Parametricity: A Story in Trivializing 15-150

Hype for Types

March 20, 2024

Motivation

Identity

Recall from last week the function $f : \forall X.X \to X$. A natural question to ask is "how many such functions are there?"

[1] Above is not a proof

Identity

Recall from last week the function $f : \forall X.X \to X$. A natural question to ask is "how many such functions are there?"

One. Because...

[1] Above is not a proof

Identity

Recall from last week the function $f : \forall X.X \to X$. A natural question to ask is "how many such functions are there?"

One. Because... you get an $x : \alpha$...

[1] Above is not a proof

Identity

Recall from last week the function $f : \forall X.X \to X$. A natural question to ask is "how many such functions are there?"

One. Because... you get an $x : \alpha$... and...

[1] Above is not a proof

Identity

Recall from last week the function $f : \forall X.X \to X$. A natural question to ask is "how many such functions are there?"

One. Because... you get an $x : \alpha$... and... what else can you do with it besides return it.

[1] Above is not a proof

Identity

Recall from last week the function $f : \forall X.X \to X$. A natural question to ask is "how many such functions are there?"

One. Because... you get an $x : \alpha$... and... what else can you do with it besides return it. Or something...

[1] Above is not a proof

Identity

Recall from last week the function $f : \forall X.X \to X$. A natural question to ask is "how many such functions are there?"

One. Because... you get an $x : \alpha$... and... what else can you do with it besides return it. Or something...

This is not very satisfying. So, we would like an equational theory for polymorphic functions to *prove*[1] that there is only one such function.

[1] Above is not a proof

More Generally...

If I give you a function $f : \forall X.\mathsf{List}(X) \to \mathsf{List}(X)$ what function do you expect it to be?

You probably said Reverse or Duplicate-Every-Element or Take-The-First-Two-Elements-And-Copy-Them-Five-Times-And-Then-Append-The-Third-Element-To-The-End[2] : $\forall X.\mathsf{List}(X) \to \mathsf{List}(X)$.

[2] Pretend this is total

More Generally...

If I give you a function $f : \forall X.\mathsf{List}(X) \to \mathsf{List}(X)$ what function do you expect it to be?

You probably said Reverse or Duplicate-Every-Element or Take-The-First-Two-Elements-And-Copy-Them-Five-Times-And-Then-Append-The-Third-Element-To-The-End[2] : $\forall X.\mathsf{List}(X) \to \mathsf{List}(X)$.

The point is that any function you described is returning some permutation/duplication/removal of the elements which *does not refer to the values themselves*.

[2] Pretend this is total

Mapping over these

Take your function f from before, and now take your favorite function $g : A \to B$. Consider the following equation:

$$(\mathsf{map}\, g) \circ f = f \circ (\mathsf{map}\, g)$$

[3] This is a lie. Induction is my favorite proof technique and it's not even close

Mapping over these

Take your function f from before, and now take your favorite function $g : A \to B$. Consider the following equation:

$$(\mathsf{map}\, g) \circ f = f \circ (\mathsf{map}\, g)$$

It turns out this is true. The intuition is that "Since f cannot refer to the elements themselves, mapping a function g then permuting the list should be the same as permuting the list then mapping a function g."

You probably proved in 15-150 something like

$$\text{For all } f : A \to B,\ (\mathsf{map}\, f) \circ \mathsf{reverse} = \mathsf{reverse} \circ (\mathsf{map}\, f)$$

By induction on the list or something.

[3] This is a lie. Induction is my favorite proof technique and it's not even close

Mapping over these

Take your function f from before, and now take your favorite function $g : A \to B$. Consider the following equation:

$$(\mathsf{map}\, g) \circ f = f \circ (\mathsf{map}\, g)$$

It turns out this is true. The intuition is that "Since f cannot refer to the elements themselves, mapping a function g then permuting the list should be the same as permuting the list then mapping a function g."

You probably proved in 15-150 something like

$$\text{For all } f : A \to B,\ (\mathsf{map}\, f) \circ \mathsf{reverse} = \mathsf{reverse} \circ (\mathsf{map}\, f)$$

By induction on the list or something. I hate induction,[3] let's do better.

[3] This is a lie. Induction is my favorite proof technique and it's not even close

What the Hype is a Type

Let's ask a fundamental question. How do you think about types?

[4] What the hype is a set? Like actually, can someone please explain it to me without "oh it's an element of V" and then laughing maniacally

[5] Kinda Sorta Not Really But...

[6] Yar, thar be domains in these seas

What the Hype is a Type

Let's ask a fundamental question. How do you think about types? You probably view types as sets[4].

- $[\![\text{Bool}]\!] = \{0, 1\}$
- $[\![\text{Int}]\!] = \mathbb{Z}$
- $[\![A \times B]\!] = [\![A]\!] \times [\![B]\!]$
- $[\![A \to B]\!] = B^A$
- $[\![\text{List}(A)]\!] = A^*$

This is generally fine[5][6], but today we will view types as relations.

[4] What the hype is a set? Like actually, can someone please explain it to me without "oh it's an element of V" and then laughing maniacally

[5] Kinda Sorta Not Really But...

[6] Yar, thar be domains in these seas

Some Notation and Ideas

- $\mathcal{A} : A \Leftrightarrow A'$ means $\mathcal{A}$ is a relation between A and A' i.e. $\mathcal{A} \subseteq A \times A'$.
- If $x \in A$ and $x' \in A'$, we write $(x, x') \in \mathcal{A}$ to mean x and x' are related by $\mathcal{A}$.
- I_A is the identity relation on A i.e. for all $x \in A$, $(x, x) \in I_A$.
- We may view any function $f : A \to B$ as a relation $A \Leftrightarrow B$ via $\{(a, f\ a) \mid a \in A\}$

Types as relations

We may interpret some basic types as relations in the following manner:

- $[\![\mathsf{Int}]\!] = I_{\mathsf{Int}}$
- $[\![\mathsf{Bool}]\!] = I_{\mathsf{Bool}}$
- $[\![A \times B]\!] = \{((x, y), (x', y')) \mid (x, x') \in A \text{ and } (y, y') \in B\}$.

Now informally:

For a relation $\mathcal{A} : A \Leftrightarrow A'$, we give the relation $\mathsf{List}(\mathcal{A})$ by two lists having the same length and their elements being pair-wise related by $\mathcal{A}$

For two relations $\mathcal{A} : A \Leftrightarrow A'$ and $\mathcal{B} : B \Leftrightarrow B'$, the relation $\mathcal{A} \to \mathcal{B}$ says two functions are related if they take related inputs under $\mathcal{A}$ to related outputs under $\mathcal{B}$.

Polymorphic functions are related if they take related types to related outputs.

The Big Theorem

What we've been working for:
The Parametricity Theorem

The Big Theorem

What we've been working for:
The Parametricity Theorem

$$\text{If } t : T, \text{ then } (t, t) \in \mathcal{T}$$

The Big Theorem

What we've been working for:
The Parametricity Theorem

$$\text{If } t : T\text{, then } (t, t) \in \mathcal{T}$$

That's... kinda underwhelming.

Why Should you Care

Hang on hang on, before you leave, let's look back at our example from earlier. Recall, we wanted to prove

$$\text{For all functions } f : A \to B \text{ and } r : \forall X.\mathsf{List}(X) \to \mathsf{List}(X),$$
$$(\mathsf{map}\, f) \circ r = r \circ (\mathsf{map}\, f)$$

Maybe our new parametricity theorem can help?

A Parametrically Polymorphic Proof

1. Parametricity tells us $(r, r) \in \forall \mathcal{X}.\mathsf{List}(\mathcal{X}) \to \mathsf{List}(\mathcal{X})$.

[7] Recall r[A] is the polymorphic function r applied to the type A

A Parametrically Polymorphic Proof

1. Parametricity tells us $(r, r) \in \forall \mathcal{X}.\mathsf{List}(\mathcal{X}) \to \mathsf{List}(\mathcal{X})$.
2. We can expand this to see that for all relations $\mathcal{A} : A \Leftrightarrow A'$, $(r[A]^7, r[A']) \in \mathsf{List}(\mathcal{A}) \to \mathsf{List}(\mathcal{A})$

[7] Recall r[A] is the polymorphic function r applied to the type A

A Parametrically Polymorphic Proof

1. Parametricity tells us $(r, r) \in \forall \mathcal{X}.\text{List}(\mathcal{X}) \to \text{List}(\mathcal{X})$.
2. We can expand this to see that for all relations $\mathcal{A} : A \Leftrightarrow A'$, $(r[A]^7, r[A']) \in \text{List}(\mathcal{A}) \to \text{List}(\mathcal{A})$
3. We can then expand this to see that for all relations $\mathcal{A} : A \Leftrightarrow A'$, for all $(xs, xs') \in \text{List}(\mathcal{A})$, $(r[A](xs), r[A'](xs')) \in \text{List}(\mathcal{A})$

This seems to be getting us somewhere.. but this is too general to be useful... Let's focus on when $\mathcal{A}$ is a relation induced by a function $f : A \to A'$.

[7] Recall r[A] is the polymorphic function r applied to the type A

A Parametrically Polymorphic Proof

1. Parametricity tells us $(r, r) \in \forall \mathcal{X}.\mathsf{List}(\mathcal{X}) \to \mathsf{List}(\mathcal{X})$.
2. We can expand this to see that for all relations $\mathcal{A} : A \Leftrightarrow A'$, $(r[A]^7, r[A']) \in \mathsf{List}(\mathcal{A}) \to \mathsf{List}(\mathcal{A})$
3. We can then expand this to see that for all relations $\mathcal{A} : A \Leftrightarrow A'$, for all $(xs, xs') \in \mathsf{List}(\mathcal{A})$, $(r[A](xs), r[A'](xs')) \in \mathsf{List}(\mathcal{A})$

This seems to be getting us somewhere.. but this is too general to be useful... Let's focus on when $\mathcal{A}$ is a relation induced by a function $f : A \to A'$.

For all functions $f : A \to A'$, for all $(\mathsf{map}\, f\, xs, xs) \in \mathcal{R}_f$, implies $(r[A](\mathsf{map}\, f\, xs), r[A'](xs)) \in \mathsf{List}(\mathcal{R}_f)$. This seems very close...

[7] Recall r[A] is the polymorphic function r applied to the type A

A Parametrically Polymorphic Proof

We now know that for all functions $f : A \to A'$, for all $(\text{map}\, f\ xs, xs) \in \text{List}(\mathcal{R}_f)$, implies $(r[A](\text{map}\, f\ xs), r[A'](xs)) \in \text{List}(\mathcal{R}_f)$.

A Parametrically Polymorphic Proof

We now know that for all functions $f : A \to A'$, for all $(\text{map}\, f\, xs, xs) \in \text{List}(\mathcal{R}_f)$, implies $(r[A](\text{map}\, f\, xs), r[A'](xs)) \in \text{List}(\mathcal{R}_f)$.

Recall, two terms are related by $\text{List}(\mathcal{R}_f)$ if they have equal length, and the elements are pointwise related.

A Parametrically Polymorphic Proof

We now know that for all functions $f : A \to A'$, for all $(\mathsf{map}\, f\ xs, xs) \in \mathsf{List}(\mathcal{R}_f)$, implies $(r[A](\mathsf{map}\, f\ xs), r[A'](xs)) \in \mathsf{List}(\mathcal{R}_f)$.

Recall, two terms are related by $\mathsf{List}(\mathcal{R}_f)$ if they have equal length, and the elements are pointwise related. Our relation here is that $(x, f\ x) \in \mathcal{R}_f$.

A Parametrically Polymorphic Proof

We now know that for all functions $f : A \to A'$, for all $(\mathsf{map}\, f\, xs, xs) \in \mathsf{List}(\mathcal{R}_f)$, implies $(r[A](\mathsf{map}\, f\, xs), r[A'](xs)) \in \mathsf{List}(\mathcal{R}_f)$.

Recall, two terms are related by $\mathsf{List}(\mathcal{R}_f)$ if they have equal length, and the elements are pointwise related. Our relation here is that $(x, f\, x) \in \mathcal{R}_f$. In otherwords,

$$\text{For all } f : A \to A',\ r[A](\mathsf{map}\, f\, xs) = \mathsf{map}\, f\, (r[A'](xs))$$

or more cleanly

$$\text{For all } r : \forall X.\mathsf{List}(X) \to \mathsf{List}(X), \text{ for all } f : A \to A', \\ r[A] \circ (\mathsf{map}\, f) = (\mathsf{map}\, f) \circ r[A']$$

15-150? More like... Parametricity Theorem

We did it! Not only did we prove that

$$\text{reverse} \circ (\text{map } f) = (\text{map } f) \circ \text{reverse}$$

we managed to prove something way more general!

The original Goal

I claim that if $f : \forall X. X \to X$, then $f = \text{id}$. You know this intuitively, but we can use parametricity to prove this!

The original Goal

I claim that if $f : \forall X.X \to X$, then $f = \text{id}$. You know this intuitively, but we can use parametricity to prove this!

1. $(f, f) \in \forall \mathcal{X}.\mathcal{X} \to \mathcal{X}$

The original Goal

I claim that if $f : \forall X.X \to X$, then $f = \text{id}$. You know this intuitively, but we can use parametricity to prove this!

1. $(f, f) \in \forall \mathcal{X}.\mathcal{X} \to \mathcal{X}$
2. For all functions $g : A \to A'$, $(f[A], f[A]) \in \mathcal{R}_g \to \mathcal{R}_g$.

The original Goal

I claim that if $f : \forall X.X \to X$, then $f = \text{id}$. You know this intuitively, but we can use parametricity to prove this!

1. $(f, f) \in \forall \mathcal{X}.\mathcal{X} \to \mathcal{X}$
2. For all functions $g : A \to A'$, $(f[A], f[A]) \in \mathcal{R}_g \to \mathcal{R}_g$.
3. For all functions $g : A \to A'$, $(g \circ f[A], f[A]) \in \mathcal{R}_g$

The original Goal

I claim that if $f : \forall X.X \to X$, then $f = \text{id}$. You know this intuitively, but we can use parametricity to prove this!

1. $(f, f) \in \forall \mathcal{X}.\mathcal{X} \to \mathcal{X}$
2. For all functions $g : A \to A'$, $(f[A], f[A]) \in \mathcal{R}_g \to \mathcal{R}_g$.
3. For all functions $g : A \to A'$, $(g \circ f[A], f[A]) \in \mathcal{R}_g$
4. For all functions $g : A \to A'$, $g \circ f[A] = f[A] \circ g$.

Hmm this seems close... we need one final trick.

The original Goal

I claim that if $f : \forall X.X \to X$, then $f = \text{id}$. You know this intuitively, but we can use parametricity to prove this!

1. $(f, f) \in \forall \mathcal{X}.\mathcal{X} \to \mathcal{X}$
2. For all functions $g : A \to A'$, $(f[A], f[A]) \in \mathcal{R}_g \to \mathcal{R}_g$.
3. For all functions $g : A \to A'$, $(g \circ f[A], f[A]) \in \mathcal{R}_g$
4. For all functions $g : A \to A'$, $g \circ f[A] = f[A] \circ g$.

Hmm this seems close... we need one final trick.

Well, by function extensionality, we know that

$$\forall x : A, \forall g : A \to A', g(f[A]\, x) = f[A](g\, x)$$

What if we pick $g = \lambda\_.x$! We then have that $g(f[A]\, x) = x$ and $f[A](g\, x) = f[A](x)$. In otherwords, $x = f[A](x)$!

Free Theorems

Theorems of this form are called “free theorems” named after Phillip Wadler’s Paper called, unsurprisingly “Theorems for Free”.

Free Theorems

Theorems of this form are called “free theorems” named after Phillip Wadler’s Paper called, unsurprisingly “Theorems for Free”.

Such theorems are direct consequences of the Parametricity Theorem and allow you to prove basically any 15-150 style equality... for free!

https://free-theorems.nomeata.de/