Carving out a Proof Theory from Cedille's Core

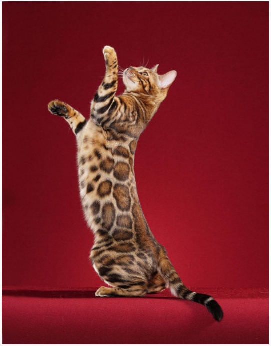
Andrew Marmaduke

10/6/2023

The University of Iowa

Entering the Multiverse

What might have Cedille looked like in another universe?



Motivating Examples

How do we build subsets like the set of Even natural numbers in a conceptually intuitive way?

How do we construct inductive types without efficiency problems with lambda encoded data?

Motivating Examples

How do we build subsets like the set of Even natural numbers in a conceptually intuitive way?

How do we construct inductive types without efficiency problems with lambda encoded data?

Refresher on Dependent Types

$$f : \mathbb{N} \to \mathbb{N}$$

function between natural numbers

$$P : \mathbb{N} \to \mathrm{Set}$$

predicate on natural numbers

$$g : (A : \mathrm{Set}) \to \mathrm{List}\ A$$

(parametric) function from types to lists

$$h : (n : \mathbb{N}) \to P\ n$$

(dependent) function from naturals to instantiated predicate

Refresher on Dependent Types

$$f : \mathbb{N} \to \mathbb{N}$$

function between natural numbers

$$P : \mathbb{N} \to \mathrm{Set}$$

predicate on natural numbers

$$g : (A : \mathrm{Set}) \to \mathrm{List}\ A$$

(parametric) function from types to lists

$$h : (n : \mathbb{N}) \to P\ n$$

(dependent) function from naturals to instantiated predicate

Refresher on Dependent Types

$$f : \mathbb{N} \to \mathbb{N}$$

function between natural numbers

$$P : \mathbb{N} \to \mathrm{Set}$$

predicate on natural numbers

$$g : (A : \mathrm{Set}) \to \mathrm{List}\ A$$

(parametric) function from types to lists

$$h : (n : \mathbb{N}) \to P\ n$$

(dependent) function from naturals to instantiated predicate

Refresher on Dependent Types

$$f : \mathbb{N} \to \mathbb{N}$$

function between natural numbers

$$P : \mathbb{N} \to \mathrm{Set}$$

predicate on natural numbers

$$g : (A : \mathrm{Set}) \to \mathrm{List}\ A$$

(parametric) function from types to lists

$$h : (n : \mathbb{N}) \to P\ n$$

(dependent) function from naturals to instantiated predicate

Constructing the Set of Even Numbers

$$\mathrm{IsEven} : \mathbb{N} \to \mathrm{Set}$$

any sound way of representing evenness of a natural

$$(a : A) \times P\ a$$

is how we will write *dependent pairs*

$$\mathrm{Even} = (n : \mathbb{N}) \times \mathrm{IsEven}\ n$$

pair a natural with evidence it's even

Constructing the Set of Even Numbers

$$\text{IsEven} : \mathbb{N} \to \text{Set}$$

any sound way of representing evenness of a natural

$$(a : A) \times P\ a$$

is how we will write *dependent pairs*

$$\text{Even} = (n : \mathbb{N}) \times \text{IsEven}\ n$$

pair a natural with evidence it's even

Constructing the Set of Even Numbers

$$\text{IsEven} : \mathbb{N} \to \text{Set}$$

any sound way of representing evenness of a natural

$$(a : A) \times P\ a$$

is how we will write *dependent pairs*

$$\text{Even} = (n : \mathbb{N}) \times \text{IsEven}\ n$$

pair a natural with evidence it's even

How to connect Even and Nat?

$$\iota : \text{Even} \hookrightarrow \mathbb{N}$$

ask a mathematician, and they say there should be an *injection* from Even to $\mathbb{N}$

$$\iota \equiv \text{fst} \quad 2 : \mathbb{N} \neq 2 : \text{Even}$$

with the current definition

$$\iota \equiv \text{id}$$

can we change things so that the injection is equal to the identity?

How to connect Even and Nat?

$$\iota : \text{Even} \hookrightarrow \mathbb{N}$$

ask a mathematician, and they say there should be an *injection* from Even to $\mathbb{N}$

$$\iota \equiv \text{fst} \quad 2 : \mathbb{N} \neq 2 : \text{Even}$$

with the current definition

$$\iota \equiv \text{id}$$

can we change things so that the injection is equal to the identity?

How to connect Even and Nat?

$$\iota : \text{Even} \hookrightarrow \mathbb{N}$$

ask a mathematician, and they say there should be an *injection* from Even to $\mathbb{N}$

$$\iota \equiv \text{fst} \quad 2 : \mathbb{N} \neq 2 : \text{Even}$$

with the current definition

$$\iota \equiv \text{id}$$

can we change things so that the injection is equal to the identity?

Quantification w.r.t. what?

$$h : (n : \mathbb{N}) \to P\ n$$

via the Curry-Howard correspondence, this is a universal quantification, but what is the domain? The natural numbers? **No**. The domain is the *proofs* for the given type.

For Even, it is the *proofs* that the element is an even natural, and hence a pair.

Quantification w.r.t. what?

$$h : (n : \mathbb{N}) \to P\ n$$

via the Curry-Howard correspondence, this is a universal quantification, but what is the domain? The natural numbers? **No**. The domain is the *proofs* for the given type.

For Even, it is the *proofs* that the element is an even natural, and hence a pair.

Quantifying over Something New

$$|\cdot| : \text{Proofs} \to \text{Objects}$$

erasure constructs an *object* (or an *individual*) from a proof

Change conversion from $t =_\beta s$ to

$$\exists\, t'\ s'.\ t \to_\beta t' \wedge s \to_\beta s' \wedge |t'| = |s'|$$

Dependent types now quantify over a domain of **objects** instead of proofs.

Quantifying over Something New

$$|\cdot| : \text{Proofs} \to \text{Objects}$$

erasure constructs an *object* (or an *individual*) from a proof

Change conversion from $t =_\beta s$ to

$$\exists\, t'\ s'.\ t \to_\beta t' \wedge s \to_\beta s' \wedge |t'| = |s'|$$

Dependent types now quantify over a domain of **objects** instead of proofs.

Quantifying over Something New

$$|\cdot| : \text{Proofs} \to \text{Objects}$$

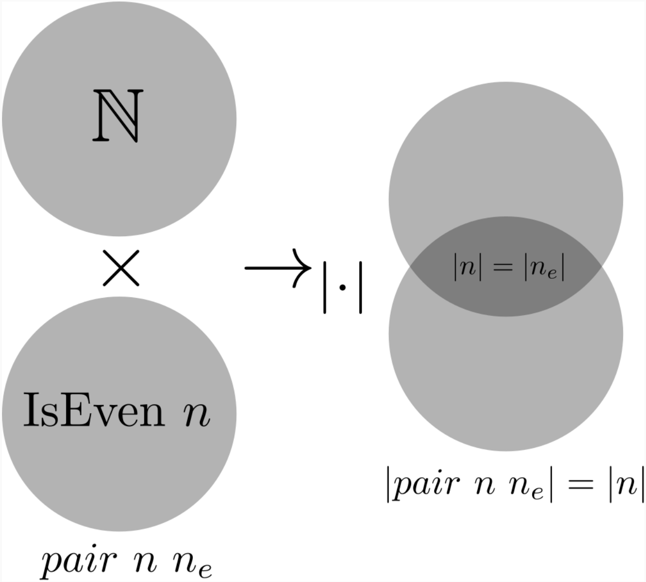
erasure constructs an *object* (or an *individual*) from a proof

Change conversion from $t =_\beta s$ to

$$\exists\, t'\ s'.\ t \to_\beta t' \wedge s \to_\beta s' \wedge |t'| = |s'|$$

Dependent types now quantify over a domain of **objects** instead of proofs.

Modifying Dependent Pairs



Modifying Even

$$\{x \in \mathbb{N} \mid \exists\ k,\ 2k = x\}$$

$$(x : \mathbb{N}) \times \text{IsEven } x$$

$$(x : \mathbb{N}) \cap \text{IsEven } x$$

Is it too hard to inhabit?

Is it too difficult to find types that can be intersected to meet the side-condition?

$$\Lambda\, x.\, t : (x : A) \Rightarrow P\ x \qquad |\Lambda\, x.\, t| = |t| \qquad x \notin \mathrm{FV}(|t|)$$

Add erased functions (or object-irrelevant functions)

Let's us mark indices as erased, so that the objects of Vec A n are the same objects as List A, additionally all types become erased

Is it too hard to inhabit?

Is it too difficult to find types that can be intersected to meet the side-condition?

$$\Lambda\, x.\, t : (x : A) \Rightarrow P\ x \qquad |\Lambda\, x.\, t| = |t| \qquad x \notin \mathrm{FV}(|t|)$$

Add erased functions (or object-irrelevant functions)

Let's us mark indices as erased, so that the objects of Vec A n are the same objects as List A, additionally all types become erased

Is it too hard to inhabit?

Is it too difficult to find types that can be intersected to meet the side-condition?

$$\Lambda\, x.\, t : (x : A) \Rightarrow P\ x \qquad |\Lambda\, x.\, t| = |t| \qquad x \notin \mathrm{FV}(|t|)$$

Add erased functions (or object-irrelevant functions)

Let's us mark indices as erased, so that the objects of Vec A n are the same objects as List A, additionally all types become erased

Inductive Types

To derive inductive types, we also need equality

Equality

Take the standard Martin-Löf Identity Type

$$\mathrm{J}\ \text{-}A\ \text{-}P\ \text{-}x\ \text{-}y\ r\ w$$

Types must be erased, but also mark indices as erased (critically, the equality evidence r cannot be erased)

$$\mathrm{refl}\ \text{-}x$$

Opportunity: we can mark the input to refl erased, there can only be one object corresponding to it!

Equality

Take the standard Martin-Löf Identity Type

$$\mathrm{J}\ \text{-}A\ \text{-}P\ \text{-}x\ \text{-}y\ r\ w$$

Types must be erased, but also mark indices as erased (critically, the equality evidence r cannot be erased)

$$\mathrm{refl}\ \text{-}x$$

Opportunity: we can mark the input to refl erased, there can only be one object corresponding to it!

Equality

Take the standard Martin-Löf Identity Type

$$\text{J -}A\text{ -}P\text{ -}x\text{ -}y\ r\ w$$

Types must be erased, but also mark indices as erased
(critically, the equality evidence r cannot be erased)

$$\text{refl -}x$$

Opportunity: we can mark the input to refl erased, there can only be one object corresponding to it!

Equality, Reasoning about Intersection

Need to add a rule to reason about dependent intersections

$$\text{promote} : (x\ y : (a : A) \cap P\ a) \Rightarrow x.1 =_A y.1 \to x =_{(a:A) \cap P a}\ y$$

Compensates for lack of an induction rule

Inductive Types are Derivable

The system as described is strong enough to derive inductive types (via an impredicative church encoding)

This is basically Cedille's Core! (except equality is much less exotic)

To get to Mendler encodings we need a bit more

Inductive Types are Derivable

The system as described is strong enough to derive inductive types (via an impredicative church encoding)

This is basically Cedille's Core! (except equality is much less exotic)

To get to Mendler encodings we need a bit more

Inductive Types are Derivable

The system as described is strong enough to derive inductive types (via an impredicative church encoding)

This is basically Cedille's Core! (except equality is much less exotic)

To get to Mendler encodings we need a bit more

Looking for a postdoc for Fall 2024

Q & A