Semantics for the λ-calculus

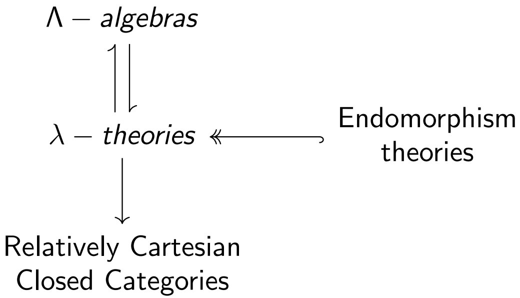
Arnoud van der Leer
Delft University of Technology
arnoudvanderleer@gmail.com

2023-11-02

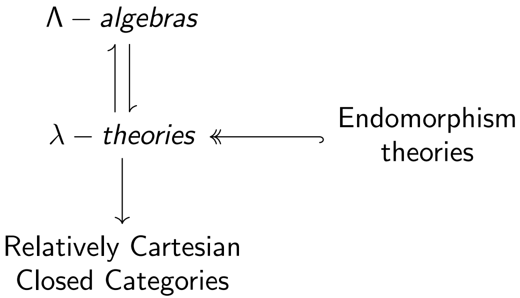
Classical lambda calculus in modern dress

- Paper by Martin Hyland.
- About models for the λ-calculus.
- Three 'big' theorems.
- My job: 'annotate'.

- Intro
- Talking about the λ-calculus
  - Models
  - Semantics
- The main theorems
  - Scott's representation theorem
  - The fundamental theorem of the λ-calculus
  - The category of retracts
- My contribution
  - Annotating the paper
  - Mechanization
- Conclusion



Intro
Talking about the λ-calculus
Models
Semantics
The main theorems
Scott's representation theorem
The fundamental theorem of the λ-calculus
The category of retracts
My contribution
Annotating the paper
Mechanization
Conclusion



The **untyped** λ-calculus

Describes a collection consisting of (only) functions.

The **untyped** λ-calculus

Describes a collection consisting of (only) functions.

Has terms, consisting of *variables*, *application* and *abstraction*:

$$x_1$$

$$x_1(x_2 x_1)$$

$$\lambda x_1, x_1$$

$$\lambda x_3 x_2 x_1, x_1(x_2 x_3).$$

Can have β- and η-equality:

$$(\lambda x_n, f)g = f[x_n := g] \qquad \lambda x_n, (f x_n) = f.$$

The **untyped** λ-calculus

Describes a collection consisting of (only) functions.

Has terms, consisting of *variables*, *application* and *abstraction*:

$$x_1$$

$$x_1(x_2x_1)$$

$$\lambda x_1, x_1$$

$$\lambda x_3x_2x_1, x_1(x_2x_3).$$

Can have β- and η-equality:

$$(\lambda x_n, f)g = f[x_n := g] \qquad \lambda x_n, (fx_n) = f.$$

The *(pure) λ-calculus*: Described exactly by the above.

Algebraic theories: objects with variables and substitution

Example

λ-calculus: $\Lambda_n = \{(\lambda x_1, x_1), x_5, (\lambda x_3, x_7) x_{42}\}$.

Algebraic theories: objects with variables and substitution

Example

λ-calculus: $\Lambda_n = \{(\lambda x_1, x_1), x_5, (\lambda x_3, x_7)x_{42}\}$.

Example

Polynomial ring: $\mathbb{Z}[x_1, ..., x_n] = \{1, x_3, 2048 + 7x_1^{37} - x_6 x_{13}^{42} x_{17}^{1729}, \ldots\}$.

Algebraic theories: objects with variables and substitution

Example

λ-calculus: $\Lambda_n = \{(\lambda x_1, x_1), x_5, (\lambda x_3, x_7)x_{42}\}$.

Example

Polynomial ring: $\mathbb{Z}[x_1, ..., x_n] = \{1, x_3, 2048 + 7x_1^{37} - x_6 x_{13}^{42} x_{17}^{1729}, \dots\}$.

Definition

An *algebraic theory* T is a sequence of sets T_n with variables $x_{i,n} \in T_n$ (for $0 \leq i < n$) and a substitution operation $\bullet : T_m \times T_n^m \to T_n$.

λ-theory: structure with app and abs

Definition

A λ-*theory* L is an algebraic theory, together with abstraction functions $\lambda : L_{n+1} \to L_n$ and application functions $\rho : L_n \to L_{n+1}$ (both compatible with the substitution).

λ-theory: structure with app and abs

Definition

A λ-*theory* L is an algebraic theory, together with abstraction functions $\lambda : L_{n+1} \to L_n$ and application functions $\rho : L_n \to L_{n+1}$ (both compatible with the substitution).

The pure λ-calculus Λ is the initial λ-theory.

λ-theory: structure with app and abs

Definition

A *λ-theory* L is an algebraic theory, together with abstraction functions $\lambda : L_{n+1} \to L_n$ and application functions $\rho : L_n \to L_{n+1}$ (both compatible with the substitution).

The pure λ-calculus Λ is the initial λ-theory.

β- and η-equality:

$$\rho_n \circ \lambda_n = \mathrm{Id}_{L_{n+1}} \qquad \lambda_n \circ \rho_n = \mathrm{Id}_{L_n}.$$

Algebras: Interpretations (or denotations)

We want to interpret terms with free variables as functions from a context to a set

Example

In $T(n) = \mathbb{Z}[x_1, \ldots, x_n]$, we can take a set $A = \mathbb{Q}$ and get

$$2x_1 + 3x_1^2 x_2 : A^2 \to A, \quad (a_1, a_2) \mapsto 2 \cdot a_1 + 3 \cdot a_1^2 \cdot a_2.$$

Algebras: Interpretations (or denotations)

We want to interpret terms with free variables as functions from a context to a set

Example

In $T(n) = \mathbb{Z}[x_1, \ldots, x_n]$, we can take a set $A = \mathbb{Q}$ and get

$$2x_1 + 3x_1^2 x_2 : A^2 \to A, \quad (a_1, a_2) \mapsto 2 \cdot a_1 + 3 \cdot a_1^2 \cdot a_2.$$

Definition

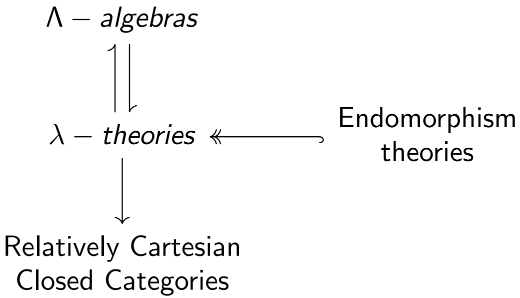
For an algebraic theory T, a *T-algebra* A is a set A, together with interpretation functions $T_n \times A^n \to A$ for all n (respecting the variables and substitution).

Intro
Talking about the λ-calculus
- Models
- Semantics

The main theorems
- Scott's representation theorem
- The fundamental theorem of the λ-calculus
- The category of retracts

My contribution
- Annotating the paper
- Mechanization

Conclusion



Scott's representation theorem (1980)

For every λ*-theory* L*, we can find a category* C *and an object* $X : C_0$*, such that* L *is isomorphic to the endomorphism theory of* X*: the* λ*-theory* $E(X)$ *given by* $E(X)_n = X^n \to X$*.*

Scott's representation theorem (1980)

For every λ-theory L, we can find a category C and an object $X : C_0$, such that L is isomorphic to the endomorphism theory of X: the λ-theory $E(X)$ given by $E(X)_n = X^n \to X$.

The variables of $E(X)_n$ are the projections $\pi_i : X^n \to X$. Also, substituting $g_1, \dots, g_m : X^n \to X$ into $f : X^m \to X$ composes f with $\langle g_1, \dots, g_m \rangle : X^n \to X^m$.

Scott's representation theorem (1980)

For every λ-theory L, we can find a category C and an object $X : C_0$, such that L is isomorphic to the endomorphism theory of X: the λ-theory $E(X)$ given by $E(X)_n = X^n \to X$.

The variables of $E(X)_n$ are the projections $\pi_i : X^n \to X$. Also, substituting $g_1, \dots, g_m : X^n \to X$ into $f : X^m \to X$ composes f with $\langle g_1, \dots, g_m \rangle : X^n \to X^m$.

We obtain $\lambda : E(X)_{n+1} \to E(X)_n$ as

$$\lambda : E(X)_{n+1} = (X^{n+1} \to X) \simeq (X^n \to X^X) \xrightarrow{\overline{abs} \circ -} (X^n \to X) = E(X)_n.$$

for some morphism $\overline{abs} : X^X \to X$. In the same way, we get $\rho : E(X)_n \to E(X)_{n+1}$ from a morphism $\overline{app} : X \to X^X$.

Scott's representation theorem (1980)

For every λ-theory L, we can find a category C and an object $X : C_0$, such that L is isomorphic to the endomorphism theory of X: the λ-theory $E(X)$ given by $E(X)_n = X^n \to X$.

C is the category of sequences of sets $(P_i)_i$ with a composition $P_m \times L_n^m \to P_n$ and X is the sequence $(L_i)_i$.

Scott's representation theorem (1980)

For every λ-theory L, we can find a category C and an object $X : C_0$, such that L is isomorphic to the endomorphism theory of X: the λ-theory $E(X)$ given by $E(X)_n = X^n \to X$.

C is the category of sequences of sets $(P_i)_i$ with a composition $P_m \times L_n^m \to P_n$ and X is the sequence $(L_i)_i$.

With Hyland's definitions and some lemmas, the representation theorem arises before you know it *(on paper)*.

Intro

Talking about the λ-calculus

- Models
- Semantics

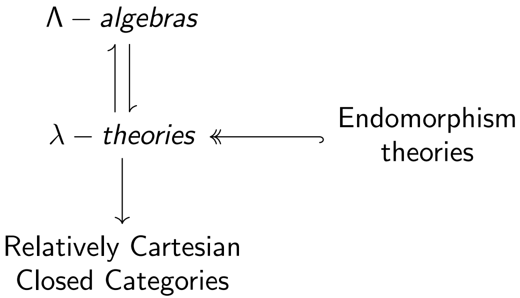
The main theorems

- Scott's representation theorem
- The fundamental theorem of the λ-calculus
- The category of retracts

My contribution

- Annotating the paper
- Mechanization

Conclusion



“The fundamental theorem of the λ-Calculus”

There is a functor from λ-theories to Λ-algebras, sending L to L_0: its set of constants.

“The fundamental theorem of the λ-Calculus”

There is a functor from λ-theories to Λ-algebras, sending L to L_0: its set of constants.

There is also a functor from Λ-algebras to λ-theories. This functor again uses the endomorphism theory $E(X)$ for some object X to construct the λ-theory.

“The fundamental theorem of the λ-Calculus”

There is a functor from λ-theories to Λ-algebras, sending L to L_0: its set of constants.

There is also a functor from Λ-algebras to λ-theories. This functor again uses the endomorphism theory $E(X)$ for some object X to construct the λ-theory.

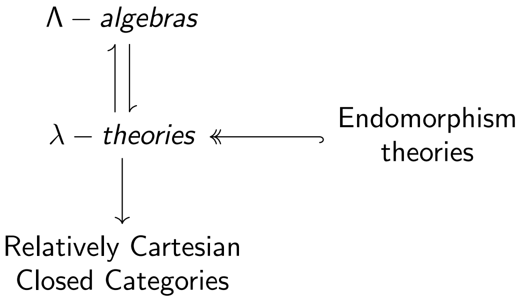
Hyland shows that these functors constitute an adjoint equivalence.

Intro
Talking about the λ-calculus
- Models
- Semantics

The main theorems
- Scott's representation theorem
- The fundamental theorem of the λ-calculus
- The category of retracts

My contribution
- Annotating the paper
- Mechanization

Conclusion



The category of retracts

Given a λ-theory L, we can view elements $f : L_1$ as one-argument functions, and we can compose them like $f \circ g := f \bullet g$.

Now we construct a category R

$$R_0 = \{a : L_1 \mid a \circ a = a\}, \qquad a \to b = \{f : L_1 \mid b \circ f \circ a = f\}.$$

The category of retracts

Given a λ-theory L, we can view elements $f : L_1$ as one-argument functions, and we can compose them like $f \circ g := f \bullet g$.

Now we construct a category R

$$R_0 = \{a : L_1 \mid a \circ a = a\}, \qquad a \to b = \{f : L_1 \mid b \circ f \circ a = f\}.$$

This category is cartesian closed: it has products, and 'exponentials'. So its morphisms constitute a simply typed λ-calculus: we can do *type theory* with the morphisms.

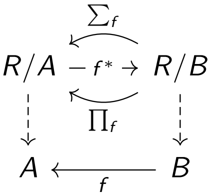
Relatively cartesian closed

This category is cartesian closed: it has products, and 'exponentials'. So its morphisms constitute a simply typed λ-calculus: we can do *type theory* with the morphisms.

Relatively cartesian closed

This category is cartesian closed: it has products, and ‘exponentials’. So its morphisms constitute a simply typed λ-calculus: we can do *type theory* with the morphisms.

If we want to do *dependent type theory*, we need dependent products and sums.

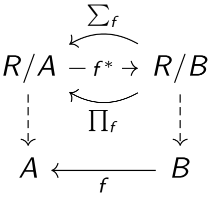


Locally cartesian closed: all pullback functors have both adjoints.

Relatively cartesian closed

This category is cartesian closed: it has products, and ‘exponentials’. So its morphisms constitute a simply typed λ-calculus: we can do *type theory* with the morphisms.

If we want to do *dependent type theory*, we need dependent products and sums.



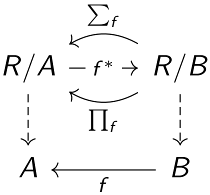
Locally cartesian closed: all pullback functors have both adjoints.

In R, not all pullback functors have both adjoints, but some do: *relatively cartesian closed*.

Relatively cartesian closed

This category is cartesian closed: it has products, and 'exponentials'. So its morphisms constitute a simply typed λ-calculus: we can do *type theory* with the morphisms.

If we want to do *dependent type theory*, we need dependent products and sums.

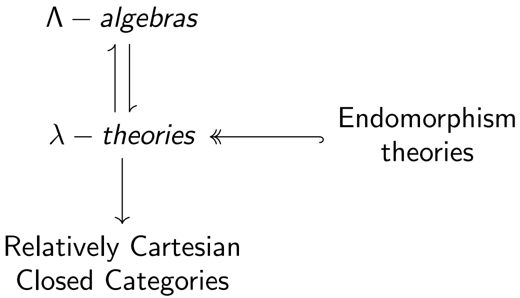


Locally cartesian closed: all pullback functors have both adjoints.

In R, not all pullback functors have both adjoints, but some do: *relatively cartesian closed*.

I am still working on understanding the proof.

Intro
Talking about the λ-calculus
Models
Semantics
The main theorems
Scott's representation theorem
The fundamental theorem of the λ-calculus
The category of retracts
My contribution
Annotating the paper
Mechanization
Conclusion



Annotating the paper

An algebraic theory T is first a functor $T : \mathbf{F} \to Sets$: so we have sets $T(n)$ of n-ary multimaps with variable renamings. In addition, T is equipped with projections $pr_1, \ldots, pr_n : T(n)$ including as special case the identity $id \in T(1)$. Finally there are compositions $T(n) \times T(m)^n \to T(m)$ which are **associative, unital, compatible with projections and natural in n and m**. *A map $F : S \to T$ of algebraic theories is a natural transformation with components $F_n : S(n) \to T(n)$ preserving projections and composition.*

Annotating the paper

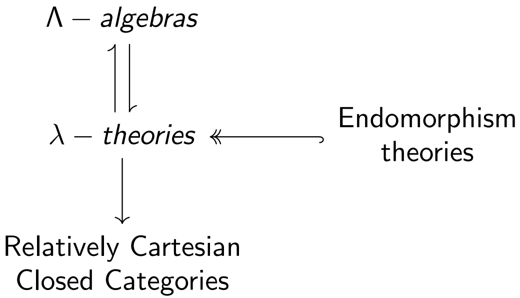
An algebraic theory T *is first a functor* $T : \mathbf{F} \to \mathit{Sets}$*: so we have sets* $T(n)$ *of* n*-ary multimaps with variable renamings. In addition,* T *is equipped with projections* $pr_1, \ldots, pr_n : T(n)$ *including as special case the identity* $id \in T(1)$*. Finally there are compositions* $T(n) \times T(m)^n \to T(m)$ *which are* **associative, unital, compatible with projections and natural in** n **and** m*. A map* $F : S \to T$ *of algebraic theories is a natural transformation with components* $F_n : S(n) \to T(n)$ *preserving projections and composition.*

- Learn the background.
- Decode the definitions and theorems.
- Find examples.
- Formalize (on paper).
- Mechanize.

Mechanization

- Displayed categories:
  - Univalence;
  - Limits (twice);
- Higher inductive types;
- $X^{n+1} = X \times X^n$;
- $X_{n+1} = X_{1+n}$;

Intro
Talking about the λ-calculus
Models
Semantics
The main theorems
Scott's representation theorem
The fundamental theorem of the λ-calculus
The category of retracts
My contribution
Annotating the paper
Mechanization
Conclusion



Conclusion

Algebraic theories, λ-theories and their algebras (and ‘presheaves’) seem to be a promising way to work with models for the λ-calculus.

Conclusion

Algebraic theories, λ-theories and their algebras (and 'presheaves') seem to be a promising way to work with models for the λ-calculus.

3 'big' theorems:

- Every model of the λ-calculus arises as the endomorphism theory of some category.
- There is an equivalence between models of the λ-calculus, and interpretations of the λ-calculus as functions on a set.
- From a model for the untyped λ-calculus, we can create a category in which we can do some form of dependent type theory.

Conclusion

Algebraic theories, λ-theories and their algebras (and 'presheaves') seem to be a promising way to work with models for the λ-calculus.

3 'big' theorems:

- Every model of the λ-calculus arises as the endomorphism theory of some category.
- There is an equivalence between models of the λ-calculus, and interpretations of the λ-calculus as functions on a set.
- From a model for the untyped λ-calculus, we can create a category in which we can do some form of dependent type theory.

I am slowly processing the paper.

Conclusion

Algebraic theories, λ-theories and their algebras (and 'presheaves') seem to be a promising way to work with models for the λ-calculus.

3 'big' theorems:

- Every model of the λ-calculus arises as the endomorphism theory of some category.
- There is an equivalence between models of the λ-calculus, and interpretations of the λ-calculus as functions on a set.
- From a model for the untyped λ-calculus, we can create a category in which we can do some form of dependent type theory.

I am slowly processing the paper.

Mechanization is hard.

Do you have questions?

Do you have questions?

Because I have one: I am still a bit unsure about the exact ‘meaning’ of relative cartesian closedness. Can someone explain that better to me?