Using LEO-II to Prove Properties of an Explicit Substitution M-set Model

Bachelor Thesis - Final Talk

Xin Zhang

Supervisor : Chad E. Brown
Reponsible Professor : Gert Smolka

October 27, 2008

Outline

We used LEO-II [3] to verify properties of an M-set model

M-set Model [4]

M-sets

A monoid is a triple $< M, op, e >$

- We write $m \cdot n$ for op(m,n)
- $(m \cdot (n \cdot k)) = ((m \cdot n) \cdot k)$
- $m \cdot e = m = e \cdot m$

An M-set is a pair $<A,\alpha>$

- We write $a * m$ *for* $\alpha(a, m)$
- $(a * m) * n = a * (m \cdot n)$
- $a * e = a$

Explicit Substitution [1, 5]

Terms $(a, b \ldots) := 1|(a\,b)|(\lambda a)|(a[s])$

Explicit Substitution $(s, t \ldots) := id| \uparrow |(s \circ t)|(a.s)$

Representation I

Let T be the set of σ-normal terms and M be the set of σ-normal substitutions. In thf syntax [2] we can represent these sets as constants of type ι.

- `in` is a constant of type $\iota \rightarrow \iota \rightarrow o$
- `term` is a constant of type ι
- `subst` is a constant of type ι

Representation I

- one $:=$ 1
- (ap a b) $:= (ab)^{\downarrow\sigma}$
- (lam a) $:= \lambda a^{\downarrow\sigma}$
- (sub a m) $:= a[m]^{\downarrow\sigma}$ where $a \in T$ and $m \in M$
- id $:=$ id
- sh $:= \uparrow$
- (push a m) $:= (a.m)^{\downarrow\sigma}$
- (comp m n) $:= (m \circ n)^{\downarrow\sigma}$

`one` is a constant of type ι
`one_p` is an abbreviation defined by

$$\mathtt{in}\,\mathtt{one}\,\mathtt{term}$$

`ap` is a constant of type $\iota \to \iota \to \iota$
`ap_p` is an abbreviation defined by

$$\forall A_\iota.\mathtt{in}\,A\,\mathtt{term} \Rightarrow \forall B_\iota.\mathtt{in}\,B\,\mathtt{term} \Rightarrow \mathtt{in}\,(\mathtt{ap}\,A\,B)\,\mathtt{term}$$

`lam` is a constant of type $\iota \to \iota$
`lam_p` is an abbreviation defined by

$$\forall A_\iota.\mathtt{in}\,A\,\mathtt{term} \Rightarrow \mathtt{in}\,(\mathtt{lam}\,A)\,\mathtt{term}$$

...

Representation II

We declared the base type `term` and `subst`.

- `one` is a constant of type `term`
- `ap` is a constant of type `term` $\rightarrow$ `term` $\rightarrow$ `term`
- `lam` is a constant of type `term` $\rightarrow$ `term`
- `sub` is a constant of type `term` $\rightarrow$ `subst` $\rightarrow$ `term`
- `id` is a constant of type `subst`
- `sh` is a constant of type `subst`
- `push` is a constant of type `term` $\rightarrow$ `subst` $\rightarrow$ `subst`
- `comp` is a constant of type `subst` $\rightarrow$ `subst` $\rightarrow$ `subst`

$\mathtt{Axapp}$: $((a\,b)[s])^{\downarrow\sigma} = ((a[s])^{\downarrow\sigma}\,(b[s])^{\downarrow\sigma})$ for $a, b \in T$ and $s \in M$.

- *Rep I*

$$\forall A_\iota.\mathtt{in}\,A\,\mathtt{term}\Rightarrow\forall B_\iota.\mathtt{in}\,B\,\mathtt{term}\Rightarrow$$

$$\forall M_\iota.\mathtt{in}\,M\,\mathtt{subst}\Rightarrow$$

$$\mathtt{sub}\,(\mathtt{ap}\,A\,B)\,M = \mathtt{ap}\,(\mathtt{sub}\,A\,M)\,(\mathtt{sub}\,B\,M)$$

- *Rep II*

$$\forall A_{\mathtt{term}} B_{\mathtt{term}} M_{\mathtt{subst}}.$$

$$\mathtt{sub}\,(\mathtt{ap}\,A\,B)\,M = \mathtt{ap}\,(\mathtt{sub}\,A\,M)\,(\mathtt{sub}\,B\,M)$$

$\mathtt{axidl}, \mathtt{axmap}, \mathtt{axvarcons}\cdots$

Basics Results

Name	*Rep I*		*Rep II*		
	`gthm`	`lthm`	`gthm`	`lthm`	`lthm` with lemmas
Substmonoid	11.589s	5.165s	1.324s	0.521s	NA
Termmset	3.299s	0.564s	1.354s	0.505s	NA
Hoasapinj1	3.573s	0.481s	1.411s	0.515s	NA
Hoasapinj2	3.680s	0.479s	1.452s	0.509s	NA
Hoaslamnotap	6.194s	0.778s	1.622s	0.508s	NA
Hoaslamnotvar	6.495s	0.760s	1.685s	0.509s	NA
Hoasapnotvar	6.671s	0.575s	1.762s	0.503s	NA
Hoasap	3.317s	0.437s	NA	NA	NA
Hoaslam	3.343s	0.636s	NA	NA	NA
Hoaslaminj	-	-	1.556s	0.533s	NA
Induction2	-	-	-	0.581s	NA
Pushprop	-	-	-	-	0.655s
Hoasinduction	-	-	-	-	0.807s
Induction2lem	-	-	-	-	-

Theorem

($\mathtt{hoasap}$) For $m, n \in M$ and $a, b \in T$, we have $\mathtt{hoasap}(m, a)(n, b) \in T$.

We encoded this theorem in LEO with Representation I:

- hoasap

$$\lambda M_\iota A_\iota N_\iota B_\iota . \mathtt{ap}\,(\mathtt{sub}\,A\,N)\,B$$

- hoasap_p

$$\forall M_\iota . \mathtt{in}\,M\,\mathtt{subst} \Rightarrow \forall A_\iota . \mathtt{in}\,A\,\mathtt{term} \Rightarrow$$

$$\forall N_\iota . \mathtt{in}\,N\,\mathtt{subst} \Rightarrow \forall B_\iota . \mathtt{in}\,B\,\mathtt{term} \Rightarrow$$

$$\mathtt{in}\,(\mathtt{hoasap}\,M\,A\,N\,B)\,\mathtt{term}$$

We encoded the definition of hoasap in LEO with Representation II:

$$\lambda M_{\mathtt{subst}} A_{\mathtt{term}} N_{\mathtt{subst}} B_{\mathtt{term}} . \mathtt{ap}\,(\mathtt{sub}\,A\,N)\,B$$

We want our model to satisfy this axiom:

$$\forall f \forall g(((Lam\ f) = (Lam\ g)) \Rightarrow (f = g))$$

We interpret this property in our model as:

Theorem

(`hoaslaminj`*) Let $f, g : M \times T \to T$ be functions such that*

$$f(m, a)n = f(mn, an)$$

and

$$g(m, a)n = g(mn, an)$$

for all $a \in T$ and $m, n \in M$. If $\mathtt{hoaslam}(id, f) = \mathtt{hoaslam}(id, g)$*, then $f = g$.*

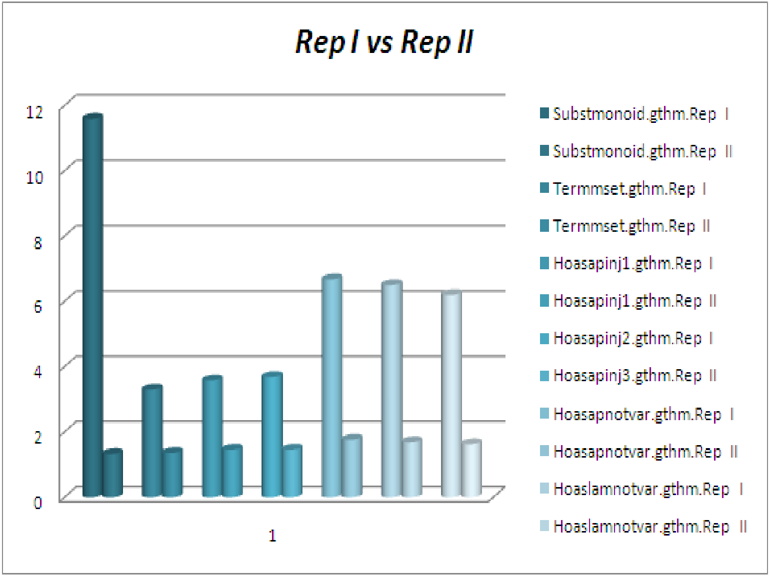
We encoded this theorem in LEO with Representation I as follows:

$$\forall F_{\iota\to\iota\to\iota}.(\forall M_\iota.\mathtt{in}\,M\,\mathtt{subst}\Rightarrow\forall A_\iota.\mathtt{in}\,A\,\mathtt{term}\Rightarrow\mathtt{in}\,(F\,M\,A)\,\mathtt{term})\Rightarrow$$
$$(\forall M_\iota.\mathtt{in}\,M\,\mathtt{subst}\Rightarrow\forall A_\iota.\mathtt{in}\,A\,\mathtt{term}\Rightarrow\forall N_\iota.\mathtt{in}\,N\,\mathtt{subst}\Rightarrow$$
$$\mathtt{sub}\,(F\,M\,A)\,N = F\,(\mathtt{comp}\,M\,N)\,(\mathtt{sub}\,A\,N))\Rightarrow$$
$$\forall G_{\iota\to\iota\to\iota}.(\forall M_\iota.\mathtt{in}\,M\,\mathtt{subst}\Rightarrow\forall A_\iota.\mathtt{in}\,A\,\mathtt{term}\Rightarrow\mathtt{in}\,(G\,M\,A)\,\mathtt{term})\Rightarrow$$
$$(\forall M_\iota.\mathtt{in}\,M\,\mathtt{subst}\Rightarrow\forall A_\iota.\mathtt{in}\,A\,\mathtt{term}\Rightarrow\forall N_\iota.\mathtt{in}\,N\,\mathtt{subst}\Rightarrow$$
$$\mathtt{sub}\,(G\,M\,A)\,N = G\,(\mathtt{comp}\,M\,N)\,(\mathtt{sub}\,A\,N))\Rightarrow$$
$$\mathtt{hoaslam}\,\mathtt{id}\,(\lambda M_\iota A_\iota.F\,M\,A) = \mathtt{hoaslam}\,\mathtt{id}\,(\lambda M_\iota A_\iota.G\,M\,A)\Rightarrow$$
$$\forall M_\iota.\mathtt{in}\,M\,\mathtt{subst}\Rightarrow\forall A_\iota.\mathtt{in}\,A\,\mathtt{term}\Rightarrow F\,M\,A = G\,M\,A$$

We encoded this theorem in LEO with Representation II as follow:

$$\forall F_{\mathtt{subst}\to\mathtt{term}\to\mathtt{term}}.$$
$$(\forall M_{\mathtt{subst}} A_{\mathtt{term}} N_{\mathtt{subst}}.\mathtt{sub}\,(F\,M\,A)\,N = F\,(\mathtt{comp}\,M\,N)\,(\mathtt{sub}\,A\,N))\Rightarrow$$
$$\forall G_{\mathtt{subst}\to\mathtt{term}\to\mathtt{term}}.$$
$$(\forall M_{\mathtt{subst}} A_{\mathtt{term}} N_{\mathtt{subst}}.\mathtt{sub}\,(G\,M\,A)\,N = G\,(\mathtt{comp}\,M\,N)\,(\mathtt{sub}\,A\,N))\Rightarrow$$
$$\mathtt{hoaslam\,id}\,(\lambda M_{\mathtt{subst}} A_{\mathtt{term}}.F\,M\,A) = \mathtt{hoaslam\,id}\,(\lambda M_{\mathtt{subst}} A_{\mathtt{term}}.G\,M\,A)\Rightarrow$$
$$\forall M_{\mathtt{subst}} A_{\mathtt{term}}.F\,M\,A = G\,M\,A$$

Hoaslaminj


Rep I vs Rep II
12
10
8
6
4
2
0
1
Substmonoid.gthm.Rep I
Substmonoid.gthm.Rep II
Termmset.gthm.Rep I
Termmset.gthm.Rep II
Hoasapinj1.gthm.Rep I
Hoasapinj1.gthm.Rep II
Hoasapinj2.gthm.Rep I
Hoasapinj3.gthm.Rep II
Hoasapnotvar.gthm.Rep I
Hoasapnotvar.gthm.Rep II
Hoaslamnotvar.gthm.Rep I
Hoaslamnotvar.gthm.Rep II

We want to prove this property in our model.

Theorem

(`induction2`) *Let Φ be a property such that the following hold:*

1. *For all $x \in Var$, x satisfies Φ.*
2. *For all $a, b \in T$, if a and b satisfy Φ, then $(a\,b)^{\downarrow\sigma}$ satisfies Φ.*
3. *For all $a \in T$, if $(a[b.id])^{\downarrow\sigma}$ satisfies Φ whenever $b \in T$ satisfies Φ, then $(\lambda a)^{\downarrow\sigma}$ satisfies Φ.*

Then for all $a \in T$, a satisfies Φ.

Global and Local Theorems

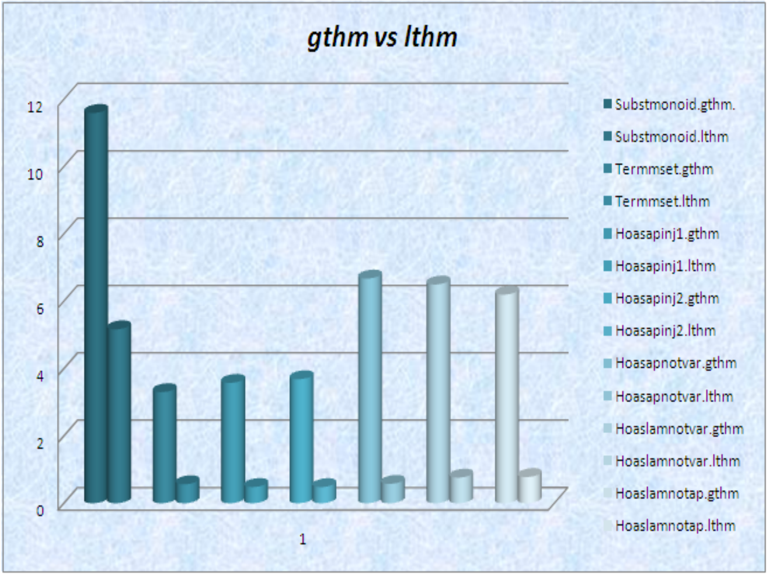
The `induction2_gthm` is:

axapp⇒axvarcons⇒axvarid⇒axabs⇒axclos⇒axidl⇒
axshiftcons⇒axassoc⇒axmap⇒axidr⇒axvarshift⇒axscons⇒
ulamvar1⇒ulamvarsh⇒ulamvarind⇒apinj1⇒apinj2⇒laminj⇒
shinj⇒lamnotap⇒apnotvar⇒lamnotvar⇒induction
⇒pushprop⇒induction2lem⇒induction2

The `induction2_lthm` is:

axvarid⇒induction2lem⇒induction2

Induction2


gthm vs lthm
12
10
8
6
4
2
0
1
Substmonoid.gthm.
Substmonoid.lthm
Termmset.gthm
Termmset.lthm
Hoasapinj1.gthm
Hoasapinj1.lthm
Hoasapinj2.gthm
Hoasapinj2.lthm
Hoasapnotvar.gthm
Hoasapnotvar.lthm
Hoaslamnotvar.gthm
Hoaslamnotvar.lthm
Hoaslamnotap.gthm
Hoaslamnotap.lthm

Hoasinduction

In the HOAS [6] theory we have the following induction axiom

$$\forall p((\forall x(\mathrm{Var}\ x \Rightarrow (px)))$$
$$\wedge(\forall x \forall y(px \wedge py \Rightarrow p(\mathrm{Ap}\ xy)))$$
$$\wedge(\forall f((\forall x(px \Rightarrow p(fx))) \Rightarrow p(\mathrm{Lam}\ f))) \Rightarrow (\forall x(px))))$$

Hoasinduction

Theorem

(`hoasinduction`*) Let* $\Psi : M \times T \to \mathcal{P}(M)$ *be a function such that*

$$kn \in \Psi(m, a) \textit{ iff } n \in \Psi(mk, ak)$$

for all $a \in T$ *and* $m, n, k \in M$*. Suppose we have the following:*

1. *For all* $x \in T$*, if* $id \in \mathtt{hoasvar}(id, x)$*, then* $id \in \Psi(id, x)$*.*
2. *For all* $a, b \in T$*, if* $id \in \Psi(id, a)$ *and* $id \in \Psi(id, b)$*, then* $id \in \Psi(id, \mathtt{hoasap}(id, a)(id, b))$
3. *For all* $f : M \times T \to T$ *such that* $f(m, a)n = f(mn, an)$ *for all* $a \in T$ *and* $m, n \in M$*, if* $id \in \Psi(id, a)$ *implies* $id \in \Psi(id, f(id, a))$ *for all* $a \in T$*, then* $id \in \Psi(id, \mathtt{hoaslam}(id, f))$*.*

Then for all $a \in T$*,* $id \in \Psi(id, a)$*.*

`hoasinduction_lthm_1` is:

$$\mathtt{induction2} \Rightarrow \mathtt{axvarid} \Rightarrow \mathtt{axclos} \Rightarrow \mathtt{axvarshift} \Rightarrow$$

$$\mathtt{axmap} \Rightarrow \mathtt{axidl} \Rightarrow \mathtt{hoasinduction}$$

Proof

Use Induction2 with $\Phi\ x$ iff $id \in \Psi(id, a)$.

Theorem

(`hoasinduction`) Let $\Psi : M \times T \longrightarrow \mathcal{P}(M)$ be a function such that

$$kn \in \Psi(m, a) \textit{ iff } n \in \Psi(mk, ak)$$

for all $a \in T$ and $m, n, k \in M$. Suppose we have the following:

1. *For all $x \in T$, if $id \in \mathtt{hoasvar}(id, x)$, then $id \in \Psi(id, x)$.*
2. *For all $a, b \in T$, if $id \in \Psi(id, a)$ and $id \in \Psi(id, b)$, then $id \in \Psi(id, \mathtt{hoasap}(id, a)(id, b))$*
3. *For all $f : M \times T \longrightarrow T$ such that $f(m, a)n = f(mn, an)$ for all $a \in T$ and $m, n \in M$, if $id \in \Psi(id, a)$ implies $id \in \Psi(id, f(id, a))$ for all $a \in T$, then $id \in \Psi(id, \mathtt{hoaslam}(id, f))$.*

Then for all $a \in T$, $id \in \Psi(id, a)$.

Theorem

(`induction2`) Let Φ be a property such that the following hold:

1. *For all $x \in Var$, x satisfies Φ.*
2. *For all $a, b \in T$, if a and b satisfy Φ, then $(a\,b)^{\downarrow\sigma}$ satisfies Φ.*
3. *For all $a \in T$, if $(a[b.id])^{\downarrow\sigma}$ satisfies Φ whenever $b \in T$ satisfies Φ, then $(\lambda a)^{\downarrow\sigma}$ satisfies Φ.*

Then for all $a \in T$, a satisfies Φ.

We define the `hoasinduction_p_and_p_prime` and `hoasinduction_lem0`.

	`Induction2`	`Hoasinduction`
Property	$Q_{\mathtt{term} \to o}$	$P_{\mathtt{subst} \to \mathtt{term} \to \mathtt{subst} \to o}$

- `hoasinduction_p_and_p_prime`

$$\lambda PQ.\forall X_{\mathtt{term}}.Q\,X \Leftrightarrow P\,\mathtt{id}\,X\,\mathtt{id}$$

- `hoasinduction_lem0`

$$\forall P.\exists Q.\mathtt{hoasinduction\_p\_and\_p\_prime}\,P\,Q$$

we match three condition of `hoasinduction` with three condition of `induction2`

- `hoasinduction_lem1v2`
- `hoasinduction_lem2v2`
- `hoasinduction_lem3v2`

`hoasinduction_lthm_2` should be:

hoasinduction_lem0⇒hoasinduction_lem1v2⇒

hoasinduction_lem2v2⇒hoasinduction_lem3v2

⇒induction2⇒hoasinduction

LEO could prove this and also following version:

hoasinduction_lem0⇒hoasinduction_lem3v2⇒

induction2⇒axvarid⇒hoasinduction

We want to prove this result in LEO.

Theorem

(`pushprop`) *Let* Φ *be a property,* $a \in T$ *and* $m \in M$*. Assume for all* $x \in Var$*,* $(x[m])^{\downarrow\sigma}$ *satisfies* Φ*. Assume* a *satisfies* Φ*. Then* $(x[a.m])^{\downarrow\sigma}$ *satisfies* Φ *for all* $x \in Var$*.*

Theorem

(`pushprop_lem0`) For every property ϕ, `term` a and `substitution` m,there is a property ϕ' such that for every term x, x satisfies ϕ' iff $(x[a.m])^{\downarrow\sigma}$ satisfies ϕ.

Proof: Just define ϕ' in this way.

Pushprop

Also LEO could prove the following version of `pushprop_lthm`:

`pushprop_lem0`⇒`ulamvar1`⇒`axvarcons`⇒`axclos`⇒

`axshiftcons`⇒`ulamvarind`⇒`pushprop`

Conclusion

- LEO is sensitive to the representation.
- LEO is sensitive to how many assumptions are given.
- Instantiation of higher order variables is hard.

Future Work

- Prove some intermediate lemmas for `hoasinduction`
- Prove `induction2lem` by creating intermediate lemma.

Build induction into LEO

Theorem

(`induction2`) *Let Φ be a property such that the following hold:*

1. *For all $x \in Var$, x satisfies Φ.*
2. *For all $a, b \in T$, if a and b satisfy Φ, then $(a\,b)^{\downarrow\sigma}$ satisfies Φ.*
3. *For all $a \in T$, if $(a[b.id])^{\downarrow\sigma}$ satisfies Φ whenever $b \in T$ satisfies Φ, then $(\lambda a)^{\downarrow\sigma}$ satisfies Φ.*

Then for all $a \in T$, a satisfies Φ.

To use `induction2`, a theorem prover should:

1. Recognize `induction2` is an induction principle.
2. Choose an appropriate Φ.
3. Prove each of 1, 2 and 3 for this Φ.

Thank you

Martín Abadi, Luca Cardelli, Pierre-Louis Curien, and Jean-Jacques Lèvy.
Explicit substitutions.
In *Conference Record of the Seventeenth Annual ACM Symposium on Principles of Programming Languages, San Francisco, California*, pages 31–46. ACM, 1990.

Christoph Benzmüller, Florian Rabe, and Geoff Sutcliffe.
The core tptp language for classical higher-order logic.
In *Fourth International Joint Conference on Automated Reasoning (IJCAR'06)*, volume 5195 of *LNAI*. Springer, 2008.

Christoph Benzmüller, Frank Theiss, Larry Paulson, and Arnaud Fietzke.
LEO-II - a cooperative automatic theorem prover for higher-order logic.
In *Fourth International Joint Conference on Automated Reasoning (IJCAR'06)*, volume 5195 of *LNAI*. Springer, 2008.

Chad E. Brown.
M-set models.
In C. E. Benzmüller, C. E. Brown, J. Siekmann, and R. Statman, editors, *Reasoning in Simple Type Theory: Festschrift in Honor of Peter B. Andrews on His 70th Birthday*, Studies in Logic and the Foundations of Mathematics. IFCoLog, 2008.
To appear.

Gilles Dowek, Thérèse Hardin, and Claude Kirchner.
Higher-order unification via explicit substitutions.
In D. Kozen, editor, *Proceedings of the Tenth Annual Symposium on Logic in Computer Science*, pages 366–374, San Diego, California, June 1995. IEEE Computer Society Press.

Frank Pfenning and Conal Elliott.
Higher-order abstract syntax.
In *Proceedings of the ACM SIGPLAN '88 Symposium on Language Design and Implementation*, pages 199–208, Atlanta, Georgia, June 1988.