

Pierre Cagne

joint work with Patricia Johann

Appalachian State University

Partiality wrecks GADTs' functoriality

How to get Initial Algebra Semantics for
GADTs?

TYPES 2023 – València

June 14th, 2023

1. First-order IAS for ADTs

2. Higher-order IAS for ADTs

3. IAS for GADTs?



1. First-order IAS for ADTs



ADTs

ADTs are families of inductive types:

ADTs

ADTs are families of inductive types:

```
data List ( $\alpha$  : Set) : Set where
  [] : List  $\alpha$ 
  _::_ :  $\alpha \rightarrow$  List  $\alpha \rightarrow$  List  $\alpha$ 
```

ADTs

ADTs are families of inductive types:

```
data List ( $\alpha$  : Set) : Set where
```

```
  [] : List  $\alpha$ 
```

```
  _::_ :  $\alpha \rightarrow$  List  $\alpha \rightarrow$  List  $\alpha$ 
```

```
data BinTree ( $\alpha$  : Set) : Set where
```

```
   $\emptyset$  : BinTree  $\alpha$ 
```

```
  _ $\otimes$ _ $\otimes$ _ : BinTree  $\alpha \rightarrow \alpha \rightarrow$  BinTree  $\alpha \rightarrow$  BinTree  $\alpha$ 
```

ADTs

ADTs are families of inductive types:

```
data List ( $\alpha$  : Set) : Set where
  [] : List  $\alpha$ 
  _::_ :  $\alpha \rightarrow$  List  $\alpha \rightarrow$  List  $\alpha$ 
```

```
data BinTree ( $\alpha$  : Set) : Set where
   $\emptyset$  : BinTree  $\alpha$ 
  _ $\otimes$ _ $\otimes$ _ : BinTree  $\alpha \rightarrow \alpha \rightarrow$  BinTree  $\alpha \rightarrow$  BinTree  $\alpha$ 
```

```
data N : Set where
  zero : N
  succ : N  $\rightarrow$  N
```

Categorical semantics of ADTs

```
data List ( $\alpha$  : Set) : Set where
  [] :  $\tau \rightarrow$  List  $\alpha$ 
  _::_ :  $\alpha \rightarrow$  List  $\alpha \rightarrow$  List  $\alpha$ 
```


Categorical semantics of ADTs

```
data List ( $\alpha$  : Set) : Set where
  [] :  $\tau \rightarrow$  List  $\alpha$ 
  _::_ :  $\alpha \rightarrow$  List  $\alpha \rightarrow$  List  $\alpha$ 
```

\mathcal{C} : category with finite products, finite sums, and colimits of ω -chains.

To interpret `List A`, take the initial algebra μL_A of:

$$L_A : \mathcal{C} \rightarrow \mathcal{C} \qquad \text{where } A \text{ interprets } A$$
$$X \mapsto 1 + (A \times X)$$

Categorical semantics of ADTs

ADTs are *uniform* families of inductive types:

```
data List ( $\alpha$  : Set) : Set where
  [] :  $\tau \rightarrow$  List  $\alpha$ 
  _::_ :  $\alpha \rightarrow$  List  $\alpha \rightarrow$  List  $\alpha$ 
```

Categorical semantics of ADTs

ADTs are *uniform* families of inductive types:

```
data List (α : Set) : Set where
  [] : τ → List α
  _::_ : α → List α → List α
```

$$\mathcal{C} \xrightarrow{L} [\mathcal{C}, \mathcal{C}]_{\omega} \xrightarrow{\mu} \mathcal{C}$$

$$A \longmapsto L_A \longmapsto \mu L_A$$



2. Higher-order IAS for ADTs



Categorical semantics of ADTs

ADTs can be seen as inductive families of types:

```
data List : Set → Set where
```

```
  [] : ∀ α → τ → List α
```

```
  _::_ : ∀ α → α → List α → List α
```

Categorical semantics of ADTs

ADTs can be seen as inductive families of types:

```
data List : Set → Set where
```

```
  [] : ∀ α → τ → List α
```

```
  _::_ : ∀ α → α → List α → List α
```

Rework the semantics: to interpret the type constructor `List`, take the initial algebra $\mu\mathcal{L}$ of

$$\mathcal{L} : [\mathcal{C}, \mathcal{C}] \rightarrow [\mathcal{C}, \mathcal{C}]$$

$$F \mapsto (X \mapsto 1 + (X \times F(X)))$$

Categorical semantics of ADTs

ADTs can be seen as inductive families of types:

```
data List : Set → Set where
```

```
  [] : ∀ α → τ → List α
```

```
  _::_ : ∀ α → α → List α → List α
```

Rework the semantics: to interpret the type constructor `List`, take the initial algebra $\mu\mathcal{L}$ of

$$\mathcal{L} : [\mathcal{C}, \mathcal{C}] \rightarrow [\mathcal{C}, \mathcal{C}]$$

$$F \mapsto (X \mapsto 1 + (X \times F(X)))$$

All is well

$$\mu\mathcal{L} \simeq \mu \circ L.$$

3. IAS for GADTs?

Generalized Algebraic Data Types

GADTs are inductive families of types, with a twist:

data Terms : Set → Set **where**

nat : ℕ → Terms ℕ

-,_ : ∀ {α β} → Terms α → Terms β → Terms (α × β)

π₁ : ∀ {α β} → Terms (α × β) → Terms α

π₂ : ∀ {α β} → Terms (α × β) → Terms β

Generalized Algebraic Data Types

GADTs are inductive families of types, with a twist:

data Terms : Set → Set where

nat : $\mathbb{N} \rightarrow \text{Terms } \mathbb{N}$

$-, -$: $\forall \{\alpha \beta\} \rightarrow \text{Terms } \alpha \rightarrow \text{Terms } \beta \rightarrow \text{Terms } (\alpha \times \beta)$

π_1 : $\forall \{\alpha \beta\} \rightarrow \text{Terms } (\alpha \times \beta) \rightarrow \text{Terms } \alpha$

π_2 : $\forall \{\alpha \beta\} \rightarrow \text{Terms } (\alpha \times \beta) \rightarrow \text{Terms } \beta$

data W : Set → Set where

\exists : $\forall \alpha \rightarrow \alpha \rightarrow W \tau$

Generalized Algebraic Data Types

GADTs are inductive families of types, with a twist:

data Terms : Set → Set where

nat : $\mathbb{N} \rightarrow$ Terms \mathbb{N}

, : $\forall \{\alpha \beta\} \rightarrow$ Terms $\alpha \rightarrow$ Terms $\beta \rightarrow$ Terms $(\alpha \times \beta)$

π_1 : $\forall \{\alpha \beta\} \rightarrow$ Terms $(\alpha \times \beta) \rightarrow$ Terms α

π_2 : $\forall \{\alpha \beta\} \rightarrow$ Terms $(\alpha \times \beta) \rightarrow$ Terms β

data W : Set → Set where

\exists : $\forall \alpha \rightarrow \alpha \rightarrow$ W τ

data _ \equiv _ : Set → Set → Set where

r : $\forall \alpha \rightarrow \alpha \equiv \alpha$

Naive categorical semantics of GADTs

data Terms : Set \rightarrow Set where

nat : $\mathbb{N} \rightarrow$ Terms \mathbb{N}

$-, -$: $\forall \{\alpha \beta\} \rightarrow$ Terms $\alpha \rightarrow$ Terms $\beta \rightarrow$ Terms $(\alpha \times \beta)$

π_1 : $\forall \{\alpha \beta\} \rightarrow$ Terms $(\alpha \times \beta) \rightarrow$ Terms α

π_2 : $\forall \{\alpha \beta\} \rightarrow$ Terms $(\alpha \times \beta) \rightarrow$ Terms β

Naive categorical semantics of GADTs

data Terms : Set \rightarrow Set where

nat : $\mathbb{N} \rightarrow$ Terms \mathbb{N}

$-, -$: $\forall \{\alpha \beta\} \rightarrow$ Terms $\alpha \rightarrow$ Terms $\beta \rightarrow$ Terms $(\alpha \times \beta)$

π_1 : $\forall \{\alpha \beta\} \rightarrow$ Terms $(\alpha \times \beta) \rightarrow$ Terms α

π_2 : $\forall \{\alpha \beta\} \rightarrow$ Terms $(\alpha \times \beta) \rightarrow$ Terms β

To interpret the type constructor Terms, take the initial algebra $\mu\mathcal{T}$ of:

$$\mathcal{T} : [\mathcal{C}, \mathcal{C}] \rightarrow [\mathcal{C}, \mathcal{C}]$$

$$F \mapsto \left(\begin{array}{c} X \mapsto \\ + \\ + \\ + \end{array} \right)$$

Naive categorical semantics of GADTs

data Terms : Set \rightarrow Set where

nat : $\mathbb{N} \rightarrow$ Terms \mathbb{N}

$-, -$: $\forall \{\alpha \beta\} \rightarrow$ Terms $\alpha \rightarrow$ Terms $\beta \rightarrow$ Terms $(\alpha \times \beta)$

π_1 : $\forall \{\alpha \beta\} \rightarrow$ Terms $(\alpha \times \beta) \rightarrow$ Terms α

π_2 : $\forall \{\alpha \beta\} \rightarrow$ Terms $(\alpha \times \beta) \rightarrow$ Terms β

To interpret the type constructor Terms, take the initial algebra $\mu\mathcal{T}$ of:

$\mathcal{T} : [\mathcal{C}, \mathcal{C}] \rightarrow [\mathcal{C}, \mathcal{C}]$

$$F \mapsto \left(\begin{array}{l} X \mapsto N \text{ if } X = N, \emptyset \text{ otherwise} \\ + \\ + \\ + \end{array} \right)$$

Naive categorical semantics of GADTs

data Terms : Set \rightarrow Set where

nat : $\mathbb{N} \rightarrow$ Terms \mathbb{N}

$-, -$: $\forall \{\alpha \beta\} \rightarrow$ Terms $\alpha \rightarrow$ Terms $\beta \rightarrow$ Terms $(\alpha \times \beta)$

π_1 : $\forall \{\alpha \beta\} \rightarrow$ Terms $(\alpha \times \beta) \rightarrow$ Terms α

π_2 : $\forall \{\alpha \beta\} \rightarrow$ Terms $(\alpha \times \beta) \rightarrow$ Terms β

To interpret the type constructor Terms, take the initial algebra $\mu\mathcal{T}$ of:

$\mathcal{T} : [\mathcal{C}, \mathcal{C}] \rightarrow [\mathcal{C}, \mathcal{C}]$

$$F \mapsto \left(\begin{array}{l} X \mapsto N \text{ if } X = N, \emptyset \text{ otherwise} \\ + \sum_{\substack{X_1, X_2 \\ X_1 \times X_2 = X}} F(X_1) \times F(X_2) \\ + \\ + \end{array} \right)$$

Naive categorical semantics of GADTs

data Terms : Set \rightarrow Set where

nat : $\mathbb{N} \rightarrow$ Terms \mathbb{N}

$-,_-$: $\forall \{\alpha \beta\} \rightarrow$ Terms $\alpha \rightarrow$ Terms $\beta \rightarrow$ Terms $(\alpha \times \beta)$

π_1 : $\forall \{\alpha \beta\} \rightarrow$ Terms $(\alpha \times \beta) \rightarrow$ Terms α

π_2 : $\forall \{\alpha \beta\} \rightarrow$ Terms $(\alpha \times \beta) \rightarrow$ Terms β

To interpret the type constructor Terms, take the initial algebra $\mu\mathcal{T}$ of:

$\mathcal{T} : [\mathcal{C}, \mathcal{C}] \rightarrow [\mathcal{C}, \mathcal{C}]$

$$F \mapsto \left(\begin{array}{l} X \mapsto N \text{ if } X = N, \emptyset \text{ otherwise} \\ + \sum_{\substack{X_1, X_2 \\ X_1 \times X_2 = X}} F(X_1) \times F(X_2) \\ + \sum_{X_2} F(X \times X_2) \\ + \end{array} \right)$$

Naive categorical semantics of GADTs

data Terms : Set \rightarrow Set where

nat : $\mathbb{N} \rightarrow$ Terms \mathbb{N}

$-, -$: $\forall \{\alpha \beta\} \rightarrow$ Terms $\alpha \rightarrow$ Terms $\beta \rightarrow$ Terms $(\alpha \times \beta)$

π_1 : $\forall \{\alpha \beta\} \rightarrow$ Terms $(\alpha \times \beta) \rightarrow$ Terms α

π_2 : $\forall \{\alpha \beta\} \rightarrow$ Terms $(\alpha \times \beta) \rightarrow$ Terms β

To interpret the type constructor Terms, take the initial algebra $\mu\mathcal{T}$ of:

$$\mathcal{T} : [\mathcal{C}, \mathcal{C}] \rightarrow [\mathcal{C}, \mathcal{C}]$$

$$F \mapsto \left(\begin{array}{l} X \mapsto N \text{ if } X = N, \emptyset \text{ otherwise} \\ + \sum_{\substack{X_1, X_2 \\ X_1 \times X_2 = X}} F(X_1) \times F(X_2) \\ + \sum_{X_2} F(X \times X_2) \\ + \sum_{X_1} F(X_1 \times X) \end{array} \right)$$

Naive categorical semantics of GADTs

data Terms : Set \rightarrow Set where

nat : $\mathbb{N} \rightarrow$ Terms \mathbb{N}

$-, -$: $\forall \{\alpha \beta\} \rightarrow$ Terms $\alpha \rightarrow$ Terms $\beta \rightarrow$ Terms $(\alpha \times \beta)$

π_1 : $\forall \{\alpha \beta\} \rightarrow$ Terms $(\alpha \times \beta) \rightarrow$ Terms α

π_2 : $\forall \{\alpha \beta\} \rightarrow$ Terms $(\alpha \times \beta) \rightarrow$ Terms β

To interpret the type constructor Terms, take the initial algebra $\mu\mathcal{T}$ of:

$\mathcal{T} : [\mathcal{C}, \mathcal{C}] \rightarrow [\mathcal{C}, \mathcal{C}]$

$$F \mapsto \left(\begin{array}{l} X \mapsto N \text{ if } X = N, \emptyset \text{ otherwise} \\ + \sum_{\substack{X_1, X_2 \\ X_1 \times X_2 = X}} F(X_1) \times F(X_2) \\ + \sum_{X_2} F(X \times X_2) \quad ?? \\ + \sum_{X_1} F(X_1 \times X) \quad ?? \end{array} \right)$$

Naive categorical semantics of GADTs

data Terms : Set \rightarrow Set **where**

nat : $\mathbb{N} \rightarrow$ Terms \mathbb{N}

$-, -$: $\forall \{\alpha \beta\} \rightarrow$ Terms $\alpha \rightarrow$ Terms $\beta \rightarrow$ Terms $(\alpha \times \beta)$

π_1 : $\forall \{\alpha \beta\} \rightarrow$ Terms $(\alpha \times \beta) \rightarrow$ Terms α

π_2 : $\forall \{\alpha \beta\} \rightarrow$ Terms $(\alpha \times \beta) \rightarrow$ Terms β

Actually, $\mu\mathcal{T} : \mathcal{C} \rightarrow \mathcal{C}$ being a **functor** is already an issue.

Naive categorical semantics of GADTs

data Terms : Set \rightarrow Set where

nat : $\mathbb{N} \rightarrow$ Terms \mathbb{N}

$-, -$: $\forall \{\alpha \beta\} \rightarrow$ Terms $\alpha \rightarrow$ Terms $\beta \rightarrow$ Terms $(\alpha \times \beta)$

π_1 : $\forall \{\alpha \beta\} \rightarrow$ Terms $(\alpha \times \beta) \rightarrow$ Terms α

π_2 : $\forall \{\alpha \beta\} \rightarrow$ Terms $(\alpha \times \beta) \rightarrow$ Terms β

Actually, $\mu\mathcal{T} : \mathcal{C} \rightarrow \mathcal{C}$ being a functor is already an issue.

Consider the parity function $p : \mathbb{N} \rightarrow \mathbb{B}$, interpreted by $p : N \rightarrow B$. Because of $\mu\mathcal{T}(p) : \mu\mathcal{T}(N) \rightarrow \mu\mathcal{T}(B)$, the interpretation of Terms \mathbb{B} contains weird elements...

Naive categorical semantics of GADTs

Even more striking with

```
data W : Set → Set where
```

```
  ∃ : ∀ {α} → α → W τ
```

Naive categorical semantics of GADTs

Even more striking with

data $W : \mathbf{Set} \rightarrow \mathbf{Set}$ **where**

$\exists : \forall \{\alpha\} \rightarrow \alpha \rightarrow W \tau$

If W is interpreted as the initial algebra $\mu\mathcal{W} : \mathcal{C} \rightarrow \mathcal{C}$ of a certain \mathcal{W} :

Naive categorical semantics of GADTs

Even more striking with

data $W : \mathbf{Set} \rightarrow \mathbf{Set}$ **where**

$\exists : \forall \{\alpha\} \rightarrow \alpha \rightarrow W \tau$

If W is interpreted as the initial algebra $\mu\mathcal{W} : \mathcal{C} \rightarrow \mathcal{C}$ of a certain \mathcal{W} :

- for each $X \in \mathcal{C}$,

$$\exists_X : X \rightarrow \mu\mathcal{W}(1),$$

Naive categorical semantics of GADTs

Even more striking with

data $W : \mathbf{Set} \rightarrow \mathbf{Set}$ **where**

$\exists : \forall \{ \alpha \} \rightarrow \alpha \rightarrow W \tau$

If W is interpreted as the initial algebra $\mu\mathcal{W} : \mathcal{C} \rightarrow \mathcal{C}$ of a certain \mathcal{W} :

- for each $X \in \mathcal{C}$,

$$\exists_X : X \rightarrow \mu\mathcal{W}(1),$$

- for each $x : 1 \rightarrow X$,

$$1 \xrightarrow{\exists_1} \mu\mathcal{W}(1) \xrightarrow{\mu\mathcal{W}(x)} \mu\mathcal{W}(X)$$

Naive categorical semantics of GADTs

Even more striking with

data $W : \mathbf{Set} \rightarrow \mathbf{Set}$ **where**

$\exists : \forall \{ \alpha \} \rightarrow \alpha \rightarrow W \tau$

If W is interpreted as the initial algebra $\mu W : \mathcal{C} \rightarrow \mathcal{C}$ of a certain W :

- for each $X \in \mathcal{C}$,

$$\exists_X : X \rightarrow \mu W(1),$$

- for each $x : 1 \rightarrow X$,

$$1 \xrightarrow{\exists_1} \mu W(1) \xrightarrow{\mu W(x)} \mu W(X)$$

When $\mathcal{C} = \mathbf{Set}$, $\mu W(X) \neq \emptyset$ whenever $X \neq \emptyset \dots$

Naive categorical semantics of GADTs

$$1 \xrightarrow{\exists_1} \mu\mathcal{W}(1) \xrightarrow{\mu\mathcal{W}(x)} \mu\mathcal{W}(X)$$

$$1 \xrightarrow{\exists_1} \mu\mathcal{W}(1) \xrightarrow{\mu\mathcal{W}(x)} \mu\mathcal{W}(X)$$

Issue

$\mu\mathcal{W}(x)$ has to make a new element from \exists_1 .

$$1 \xrightarrow{\exists_1} \mu\mathcal{W}(1) \xrightarrow{\mu\mathcal{W}(x)} \mu\mathcal{W}(X)$$

Issue

$\mu\mathcal{W}(x)$ has to make a new element from \exists_1 .

Potential solution

Allowing $\mu\mathcal{W}(x)$ to be only partially defined.

Less naive categorical semantics of GADTs

Definition

A structure of partiality on \mathcal{C} is a wide subcategory \mathcal{D} such that

$$\forall f, g \in \mathcal{C}, gf \in \mathcal{D} \implies f \in \mathcal{D}$$

Less naive categorical semantics of GADTs

Definition

A structure of partiality on \mathcal{C} is a wide subcategory \mathcal{D} such that

$$\forall f, g \in \mathcal{C}, gf \in \mathcal{D} \implies f \in \mathcal{D}$$

Example

- $\mathcal{D} = \mathcal{C}$ (trivial)

Less naive categorical semantics of GADTs

Definition

A structure of partiality on \mathcal{C} is a wide subcategory \mathcal{D} such that

$$\forall f, g \in \mathcal{C}, gf \in \mathcal{D} \implies f \in \mathcal{D}$$

Example

- $\mathcal{D} = \mathcal{C}$ (trivial)
- $\mathcal{C} = \mathbf{PSet}$ and $\mathcal{D} = \mathbf{Set}$

Less naive categorical semantics of GADTs

Definition

A structure of partiality on \mathcal{C} is a wide subcategory \mathcal{D} such that

$$\forall f, g \in \mathcal{C}, gf \in \mathcal{D} \implies f \in \mathcal{D}$$

Example

- $\mathcal{D} = \mathcal{C}$ (trivial)
- $\mathcal{C} = \mathbf{PSet}$ and $\mathcal{D} = \mathbf{Set}$
- $\mathcal{C} = \mathbf{CPO}$ and $\mathcal{D} = \mathbf{CPO}_{\text{tot}}$

Less naive categorical semantics of GADTs

Definition

A structure of partiality on \mathcal{C} is a wide subcategory \mathcal{D} such that

$$\forall f, g \in \mathcal{C}, gf \in \mathcal{D} \implies f \in \mathcal{D}$$

Example

- $\mathcal{D} = \mathcal{C}$ (trivial)
- $\mathcal{C} = \mathbf{PSet}$ and $\mathcal{D} = \mathbf{Set}$
- $\mathcal{C} = \mathbf{CPO}$ and $\mathcal{D} = \mathbf{CPO}_{\text{tot}}$

Idea

Interpret the total functions of the language in \mathcal{D} and “spill” in \mathcal{C} for partial functions.

Less naive categorical semantics of GADTs

But...

Less naive categorical semantics of GADTs

But...

```
data _≡_ : Set → Set → Set where  
  r : ∀ α → α ≡ α
```

Less naive categorical semantics of GADTs

But...

```
data _≡_ : Set → Set → Set where
  r : ∀ α → α ≡ α
```

- `_≡_` interpreted as: an object $(X \equiv Y)$ for every $X, Y \in \mathcal{C}$

Less naive categorical semantics of GADTs

But...

```
data _≡_ : Set → Set → Set where
```

```
  r : ∀ α → α ≡ α
```

- `_≡_` interpreted as: an object $(X \equiv Y)$ for every $X, Y \in \mathcal{C}$
- `r` interpreted as: a morphism $r_X : 1 \rightarrow (X \equiv X)$ in \mathcal{D} for every $X \in \mathcal{C}$

Less naive categorical semantics of GADTs

But...

```
data _≡_ : Set → Set → Set where
  r : ∀ α → α ≡ α
```

- `_≡_` interpreted as: an object $(X \equiv Y)$ for every $X, Y \in \mathcal{C}$
- `r` interpreted as: a morphism $r_X : 1 \rightarrow (X \equiv X)$ in \mathcal{D} for every $X \in \mathcal{C}$

If `_ ≡ _` extends to a functor $\mathcal{C}^2 \rightarrow \mathcal{C}$: for any $x : 1 \rightarrow X$,

Less naive categorical semantics of GADTs

But...

data $_ \equiv _$: **Set** \rightarrow **Set** \rightarrow **Set** **where**

r : $\forall \alpha \rightarrow \alpha \equiv \alpha$

- $_ \equiv _$ interpreted as: an object $(X \equiv Y)$ for every $X, Y \in \mathcal{C}$
- r interpreted as: a morphism $r_X : 1 \rightarrow (X \equiv X)$ in \mathcal{D} for every $X \in \mathcal{C}$

If $_ \equiv _$ extends to a functor $\mathcal{C}^2 \rightarrow \mathcal{C}$: for any $x : 1 \rightarrow X$,

$$p_x : 1 \xrightarrow{r_1} 1 \equiv 1 \xrightarrow{(x \equiv \text{id}_1)} (X \equiv 1)$$

Less naive categorical semantics of GADTs

$\text{trp} : \forall \alpha \beta \rightarrow \alpha \equiv \beta \rightarrow \alpha \rightarrow \beta$

$\text{trp } \alpha \ \alpha \ r \ x = x$

$\text{trp}^{-1} : \forall \alpha \beta \rightarrow \alpha \equiv \beta \rightarrow \beta \rightarrow \alpha$

$\text{trp } \beta \ \beta \ r \ y = y$

Less naive categorical semantics of GADTs

$\text{trp} : \forall \alpha \beta \rightarrow \alpha \equiv \beta \rightarrow \alpha \rightarrow \beta$

$\text{trp } \alpha \alpha \ r \ x = x$

$\text{trp}^{-1} : \forall \alpha \beta \rightarrow \alpha \equiv \beta \rightarrow \beta \rightarrow \alpha$

$\text{trp } \beta \beta \ r \ y = y$

- trp interpreted as: a morphism $t_{X,Y} : (X \equiv Y) \times X \rightarrow Y$ in \mathcal{D} for every $X, Y \in \mathcal{C}$

Less naive categorical semantics of GADTs

$$\text{trp} : \forall \alpha \beta \rightarrow \alpha \equiv \beta \rightarrow \alpha \rightarrow \beta$$
$$\text{trp } \alpha \alpha \ r \ x = x$$
$$\text{trp}^{-1} : \forall \alpha \beta \rightarrow \alpha \equiv \beta \rightarrow \beta \rightarrow \alpha$$
$$\text{trp } \beta \beta \ r \ y = y$$

- trp interpreted as: a morphism $t_{X,Y} : (X \equiv Y) \times X \rightarrow Y$ in \mathcal{D} for every $X, Y \in \mathcal{C}$

Less naive categorical semantics of GADTs

$\text{trp} : \forall \alpha \beta \rightarrow \alpha \equiv \beta \rightarrow \alpha \rightarrow \beta$

$\text{trp } \alpha \alpha \ r \ x = x$

$\text{trp}^{-1} : \forall \alpha \beta \rightarrow \alpha \equiv \beta \rightarrow \beta \rightarrow \alpha$

$\text{trp } \beta \beta \ r \ y = y$

- trp interpreted as: a morphism $t_{X,Y} : (X \equiv Y) \times X \rightarrow Y$ in \mathcal{D} for every $X, Y \in \mathcal{C}$
- trp^{-1} interpreted as: a morphism $t_{X,Y}^{-1} : (X \equiv Y) \times Y \rightarrow X$ in \mathcal{D} for every $X, Y \in \mathcal{C}$

Less naive categorical semantics of GADTs

$\text{trp} : \forall \alpha \beta \rightarrow \alpha \equiv \beta \rightarrow \alpha \rightarrow \beta$

$\text{trp } \alpha \alpha \ r \ x = x$

$\text{trp}^{-1} : \forall \alpha \beta \rightarrow \alpha \equiv \beta \rightarrow \beta \rightarrow \alpha$

$\text{trp } \beta \beta \ r \ y = y$

- trp interpreted as: a morphism $t_{X,Y} : (X \equiv Y) \times X \rightarrow Y$ in \mathcal{D} for every $X, Y \in \mathcal{C}$
- trp^{-1} interpreted as: a morphism $t_{X,Y}^{-1} : (X \equiv Y) \times Y \rightarrow X$ in \mathcal{D} for every $X, Y \in \mathcal{C}$
- $(\lambda p \ x \rightarrow \text{trp}^{-1} \ p \ (\text{trp} \ p \ x))$ reduces to $(\lambda p \ x \rightarrow x)$:

$$\begin{array}{ccc} (X \equiv Y) \times X & \xrightarrow{\langle \pi_1, t_{X,Y} \rangle} & (X \equiv Y) \times Y \\ & \searrow \pi_2 & \downarrow t_{X,Y}^{-1} \\ & & X \end{array}$$

Less naive categorical semantics of GADTs

$$\begin{array}{ccc} (X \equiv Y) \times X & \xrightarrow{\langle \pi_1, t_{X,Y} \rangle} & (X \equiv Y) \times Y \\ & \searrow \pi_2 & \downarrow t_{X,Y}^{-1} \\ & & X \end{array}$$

Less naive categorical semantics of GADTs

$$\begin{array}{ccc} (X \equiv 1) \times X & \xrightarrow{\langle \pi_1, t_{X,1} \rangle} & (X \equiv 1) \times 1 \\ & \searrow \pi_2 & \downarrow t_{X,1}^{-1} \\ & & X \end{array}$$

Less naive categorical semantics of GADTs

$$\begin{array}{ccccc} X & \xrightarrow{\langle p_x \circ !, \text{id}_X \rangle} & (X \equiv 1) \times X & \xrightarrow{\langle \pi_1, t_{X,1} \rangle} & (X \equiv 1) \times 1 \\ & & & \searrow \pi_2 & \downarrow t_{X,1}^{-1} \\ & & & & X \end{array}$$

Less naive categorical semantics of GADTs

$$\begin{array}{ccccc} X & \xrightarrow{\langle p_x \circ !, \text{id}_X \rangle} & (X \equiv 1) \times X & \xrightarrow{\langle \pi_1, t_{X,1} \rangle} & (X \equiv 1) \times 1 \\ & \searrow \text{id}_X & & & \downarrow t_{X,1}^{-1} \\ & & & & X \end{array}$$

Less naive categorical semantics of GADTs

$$\begin{array}{ccccc} & & & & 1 \\ & & & & \downarrow \langle p_x, \text{id}_1 \rangle \\ X & \xrightarrow{\langle p_x \circ !, \text{id}_X \rangle} & (X \equiv 1) \times X & \xrightarrow{\langle \pi_1, t_{X,1} \rangle} & (X \equiv 1) \times 1 \\ & \searrow \text{id}_X & & & \downarrow t_{X,1}^{-1} \\ & & & & X \end{array}$$

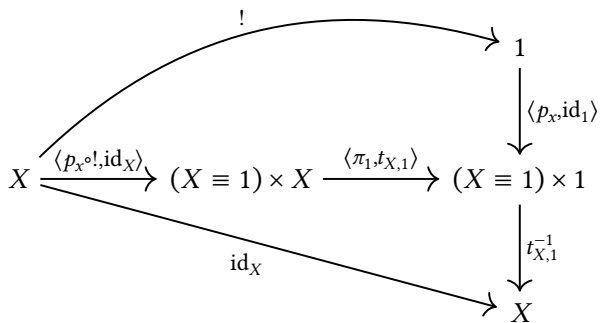
The diagram illustrates a commutative square in a categorical semantics of GADTs. The objects are X , $(X \equiv 1) \times X$, $(X \equiv 1) \times 1$, and X . The top-left corner is X , the top-right is 1 , the bottom-left is $(X \equiv 1) \times X$, and the bottom-right is X . The horizontal arrows are $\langle p_x \circ !, \text{id}_X \rangle$ and $\langle \pi_1, t_{X,1} \rangle$. The vertical arrows are $\langle p_x, \text{id}_1 \rangle$ and $t_{X,1}^{-1}$. A curved arrow labeled $!$ points from X to 1 . A diagonal arrow labeled id_X points from X to X .

Less naive categorical semantics of GADTs

$$\begin{array}{ccccc} & & & & 1 \\ & & & \searrow & \downarrow \langle p_x, \text{id}_1 \rangle \\ & & & & (X \equiv 1) \times 1 \\ X & \xrightarrow{\langle p_x \circ !, \text{id}_X \rangle} & (X \equiv 1) \times X & \xrightarrow{\langle \pi_1, t_{X,1} \rangle} & \\ & \searrow \text{id}_X & & & \downarrow t_{X,1}^{-1} \\ & & & & X \end{array}$$

$\implies X$ is a retract of 1

Less naive categorical semantics of GADTs



$\implies X \simeq 1$ whenever $t_{X,1}^{-1} \langle p_x, \text{id}_1 \rangle$ in \mathcal{D}

Less naive categorical semantics of GADTs

$$\begin{array}{ccccc} & & & & 1 \\ & & & \searrow & \downarrow \langle p_x, \text{id}_1 \rangle \\ & & & & (X \equiv 1) \times 1 \\ X & \xrightarrow{\langle p_x \circ !, \text{id}_X \rangle} & (X \equiv 1) \times X & \xrightarrow{\langle \pi_1, t_{X,1} \rangle} & (X \equiv 1) \times 1 \\ & \searrow \text{id}_X & & & \downarrow t_{X,1}^{-1} \\ & & & & X \\ & & & \nearrow & \\ & & & & 1 \end{array}$$

$\implies X \simeq 1$ whenever p_x in \mathcal{D}

Less naive categorical semantics of GADTs

$$\begin{array}{ccccc} & & & & 1 \\ & & & & \downarrow \langle p_x, \text{id}_1 \rangle \\ & & & & (X \equiv 1) \times 1 \\ X & \xrightarrow{\langle p_x \circ !, \text{id}_X \rangle} & (X \equiv 1) \times X & \xrightarrow{\langle \pi_1, t_{X,1} \rangle} & \\ & \searrow \text{id}_X & & & \downarrow t_{X,1}^{-1} \\ & & & & X \end{array}$$

The diagram shows a commutative square with an additional arrow. The top-left node is X . The top-right node is 1 . The bottom-left node is $(X \equiv 1) \times X$. The bottom-right node is $(X \equiv 1) \times 1$. The bottom-most node is X . Arrows are: $X \rightarrow 1$ (curved, labeled $!$), $X \rightarrow (X \equiv 1) \times X$ (labeled $\langle p_x \circ !, \text{id}_X \rangle$), $(X \equiv 1) \times X \rightarrow (X \equiv 1) \times 1$ (labeled $\langle \pi_1, t_{X,1} \rangle$), $(X \equiv 1) \times 1 \rightarrow X$ (labeled $t_{X,1}^{-1}$), and $X \rightarrow X$ (labeled id_X). There is also a vertical arrow from 1 to $(X \equiv 1) \times 1$ labeled $\langle p_x, \text{id}_1 \rangle$.

$\implies X \simeq 1$ whenever x in \mathcal{D}

Theorem

If GADTs' interpretations extend to functors, the interpretation of any non-empty closed type is trivial.

Thank you.