Classical Lambda Calculus in Modern Dress

April 18, 2023

Denotational Semantics

```
expr = 1 | -<expr> | <expr> + <expr>
```

Denotational Semantics

`expr = 1 | -<expr> | <expr> + <expr>`

For $t \in \texttt{expr}$, $[\![t]\!] \in \mathbb{Z}$.

Denotational Semantics

```
expr = 1 | x | -<expr> | <expr> + <expr>
```

Denotational Semantics

```
expr = 1 | x | -<expr> | <expr> + <expr>
```

For $t \in$ `expr`, $[\![t]\!] \in \mathbb{Z} \oplus x \cdot \mathbb{Z}$.

Denotational Semantics

```
expr = 1 | x * <expr> | -<expr> | <expr> + <expr>
```

Denotational Semantics

```
expr = 1 | x * <expr> | -<expr> | <expr> + <expr>
```

For $t \in \texttt{expr}$, $[\![t]\!] \in \mathbb{Z}[x]$: polynomials with coefficients in $\mathbb{Z}$.

Denotational Semantics

```
expr = 1 | x * <expr> | -<expr> | <expr> + <expr>
```

For $t \in \texttt{expr}$, $[\![t]\!] \in \mathbb{Z}[x]$: polynomials with coefficients in $\mathbb{Z}$. Or $[\![t]\!] \in \mathbb{Z} \to \mathbb{Z}$. Or maybe $[\![t]\!] \in \mathbb{R} \to \mathbb{R}$.

Denotational Semantics

```
expr = Var(<nat>) | App(<expr>, <expr>) | Abs(Var(<nat>), <expr>)
```

Denotational Semantics

```
expr = Var(<nat>) | App(<expr>, <expr>) | Abs(Var(<nat>), <expr>)
```

For $t \in \texttt{expr}$, $[\![t]\!] \in ?$.

Algebraic Theory

Definition

An algebraic theory consists of

- a functor $\mathcal{T} : F \to \mathrm{SET}$,
- elements (projections) $x_{n,i} \in \mathcal{T}(n)$ for all $1 \leq i \leq n$,
- a composition morphism $\bullet : \mathcal{T}(m) \times \mathcal{T}(n)^m \to \mathcal{T}(n)$ for all m, n.

The composition must be associative, unital, compatible with projections and dinatural in m.

Abstract Clone

Definition

An abstract clone is

- a function $C : \mathbb{N} \to \mathrm{SET}$,
- a composition morphism $\bullet : C(m) \times C(n)^m \to C(n)$ for all m, n,
- elements (projections) $x_{n,i} \in C(n)$ for all $1 \leq i \leq n$.

The composition must satisfy $x_{n,i} \bullet (f_1, \ldots, f_n) = f_i$ and $f \bullet (\pi_{1,n}, \ldots, \pi_{n,n}) = f$ and

$$(f \bullet (g_1, \ldots, g_m)) \bullet (h_1, \ldots, h_n) = f \bullet (g_1 \bullet (h_1, \ldots, h_n), \ldots, g_m \bullet (h_1, \ldots, h_n)).$$

Abstract Clone - Example

Definition

An abstract clone is

- a function $C : \mathbb{N} \to \mathrm{SET}$,
- a composition morphism $\bullet : C(m) \times C(n)^m \to C(n)$ for all m, n,
- elements (projections) $x_{n,i} \in C(n)$ for all $1 \leq i \leq n$.

The composition must satisfy $x_{n,i} \bullet (f_1, \ldots, f_n) = f_i$ and $f \bullet (\pi_{1,n}, \ldots, \pi_{n,n}) = f$ and

$$(f \bullet (g_1, \ldots, g_m)) \bullet (h_1, \ldots, h_n) = f \bullet (g_1 \bullet (h_1, \ldots, h_n), \ldots, g_m \bullet (h_1, \ldots, h_n)).$$

Example

Take $C(n) = \{1, 2, \ldots, n\}$, $x_{n,i} = i$ and $i \bullet (f_1, \ldots, f_n) = f_i$.

Abstract Clone - Example

Definition

An abstract clone is

- a function $C : \mathbb{N} \to \mathrm{SET}$,
- a composition morphism $\bullet : C(m) \times C(n)^m \to C(n)$ for all m, n,
- elements (projections) $x_{n,i} \in C(n)$ for all $1 \leq i \leq n$.

The composition must satisfy $x_{n,i} \bullet (f_1, \ldots, f_n) = f_i$ and $f \bullet (\pi_{1,n}, \ldots, \pi_{n,n}) = f$ and

$$(f \bullet (g_1, \ldots, g_m)) \bullet (h_1, \ldots, h_n) = f \bullet (g_1 \bullet (h_1, \ldots, h_n), \ldots, g_m \bullet (h_1, \ldots, h_n)).$$

Example

Take $C(n)$ the free monoid on n generators. The $x_{n,i}$ are the generators and $f \bullet (g_1, \ldots, g_m)$ applies the mapping $C(m) \to C(n)$ on f, given by sending $x_{m,i}$ to g_i.

Abstract Clone - Example

Definition

An abstract clone is

- a function $C : \mathbb{N} \to \mathrm{SET}$,
- a composition morphism $\bullet : C(m) \times C(n)^m \to C(n)$ for all m, n,
- elements (projections) $x_{n,i} \in C(n)$ for all $1 \leq i \leq n$.

The composition must satisfy $x_{n,i} \bullet (f_1, \dots, f_n) = f_i$ and $f \bullet (\pi_{1,n}, \dots, \pi_{n,n}) = f$ and

$$(f \bullet (g_1, \dots, g_m)) \bullet (h_1, \dots, h_n) = f \bullet (g_1 \bullet (h_1, \dots, h_n), \dots, g_m \bullet (h_1, \dots, h_n)).$$

Example

Let X be an object in a category with finite products. The endomorphism clone has $C(n) = (X^n \to X)$. Then $x_{n,i}$ is the ith projection morphism. The universal property of the product gives $\bullet$.

Abstract Clone - Example

Definition

An abstract clone is

- a function $C : \mathbb{N} \to \mathrm{SET}$,
- a composition morphism $\bullet : C(m) \times C(n)^m \to C(n)$ for all m, n,
- elements (projections) $x_{n,i} \in C(n)$ for all $1 \leq i \leq n$.

The composition must satisfy $x_{n,i} \bullet (f_1, \ldots, f_n) = f_i$ and $f \bullet (\pi_{1,n}, \ldots, \pi_{n,n}) = f$ and

$$(f \bullet (g_1, \ldots, g_m)) \bullet (h_1, \ldots, h_n) = f \bullet (g_1 \bullet (h_1, \ldots, h_n), \ldots, g_m \bullet (h_1, \ldots, h_n)).$$

Example

The λ-calculus Λ, in which $\Lambda(n)$ consists of the terms with n free variables, $x_{n,i} = \mathtt{Var}(i)$ (with De Bruijn indices) and $\bullet$ is substitution.

Abstract Clone Algebra

Definition

An abstract clone is

- a function $C : \mathbb{N} \to \mathrm{SET}$,
- a composition morphism $\bullet : C(m) \times C(n)^m \to C(n)$ for all m, n,
- elements (projections) $x_{n,i} \in C(n)$ for all $1 \leq i \leq n$.

The composition must satisfy $x_{n,i} \bullet (f_1, \ldots, f_n) = f_i$ and $f \bullet (\pi_{1,n}, \ldots, \pi_{n,n}) = f$ and

$$(f \bullet (g_1, \ldots, g_m)) \bullet (h_1, \ldots, h_n) = f \bullet (g_1 \bullet (h_1, \ldots, h_n), \ldots, g_m \bullet (h_1, \ldots, h_n)).$$

Abstract Clone Algebra - Example

Definition

An abstract clone is

- a function $C : \mathbb{N} \to \mathrm{SET}$,
- a composition morphism $\bullet : C(m) \times C(n)^m \to C(n)$ for all m, n,
- elements (projections) $x_{n,i} \in C(n)$ for all $1 \leq i \leq n$.

The composition must satisfy $x_{n,i} \bullet (f_1, \dots, f_n) = f_i$ and $f \bullet (\pi_{1,n}, \dots, \pi_{n,n}) = f$ and

$$(f \bullet (g_1, \dots, g_m)) \bullet (h_1, \dots, h_n) = f \bullet (g_1 \bullet (h_1, \dots, h_n), \dots, g_m \bullet (h_1, \dots, h_n)).$$

Example

For the clone $C(n) = \{1, \dots, n\}$, any set A can be an algebra, setting $\alpha_n(i, a) = a_i$.

Abstract Clone Algebra - Example

Definition

An abstract clone is

- a function $C : \mathbb{N} \to \mathrm{SET}$,
- a composition morphism $\bullet : C(m) \times C(n)^m \to C(n)$ for all m, n,
- elements (projections) $x_{n,i} \in C(n)$ for all $1 \leq i \leq n$.

The composition must satisfy $x_{n,i} \bullet (f_1, \ldots, f_n) = f_i$ and $f \bullet (\pi_{1,n}, \ldots, \pi_{n,n}) = f$ and

$$(f \bullet (g_1, \ldots, g_m)) \bullet (h_1, \ldots, h_n) = f \bullet (g_1 \bullet (h_1, \ldots, h_n), \ldots, g_m \bullet (h_1, \ldots, h_n)).$$

Example

For the clone with $C(n)$ the free monoid on n generators, the algebras are exactly the monoids. Setting $a \star b := \alpha_2(x_1 \star x_2, (a, b))$.

Abstract Clone Algebra - Properties

Example

An algebra A for the lambda calculus clone Λ gets a lot of structure. For each term $t \in \Lambda(n)$ and $a \in A^n$, we have an interpretation $t(a) \in A$.

Abstract Clone Algebra - Properties

Example

An algebra A for the lambda calculus clone Λ gets a lot of structure. For each term $t \in \Lambda(n)$ and $a \in A^n$, we have an interpretation $t(a) \in A$.
A has function application $ab := \alpha_2(x_1 x_2, (a, b))$.

Abstract Clone Algebra - Properties

Example

An algebra A for the lambda calculus clone Λ gets a lot of structure. For each term $t \in \Lambda(n)$ and $a \in A^n$, we have an interpretation $t(a) \in A$.
A has function application $ab := \alpha_2(x_1 x_2, (a, b))$.
And lambda abstraction $\lambda x, ax := \alpha_1((\lambda x, x_1 x), (a))$.

Abstract Clone Algebra - Properties

Example

An algebra A for the lambda calculus clone Λ gets a lot of structure. For each term $t \in \Lambda(n)$ and $a \in A^n$, we have an interpretation $t(a) \in A$.
A has function application $ab := \alpha_2(x_1x_2, (a, b))$.
And lambda abstraction $\lambda x, ax := \alpha_1((\lambda x, x_1x), (a))$.
It can inherit beta and eta reduction:

$$
\begin{aligned}
(\lambda x, ax)(b) &= \alpha_2(x_1x_2, (\alpha_1((\lambda x, x_1x), (a)), b)) \\
&= \alpha_2(x_1x_2, (\alpha_2((\lambda x, x_1x), (a, b)), \alpha_2(x_2, (a, b)))) \\
&= \alpha_2((x_1x_2) \bullet ((\lambda x, x_1x), x_2), (a, b)) \\
&= \alpha_2((\lambda x, x_1x)(x_2), (a, b)) \\
&= \alpha_2(x_1x_2, (a, b)) \\
&= ab.
\end{aligned}
$$

λ-clone

Definition

A λ-clone consists of

- an abstract clone $\mathcal{L}$,
- functions $\rho_n : \mathcal{L}(n) \to \mathcal{L}(n+1)$ and $\lambda_n : \mathcal{L}(n+1) \to \mathcal{L}(n)$.

Such that

$$\rho(f \bullet g) = \rho(f) \bullet (g_1, \dots, g_m, x_{n+1,n+1})$$

and

$$\lambda(f) \bullet g = \lambda(f \bullet (g_1, \dots, g_m, x_{n+1,n+1})).$$

Hyland also requires (λ, ρ) to be a section-retraction pair.

Given a λ-clone $\mathcal{L}$, we can interpret a term t of the lambda calculus (that has a context Γ of length n) as an element $[\![t]\!] \in \mathcal{L}(n)$.

λ-clone - Example

Definition

A λ-clone consists of

- an abstract clone $\mathcal{L}$,
- functions $\rho_n : \mathcal{L}(n) \to \mathcal{L}(n+1)$ and $\lambda_n : \mathcal{L}(n+1) \to \mathcal{L}(n)$.

Such that

$$\rho(f \bullet g) = \rho(f) \bullet (g_1, \dots, g_m, x_{n+1,n+1})$$

and

$$\lambda(f) \bullet g = \lambda(f \bullet (g_1, \dots, g_m, x_{n+1,n+1})).$$

Hyland also requires (λ, ρ) to be a section-retraction pair.

Example

If we have an object X in a category with products, and a retraction $X \to (X \to X)$, the endomorphism clone of X is a λ-clone.

λ-clone - Example

Definition

A λ-clone consists of

- an abstract clone $\mathcal{L}$,
- functions $\rho_n : \mathcal{L}(n) \to \mathcal{L}(n+1)$ and $\lambda_n : \mathcal{L}(n+1) \to \mathcal{L}(n)$.

Such that

$$\rho(f \bullet g) = \rho(f) \bullet (g_1, \ldots, g_m, x_{n+1,n+1})$$

and

$$\lambda(f) \bullet g = \lambda(f \bullet (g_1, \ldots, g_m, x_{n+1,n+1})).$$

Hyland also requires (λ, ρ) to be a section-retraction pair.

Example

The lambda calculus clone Λ is a λ-clone, with $\lambda_n f = \lambda x_{n+1}, f$ and $\rho_n(f) = f x_{n+1}$.
It is the initial λ-clone, so any algebra for a λ-clone is a "Λ-algebra".

λ-clone - Properties

Definition

A λ-clone consists of

- an abstract clone $\mathcal{L}$,
- functions $\rho_n : \mathcal{L}(n) \to \mathcal{L}(n+1)$ and $\lambda_n : \mathcal{L}(n+1) \to \mathcal{L}(n)$.

Such that

$$\rho(f \bullet g) = \rho(f) \bullet (g_1, \ldots, g_m, x_{n+1,n+1})$$

and

$$\lambda(f) \bullet g = \lambda(f \bullet (g_1, \ldots, g_m, x_{n+1,n+1})).$$

Hyland also requires (λ, ρ) to be a section-retraction pair.

Any algebra $(A, (\alpha_i)_i)$ for a λ-theory can be interpreted as a Λ-algebra.

work in progress

Preliminaries to the main theorem

For a λ-theory $\mathcal{L}$, $\mathcal{L}(0)$ is a $\mathcal{L}$-algebra and therefore a Λ-algebra.

Preliminaries to the main theorem

For a λ-theory $\mathcal{L}$, $\mathcal{L}(0)$ is a $\mathcal{L}$-algebra and therefore a Λ-algebra.

Given a Λ-algebra A. Take $\Lambda_A(n) = A + \Lambda(n)$, as a coproduct of Λ-algebras, defined as a coend of sets

$$A + B = \int^{m,n} Alg_\Lambda(\Lambda(m), A) \times Alg_\Lambda(\Lambda(n), B) \times \Lambda(m+n)$$

Preliminaries to the main theorem

For a λ-theory $\mathcal{L}$, $\mathcal{L}(0)$ is a $\mathcal{L}$-algebra and therefore a Λ-algebra.

Given a Λ-algebra A. Take $\Lambda_A(n) = A + \Lambda(n)$, as a coproduct of Λ-algebras, defined as a coend of sets

$$A + B = \int^{m,n} Alg_\Lambda(\Lambda(m), A) \times Alg_\Lambda(\Lambda(n), B) \times \Lambda(m+n)$$

Lemma

Λ_A is a λ-theory.

Proof.

We can identify Λ_A with $\mathcal{U}_A$.
We have a retraction $U_A \to U_A^{U_A}$. Composition with this gives a retraction $\mathcal{U}_A(n) \to \mathcal{U}_A(n+1)$. $\square$

$\mathcal{U}_A$

Let A be a Λ-algebra.

Definition

We define a monoid $M_A = (\{a \in A \mid \mathbf{1}a = a\}, \circ)$ with $\mathbf{1} := \lambda xy, xy = \alpha_0((\lambda xy, xy), ())$ and $a \circ b := \alpha_2((\lambda x, x_1(x_2 x)), (a, b))$.

$\mathcal{U}_A$

Let A be a Λ-algebra.

Definition

We define a monoid $M_A = (\{a \in A \mid \mathbf{1}a = a\}, \circ)$ with $\mathbf{1} := \lambda xy, xy = \alpha_0((\lambda xy, xy), ())$ and $a \circ b := \alpha_2((\lambda x, x_1(x_2 x)), (a, b))$.

Definition

We define $P(A)$ to be the category of presheaves on the category M_A. This has 'universal objec t' $U_A = M_A$ with the obvious right action of M_A.

$\mathcal{U}_A$

Let A be a Λ-algebra.

Definition

We define a monoid $M_A = (\{a \in A \mid \mathbf{1}a = a\}, \circ)$ with $\mathbf{1} := \lambda xy, xy = \alpha_0((\lambda xy, xy), ())$ and $a \circ b := \alpha_2((\lambda x, x_1(x_2 x)), (a, b))$.

Definition

We define $P(A)$ to be the category of presheaves on the category M_A. This has 'universal objec t' $U_A = M_A$ with the obvious right action of M_A.

Lemma

We have a retraction $U_A \to (U_A \to U_A)$.

Proof.

We can identify $U_A \to U_A$ with $\{a \in A \mid \mathbf{1}_2 a = a\}$ with $\mathbf{1}_2 := \lambda xy_1y_2, (xy_1)y_2$.
Composition on the left with $\mathbf{1}$ gives the retraction. $\square$

$\mathcal{U}_A$

Let A be a Λ-algebra.

Definition

We define a monoid $M_A = (\{a \in A \mid \mathbf{1}a = a\}, \circ)$ with $\mathbf{1} := \lambda xy, xy = \alpha_0((\lambda xy, xy), ())$ and $a \circ b := \alpha_2((\lambda x, x_1(x_2 x)), (a, b))$.

Definition

We define $P(A)$ to be the category of presheaves on the category M_A. This has 'universal objec t' $U_A = M_A$ with the obvious right action of M_A.

Lemma

We have a retraction $U_A \to (U_A \to U_A)$.

Proof.

We can identify $U_A \to U_A$ with $\{a \in A \mid \mathbf{1}_2 a = a\}$ with $\mathbf{1}_2 := \lambda x y_1 y_2, (x y_1) y_2$. Composition on the left with $\mathbf{1}$ gives the retraction. $\square$

Definition

We take $\mathcal{U}_A$ to be the endomorphism theory of the reflexive universal $U_A \in P(A)$.

The Main Theorem of the Lambda Calculus

Theorem

There is an adjoint equivalence $\mathcal{L} \mapsto \mathcal{L}(0)$ and $A \mapsto \Lambda_A$ between λ-theories and Λ-algebras.
In particular, each λ-theory $\mathcal{L}$ is isomorphic to the theory of extensions of its initial algebra $\mathcal{L}(0)$.

Conclusion

In this framework, we can study the denotations/interpretations/models for the lambda calculus by studying the λ-theories.