T$ has an extension $\mathcal{M}'$ for $T'$.
Might have $\mathcal{M}(c_{\text{bool}}) \neq \mathcal{M}'(c_{\text{bool}})$ because $c_{\text{bool}} \not\in F_{d_{\text{bool}}}$. But $\mathcal{M}(c_\alpha \text{ list}) = \mathcal{M}'(c_\alpha \text{ list})$. 

A. Gengelbach, J. Åman Pohjola, T. Weber : Model-theoretic Conservative Extension for HOL with Ad-hoc Overloading
**Implementation:**

**Mutually Recursive Model Construction**

\[
\text{type\_interpretation\_ext \ ind \ upd \ } T \ \Delta \ \Gamma \ \tau = \\
\text{if } \neg \text{wellformed} \ (T \cup \{\text{upd}\}) \\
\text{then One} \\
\text{else if } (\forall \text{tm. } \text{upd} \neq \text{NewAxiom tm}) \\
\quad \land \ \tau \in \text{indep\_frag\_upd} \ (T \cup \{\text{upd}\}) \ \text{upd} \\
\text{then } \Delta \ \tau \\
\text{else } \ldots \ // \ as \ in \ Åman \ Pohjola \ & \ Gengelbach, \ LPAR \ 2020
\]

\[
\text{term\_interpretation\_ext \ ind \ upd \ } T \ \Delta \ \Gamma \ c_\tau = \ldots
\]
For a theory $T$ with model $\mathcal{M}$ any axiom from $T$ is valid in the constructed model $\mathcal{M}'$ for $T \cup \{\text{upd}\}$.
Assumption for constant specification

For a constant specification $d_{bool}, e_{bool} \equiv False, (c_{bool} \Rightarrow True)$ with axiom $d_{bool} \neq e_{bool}$ updated with $c_{bool} = True$, show $M'(d_{bool} \neq e_{bool}) = true$

Here, $d_{bool} \not\rightsquigarrow c_{bool}$ and $e_{bool} \rightsquigarrow c_{bool}$, thus $d_{bool} \in F_{c_{bool}}$ and $e_{bool} \notin F_{c_{bool}}$.

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

By construction the additional assumption holds for \( M' \).

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 $u$.

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

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

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