Scott's Representation Theorem and the Univalent Karoubi Envelope

Benedikt Ahrens

joint work with Arnoud van der Leer and Kobe Wullaert

ITP 2025

Outline

1 Overview of Our Work

2 Review: Univalent Foundations and Category Theory

3 Interpretation of Lambda Theories

4 A Tactic for Rewriting

5 Conclusion

Simply-Typed Lambda Calculi and Cartesian Closed Categories

$$\begin{tikzcd} \mathsf{STLC} \arrow[rr, bend left, "\text{interpretation}"] & & \mathsf{CCC} \arrow[ll, bend left, "\text{internal logic}"] \end{tikzcd}$$

Theorem (Joachim Lambek & Phil Scott)

“Interpretation” and “internal logic” form an equivalence of categories.

This Work: Untyped Lambda Calculus and Reflexive Objects

$$\begin{tikzcd} \mathsf{LamTh} \arrow[rr, bend left, "\text{interpretation}"] & & \mathsf{RO} \arrow[ll, bend left, "\text{endomorphism theory}"] \end{tikzcd}$$

Here, RO is the collection of Cartesian closed categories equipped with a reflexive object A:

$$\begin{tikzcd} A^A \arrow[r, "\lambda"] \arrow[rr, bend right, "1"'] & A \arrow[r, "\rho"] & A^A \end{tikzcd}$$

Theorem (Dana Scott 1980, Martin Hyland 2012)

$E : \mathsf{LamTh} \to \mathsf{RO}$ *has a section* $I : \mathsf{RO} \to \mathsf{LamTh}$*, that is, every lambda theory is the endomorphism theory of some reflexive object.*

Our Work

1. Formalization of the constructions of Scott and Hyland in UniMath, a computer-checked library of univalent mathematics in Rocq
2. Explanation of the difference between these constructions using notions of univalent foundations
3. Implementation of an extensible tactic for automated rewriting (not just) lambda terms, used in the formalization of Scott's construction

Outline

1. Overview of Our Work
2. Review: Univalent Foundations and Category Theory
3. Interpretation of Lambda Theories
4. A Tactic for Rewriting
5. Conclusion

Univalent Foundations

Setting

Martin-Löf type theory + Univalence Axiom

Homotopy Levels

Type X is

- a **set** if it satisfies Uniqueness of Identity Proofs
- a **groupoid** if $a = b$ is a set for any $a, b : X$

Univalence Axiom

$$(X = Y) \simeq (X \simeq Y)$$

Two Category Theories

Definition

A category $\mathbf{C}$ consists of

1. a type of objects $\mathbf{C}_0$
2. a family of **sets** $\mathbf{C}(a, b)$ for objects $a, b : \mathbf{C}_0$
3. composition, identity

$\mathbf{C}$ is called

- **set category** if $\mathbf{C}_0$ is a set
- **univalent category** if $(a = b) \simeq (a \cong b)$

Lemma

In a univalent category, the type of objects is a groupoid.

Set Category Theory vs Univalent Category Theory

Set Category Theory

- Constructions are invariant under **isomorphism** of categories

$$(\mathbf{C} = \mathbf{D}) \simeq (\mathbf{C} \cong \mathbf{D})$$

- Examples
  - Finite categories
  - Categories built from syntax

Univalent Category Theory

- Constructions are invariant under **equivalence** of categories

$$(\mathbf{C} = \mathbf{D}) \simeq (\mathbf{C} \simeq \mathbf{D})$$

- Examples
  - Category of sets, of groups, of rings, of set categories, ...
  - Functor category, if target category is univalent

The Rezk Completion

Theorem (Ahrens, Kapulkin, Shulman)

For any category $\mathbf{C}$*, there is a univalent category* $\mathbf{RC(C)}$ *and* $\eta : \mathbf{C} \to \mathbf{RC(C)}$ *such that any functor* $F : \mathbf{C} \to \mathbf{D}$ *with* $\mathbf{D}$ *univalent factors uniquely via* η.

$$\begin{tikzcd} \mathbf{C} \arrow[d, hook, "\eta"'] \arrow[dr, "\forall"] & \\ \mathbf{RC(C)} \arrow[r, "\exists!"'] & \mathbf{D} \end{tikzcd}$$

Provides a systematic way to turn a (set) category into a univalent one.

Outline

1. Overview of Our Work
2. Review: Univalent Foundations and Category Theory
3. Interpretation of Lambda Theories
4. A Tactic for Rewriting
5. Conclusion

Lambda Theories and Reflexive Objects

$$
\begin{tikzcd}
\mathsf{LamTh} \arrow[rr, bend left=40, "\text{Hyland}"] \arrow[rr, bend left=20, "\text{Scott}"'] & & \mathsf{RO} \arrow[ll, bend left=30, "\text{endomorphism theory}"]
\end{tikzcd}
$$

Remarks

- Constructions are taken from Hyland's "Classical Lambda Calculus in Modern Dress"
- Translation into univalent foundations mostly straightforward
- Relating Scott's and Hyland's constructions is our work

Lambda Theories

Definition (Algebraic Theory)

terms	sets L_n
variables	$x_{n,i} : L_n$
substitution	$(\_ \bullet \_) : L_m \times L_n^m \to L_n$
laws	$x_j \bullet g = g_j, \quad f \bullet (x_{l,i})_i = f \quad \text{and} \quad (f \bullet g) \bullet h = f \bullet (g_i \bullet h)_i$

Definition (Lambda Theory)

	algebraic theory L
abstraction	$\lambda_n : L_{n+1} \to L_n$
application	$\rho_n : L_n \to L_{n+1}$
laws	$\lambda_m(f) \bullet h = \lambda_n(f \bullet ((\iota_{n,1}(h_i))_i + (x_{n+1})))$ $\rho_n(g \bullet h) = \rho_m(g) \bullet ((\iota_{n,1}(h_i))_i + (x_{n+1}))$
β-equality	$\rho_n \circ \lambda_n = \mathrm{id}_{L_n}$

Endomorphism Theory of a Reflexive Object

Definition

Let $(X, \lambda : X^X \to X, \rho : X \to X^X)$ be a reflexive object in a Cartesian closed category $\mathbf{C}$.
The **endomorphism theory** $E(X)$ of X is the lambda theory given by

terms	$E(X)_n := \mathbf{C}(X^n, X)$
variables	$x_{n,i} : X^n \to X$
substitution	$f : X^m \to X$ and $g_1, \ldots, g_m : X^n \to X$ yield substitution $f \circ \langle g_i \rangle_i : X^n \to X$
abstraction	using λ
application	using ρ

Interpretation à la Scott

$$\mathsf{LamTh} \underset{E}{\overset{\text{Scott}}{\rightleftarrows}} \mathsf{RO}$$

Sketch of Scott's construction: given lambda theory L

- Build a category from L_o (terms in empty context):

  objects $A : L_o$ such that $A \circ A = A$

  morphisms $f : A \to B$ is $f : L_o$ such that $B \circ f \circ A = f$
- This category is Cartesian closed.
- The object $\lambda x.x$ is a reflexive object in this category.
- $E(\text{Scott}(L)) \cong L$

Interpretation à la Hyland

$$\begin{tikzcd} \mathsf{LamTh} \arrow[r, bend left, "\text{Hyland}"] & \mathsf{RO} \arrow[l, bend left, "\text{E}"] \end{tikzcd}$$

Sketch of Hyland's construction: given lambda theory L

- Consider the category of presheaves over the Lawvere theory induced by L.
- This category is Cartesian closed.
- The "theory presheaf" induced by L is a reflexive object in this category.
- $E(\text{Hyland}(L)) \cong L$

Relation Between Scott's and Hyland's Construction

	Scott	Hyland
Category	$\mathbf{R}$	$P\mathbf{L}$
Objects	Terms A s.t. $A \circ A = A$	Functors from $\mathbf{L}^{\mathrm{op}}$ to $\mathbf{SET}$
Reflexive object	$\lambda x.x$	Theory presheaf

Theorem

$$\begin{tikzcd} \mathbf{R} \arrow[rr, dash, "\sim"] & & \mathbf{K}(\mathbf{L}_o) \arrow[d, hook] & \textit{set categories} \\ P\mathbf{L} \arrow[r, dash, "\sim"] & P\mathbf{L}_o & \mathbf{K}'(\mathbf{L}_o) \arrow[l, hook'] & \textit{univalent categories} \end{tikzcd}$$

$\mathbf{K}$ and $\mathbf{K}'$ are the set Karoubi envelope and univalent Karoubi envelope, respectively, and $\mathbf{K}(\mathbf{L}_o) \hookrightarrow\!\!\!\rightarrow \mathbf{K}'(\mathbf{L}_o)$ *is the Rezk completion.*

Outline

1. Overview of Our Work
2. Review: Univalent Foundations and Category Theory
3. Interpretation of Lambda Theories
4. A Tactic for Rewriting
5. Conclusion

Challenge: Rewriting Lambda Terms

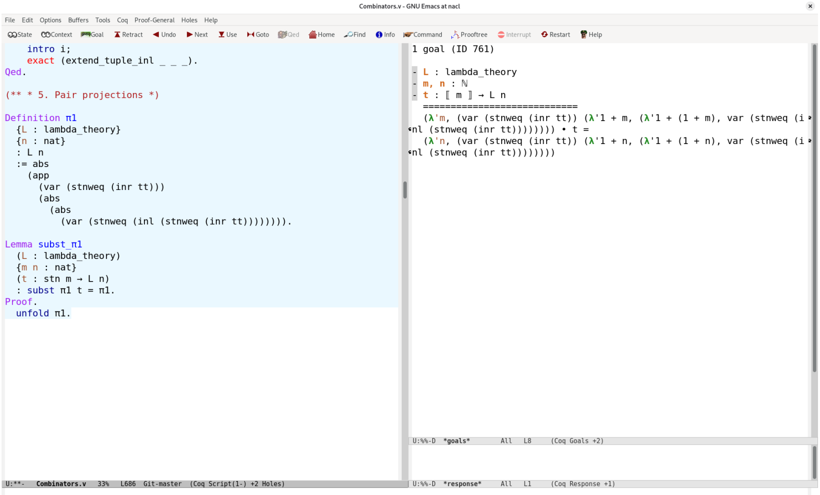
Challenge

- Scott's construction involves reasoning about elements of a lambda theory
- Manual rewriting is tedious
- Automation is avoided in UniMath

Solution

An automated tactic `propagate_subst` that produces a proof script with a sequence of `refine` tactics.

Preparing the Goal



Running the Tactic

Combinators.v - GNU Emacs at nacl

File Edit Options Buffers Tools Coq Proof-General Holes Help

State Context Goal Retract Undo Next Use Goto Qed Home Find Info Command Prooftree Interrupt Restart Help

```
    intro i;
    exact (extend_tuple_inl _ _ _).
Qed.

(** * 5. Pair projections *)

Definition π1
  {L : lambda_theory}
  {n : nat}
  : L n
  := abs
    (app
      (var (stnweq (inr tt)))
      (abs
        (abs
          (var (stnweq (inl (stnweq (inr tt)))))))).

Lemma subst_π1
  (L : lambda_theory)
  {m n : nat}
  (t : stn m → L n)
  : subst π1 t = π1.
Proof.
  unfold π1.
  propagate_subst ().
```

U:**- **Combinators.v** 33% L686 Git-master (Coq Script(1-) +2 Holes)

```
1 goal (ID 2340)

- L : lambda_theory
- m, n : ℕ
- t : ⟦ m ⟧ → L n
  ============================
  (λ'n, (var (stnweq (inr tt)) (λ'S n, (λ'S (S n), var (stnweq (inl (st
nweq (inr tt)))))))) =
  (λ'n, (var (stnweq (inr tt)) (λ'S n, (λ'S (S n), var (stnweq (inl (st
nweq (inr tt))))))))
```

U:%%-D ***goals*** All L8 (Coq Goals +2)

```
refine '(subst_abs _ _ _ @ _).
refine '(maponpaths (λ x, (abs x)) (subst_app _ _ _ _) @ _).
refine '(maponpaths (λ x, (abs (app x _))) (var_subst _ _ _) @ _).
refine '(maponpaths (λ x, (abs (app _ x))) (subst_abs _ _ _) @ _).
refine '(maponpaths (λ x, (abs (app x _))) (extend_tuple_inr _ _ _) @ _)
refine '(maponpaths (λ x, (abs (app _ (abs x)))) (subst_abs _ _ _) @ _).
refine '(maponpaths (λ x, (abs (app _ (abs (abs x))))) (var_subst _ _ _)
refine '(maponpaths (λ x, (abs (app _ (abs (abs x))))) (extend_tuple_inl
refine '(maponpaths (λ x, (abs (app _ (abs (abs (inflate x)))))) (extenc
refine '(maponpaths (λ x, (abs (app _ (abs (abs x))))) (inflate_var _ _)
```

U:%%-D ***response*** All L10 (Coq Response +2)

Replacing the Tactic with the Generated Proof Script

Combinators.v - GNU Emacs at nacl

File Edit Options Buffers Tools Coq Proof-General Holes Help

State Context Goal Retract Undo Next Use Goto Qed Home Find Info Command Prooftree Interrupt Restart Help

```
        (var (stnweq (inl (stnweq (inr tt)))))))).

Lemma subst_π1
  (L : lambda_theory)
  {m n : nat}
  (t : stn m → L n)
  : subst π1 t = π1.
Proof.
  unfold π1.
  (* propagate_subst (). *)
  refine '(subst_abs _ _ _ @ _).
  refine '(maponpaths (λ x, (abs x)) (subst_app _ _ _ _) @ _).
  refine '(maponpaths (λ x, (abs (app x _))) (var_subst _ _ _) @ _).
  refine '(maponpaths (λ x, (abs (app _ x))) (subst_abs _ _ _) @ _).
  refine '(maponpaths (λ x, (abs (app x _))) (extend_tuple_inr _ _ _) @ _).
  refine '(maponpaths (λ x, (abs (app _ (abs x)))) (subst_abs _ _ _) @ _).
  refine '(maponpaths (λ x, (abs (app _ (abs (abs x))))) (var_subst _ _ _) @ _).
  refine '(maponpaths (λ x, (abs (app _ (abs (abs x))))) (extend_tuple_inl _ _ _) @ _).
  refine '(maponpaths (λ x, (abs (app _ (abs (abs (inflate x)))))) (extend_tuple_inr _ _ _) @ _).
  refine '(maponpaths (λ x, (abs (app _ (abs (abs x))))) (inflate_var _ _) @ _).
  apply idpath.
Qed.
```

```
1 goal (ID 1081)

- L : lambda_theory
- m, n : ℕ
- t : ⟦ m ⟧ → L n
  ============================
  (λ'n, (var (stnweq (inr tt)) (λ'S n, (λ'S (S n), var (stnweq (inl (stnweq (inr tt)))))))) =
  (λ'n, (var (stnweq (inr tt)) (λ'1 + n, (λ'1 + (1 + n), var (stnweq (inl (stnweq (inr tt))))))))
```

U:%%-D *goals* All L8 (Coq Goals +2)

U:%%-D *response* All L1 (Coq Response +2)

U:--- Combinators.v 33% L697 Git:master (Coq Script(1-) +2 Holes)

An Extensible Tactic for Rewriting

Benefits

fast	Sequence of `refine` tactics runs in a fraction of the time that proof search takes
modular	Built from tactics, e.g., generic traversal of terms
extensible	Can add more language constructions and equations governing their behavior after initial setup
generalizable	A variant has been applied in domain of hyperdoctrines

Things to Improve

usage	Awkward copy-pasting
not complete	Not all goals are solved entirely

Outline

1. Overview of Our Work
2. Review: Univalent Foundations and Category Theory
3. Interpretation of Lambda Theories
4. A Tactic for Rewriting
5. Conclusion

About the Formalization

- Integrated into UniMath, a library of univalent mathematics
- $> 11,000$ loc, for instance:
  - Basic categories involved in Scott's and Hyland's constructions (3,000)
  - Reasoning about lambda terms (3,000)
  - Karoubi envelope (2,000)
  - Examples of algebraic theories (1,000)
  - Diagram relating different constructions (1,000)
  - ...
- Use of displayed categories for building complicated categories layer-wise

Summary

- $$\begin{tikzcd} \mathsf{LamTh} \arrow[rr, bend left=40, "\text{Hyland}"] \arrow[rr, bend left=20, "\text{Scott}"'] & & \mathsf{RO} \arrow[ll, bend left=30, "\text{endomorphism theory}"] \end{tikzcd}$$

- Intuition of difference between Scott and Hyland,

  “syntactic” vs “categorical”

  can be formalized, to some extent, in UF as

  “set categorical” vs ”univalent categorical”

- A tactic for rewriting

Summary

- $$\begin{tikzcd}
\mathsf{LamTh} \arrow[rr, bend left=40, "\text{Hyland}"] \arrow[rr, bend left=20, "\text{Scott}"'] & & \mathsf{RO} \arrow[ll, bend left=30, "\text{endomorphism theory}"]
\end{tikzcd}$$

- Intuition of difference between Scott and Hyland,

  “syntactic” vs “categorical”

  can be formalized, to some extent, in UF as

  “set categorical” vs ”univalent categorical”

- A tactic for rewriting

Thanks for your attention!