Build, Augment and Destroy. Universally

Varmo Vene (Uni. Tartu, Estonia)

Joint work with

Neil Ghani (Uni. Leicester, UK)

Tarmo Uustalu (Inst. of Cybernetics, Tallinn, Estonia)

Teooriapäevad, Veskisilla, 1 Oct 2004

Motivation

- The usual in/fold paradigm of programming with inductive types is very elegant and useful since it is directly based on the initial algebra semantics,
  a universal construction, syntax and equational laws for programming and reasoning follow directly.
- But in functional programming, in shortcut deforestation, one also uses build and fold/build fusion, the semantics has been unclear.
- Related is the question about the adequacy of the "impredicative encoding" of inductive types (Freyd, Wadler, Hasegawa. . .)

Background

- Gill, Launchbury, Peyton Jones (1993) –
  - `build` to capture uniform production of lists.
  - `foldr/build` fusion to eliminate intermediate lists.
  - correctness "proved" informally by reference to "theorems for free"
- Gill (1996) – `augment` for lists
- Takano, Meijer (1995) – `build` for arbitrary inductive types
- Johann (2002, 2003) – correctness proof via parametricity of contextual equivalence

Shortcut deforesation

- Program transformation for automatic removal of intermediate data structures
- Uses `foldr` as standard list processing function

```
foldr :: (a -> b -> b) -> b -> [a] -> b
foldr f b []     = b
foldr f b (x:xs) = f x (foldr f b xs)
```

- List producers are defined using `build`

```
build :: (forall x . (a -> x -> x) -> x -> x) -> [a]
build theta = theta (:) []
```

- `foldr/build` fusion

```
foldr f b (build theta)   ==>  theta f b
```

Shortcut deforesation: example

- Modular (but inefficient) sum of squares:

```
sumSq :: Int -> Int
sumSq m = sum (map square (upto 1 m))
```

- Definitions of `sum`, `map` and `upto`

```
sum = foldr (+) 0
map f xs = build (\ c n ->
                 foldr (\ x ys -> f x ‘c‘ ys) n xs)
upto i1 i2 = build (\ c n -> upto’ c n i1 i2)
upto’ c n i1 i2
     = if i1 > i2 then n
                  else i1 ‘c‘ upto’ c n (i1 + 1) i2
```

Shortcut deforesation: example

- Transformation

```
sum (map square (upto 1 m))
==> foldr (+) 0 (build (\ c n ->
          foldr (\ x ys -> square x ‘c‘ ys) n (upto 1 m)))
==> foldr (\ x ys -> square x + ys) 0 (upto 1 m)
==> foldr (\ x ys -> square x + ys) 0
                          (build (\ c n -> upto’ c n 1 m)
==> upto’ (\ x ys -> square x + ys) 0 1 m
```

- Efficient sum of squares

```
sumSq :: Int -> Int
sumSq m = sumSq’ 1 m
  where sumSq’ i1 i2
             = if i1 > i2 then 0
               else square i1 + (sumSq’ (i1 + 1) i2)
```

Initial algebras

- Let $\mathcal{C}$ be a category and $F : \mathcal{C} \to \mathcal{C}$ be a functor. An F-algebra is an object X in $\mathcal{C}$ together with a map $\varphi : F\ X \to X$ in $\mathcal{C}$. An F-algebra map $(X, \varphi) \to (Y, \psi)$ is a map $f : X \to Y$ such that the square

$$\begin{array}{ccc} FX & \xrightarrow{\varphi} & X \\ {\scriptstyle Ff}\downarrow & & \downarrow{\scriptstyle f} \\ FY & \xrightarrow[\psi]{} & Y \end{array}$$

  commutes. An initial F-algebra is an initial object in the category F-**alg** of F-algebras, i.e., an F-algebra with a unique map from it to any F-algebra.

Initial algebras

- Syntax:

$$\mathsf{in}_F : F(\mu F) \to \mu F \qquad \frac{(X, \varphi) \in F\text{-}\mathbf{alg}}{\mathsf{fold}_{F,X}\varphi : \mu F \to X}$$

- Evaluation:

$$\frac{(X, \varphi) \in F\text{-}\mathbf{alg}}{\mathsf{fold}_{F,X}\varphi \circ \mathsf{in}_F = \varphi \circ F\mathsf{fold}_{F,X}\varphi}$$

- Extensionality:

$$\mathsf{fold}_{F,F(\mu F)}\mathsf{in}_F = \mathsf{id}_{\mu F} \qquad \frac{f : (X, \varphi) \to (Y, \psi) \in F\text{-}\mathbf{alg}}{f \circ \mathsf{fold}_{F,X}\varphi = \mathsf{fold}_{F,Y}\psi}$$

Building build: pre-draft

- Type:

$$\frac{\Theta \ : \ \forall X.(FX \to X) \to (C \to X)}{\mathsf{build}_{F,C}\Theta : C \to \mu F}$$

- Definition:

$$\frac{\Theta \ : \ \forall X.(FX \to X) \to (C \to X)}{\mathsf{build}_{F,C}\Theta = \Theta\mathsf{in}_F}$$

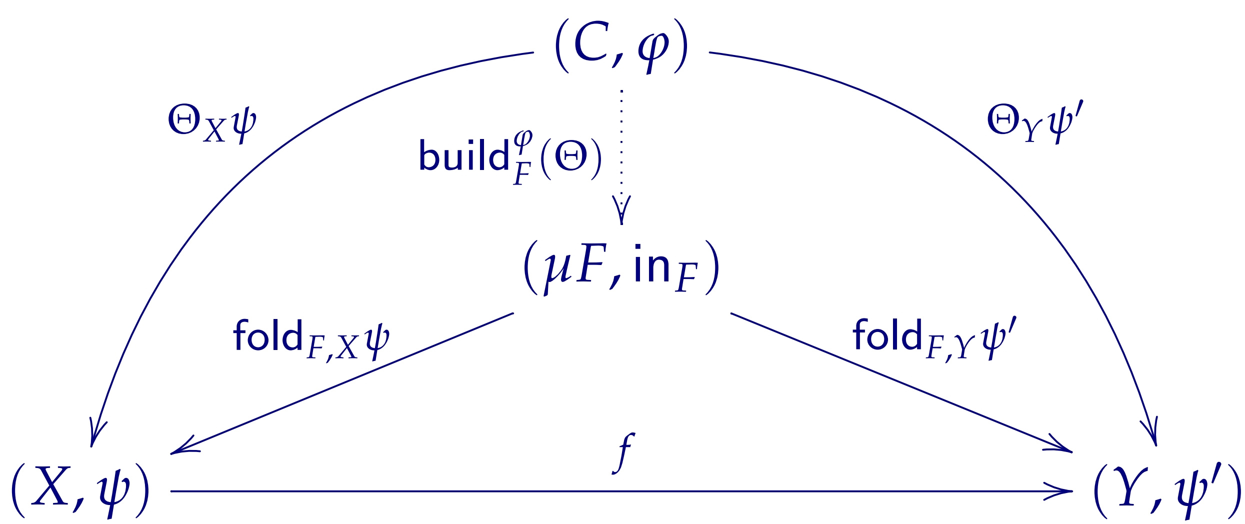
- Shortcut deforestation:

$$\frac{\Theta \ : \ \forall X.(FX \to X) \to (C \to X) \quad \varphi \ : \ FA \to A}{\mathsf{fold}_{F,X}\varphi \circ \mathsf{build}_{F,C}\Theta = \Theta\varphi}$$

Building build: 1st attempt

- `Prop.` Let $\mathcal{C}$ be a category. If $\mathcal{C}$ has an initial object 0, then the limit of the identity functor $\mathsf{Id} : \mathcal{C} \to \mathcal{C}$ is 0. Conversely if the identity functor has a limit, then this is the initial object of $\mathcal{C}$.

- `Cor.` A functor $F : \mathcal{C} \to \mathcal{C}$ has an initial algebra $(\mu F, \mathsf{in}_F)$ iff $(\mu F, \mathsf{in}_F)$ is a limit of the identity functor $\mathsf{Id} : F\text{-}\mathbf{alg} \to F\text{-}\mathbf{alg}$.



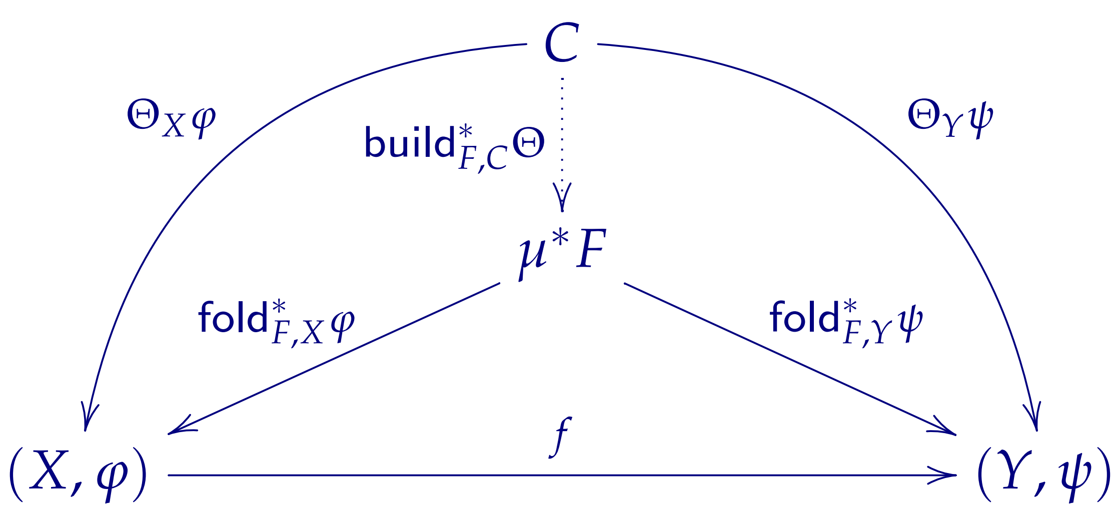
Building build: 2nd attempt

- Let C be a category and $F : \mathcal{C} \to \mathcal{C}$ be a functor.
- Let $U_F : F\text{-}\mathbf{alg} \to \mathcal{C}$ be a forgetful functor.
- A U_F-cone is an object C in $\mathcal{C}$ and, for any F-algebra (X, φ), a map $\Theta_X \varphi : C \to X$ in $\mathcal{C}$, such that for any F-algebra map $f : (X, \varphi) \to (Y, \psi)$
$$f \circ \Theta_X \varphi = \Theta_Y \psi$$
- A U_F-cone map $h : (C, \Theta) \to (D, \Xi)$ is a map $h : C \to D$ in $\mathcal{C}$ such that, for any F-algebra (X, φ)
$$\Xi_X \varphi \circ h = \Theta_X \varphi$$
- A U_F-limit is a final object in the category of U_F-cones.

Building build: 2nd attempt

- Syntax:

$$\frac{(X,\varphi) \in F\text{-}\mathbf{alg}}{\mathsf{fold}^*_{F,X}\varphi : \mu^* F \to X} \qquad \frac{(C,\Theta) \in U_F\text{-}\mathbf{cone}}{\mathsf{build}^*_{F,C}\Theta : C \to \mu^* F}$$



Building build: 2nd attempt

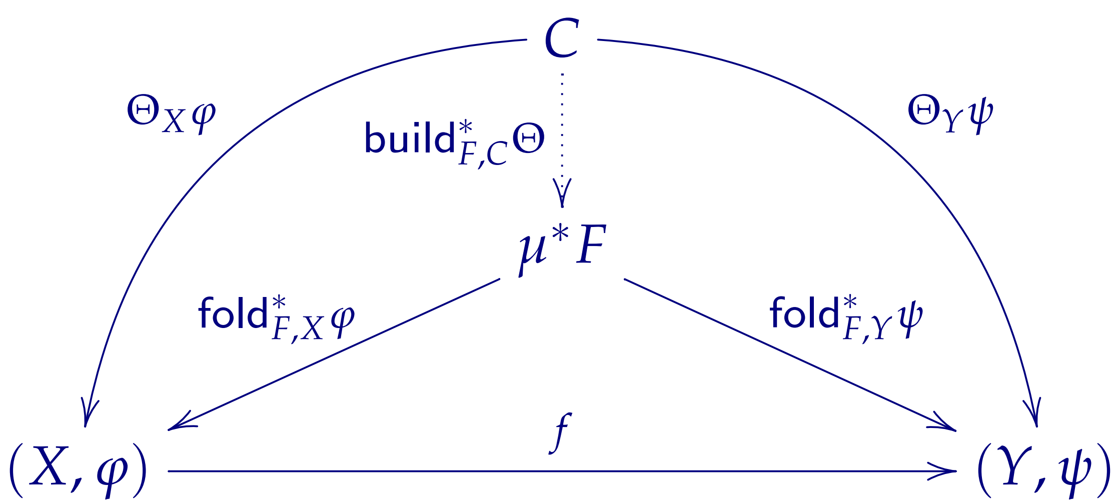
- Laws:

$$\frac{f : (X, \varphi) \to (Y, \psi) \in F\text{-}\mathbf{alg}}{f \circ \mathsf{fold}^*_{F,X}\varphi = \mathsf{fold}^*_{F,Y}\psi}$$

$$\frac{(C, \Theta) \in U_F\text{-}\mathbf{cone} \quad (X, \varphi) \in F\text{-}\mathbf{alg}}{\mathsf{fold}^*_{F,X}\varphi \circ \mathsf{build}^*_{F,C}\Theta = \Theta_X\varphi}$$

$$\mathsf{id}_{\mu^* F} = \mathsf{build}^*_{F,\mu F}\mathsf{fold}^*_F$$

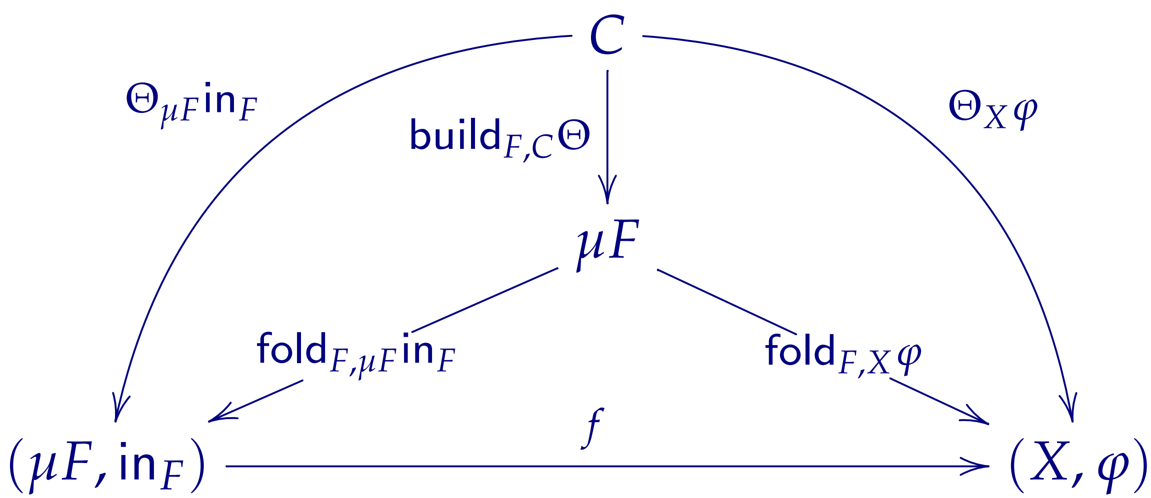
$$\frac{h : (C, \Theta) \to (D, \Xi) \in U_F\text{-}\mathbf{cone}}{\mathsf{build}^*_{F,C}\Theta = \mathsf{build}^*_{F,D}\Xi \circ h}$$



Building build: 2nd attempt

- **Prop.** Let C be a category and $F : \mathcal{C} \to \mathcal{C}$ be a functor. If there is an initial F-algebra $(\mu F, \mathsf{in}_F)$, then $(\mu F, \mathsf{fold}_F)$ is an U_F-limit.
- For any U_F-cone (C, Θ), define

$$\mathsf{build}_{F,C}\Theta \quad =_{\mathrm{df}} \quad \Theta_{\mu F}\ \mathsf{in}_F \quad : \quad C \to \mu F$$

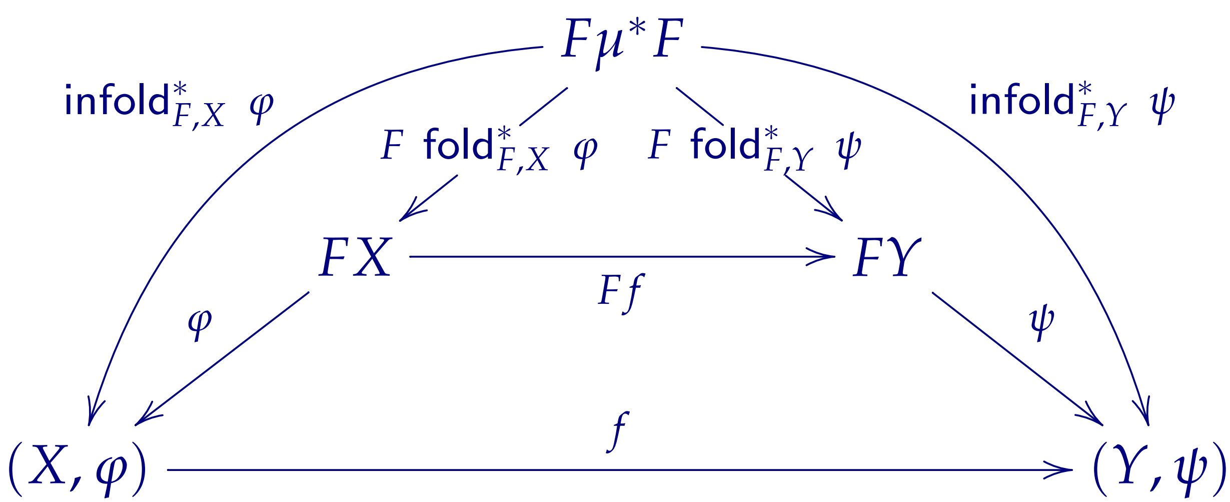


Building build: 2nd attempt

- **Prop.** Let C be a category and $F : \mathcal{C} \to \mathcal{C}$ be a functor. If there is a U_F-limit $(\mu^* F, \mathsf{fold}^*_F)$, then $\mu^* F$ is a carrier of an initial F-algebra.

- For any F-algebra (X, φ), define

$$\mathsf{infold}^*_{F,X}\ \varphi \quad =_{\mathrm{df}} \quad \varphi \circ F\ \mathsf{fold}^*_{F,X}\ \varphi \quad : \quad F\,\mu^* F \to X$$



Building build: 2nd attempt

- **Prop.** Let C be a category and $F : \mathcal{C} \to \mathcal{C}$ be a functor. If there is a U_F-limit $(\mu^* F, \mathsf{fold}^*_F)$, then $\mu^* F$ is a carrier of an initial F-algebra.
- Define

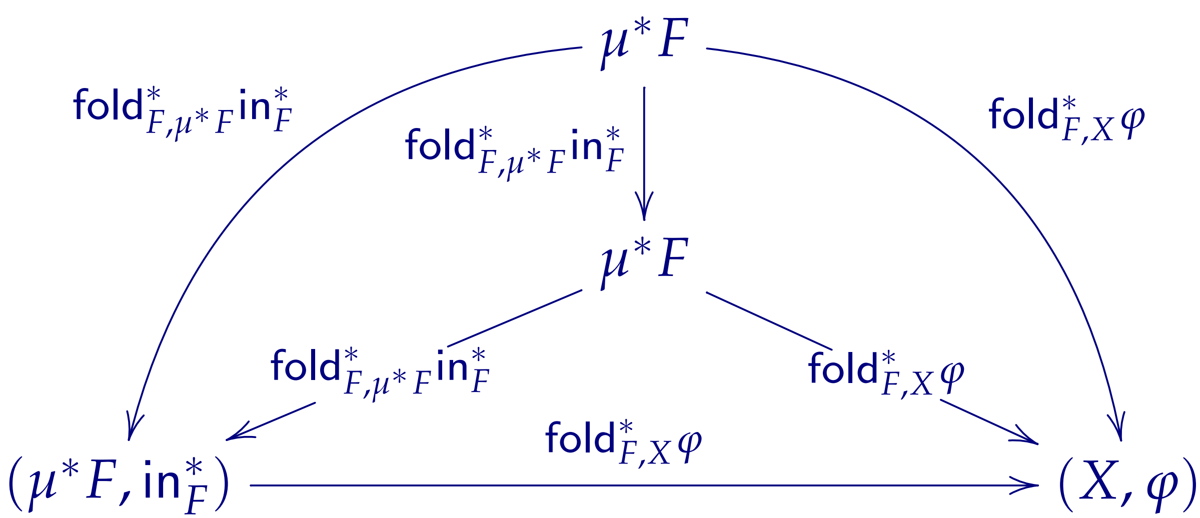
$$\mathsf{in}^*_F \quad =_{\mathrm{df}} \quad \mathsf{build}^*_{F,F(\mu^* F)} \mathsf{infold}^*_F \quad : \quad F\,\mu^* F \to \mu^* F$$

$$\begin{array}{ccc}
F\mu^* F & \xrightarrow{\mathsf{in}^*_F} & \mu^* F \\
{\scriptstyle F\mathsf{fold}^*_{F,X}\varphi}\downarrow & & \downarrow{\scriptstyle \mathsf{fold}^*_{F,X}\varphi} \\
FX & \xrightarrow{\varphi} & X
\end{array}$$

Building build: 2nd attempt

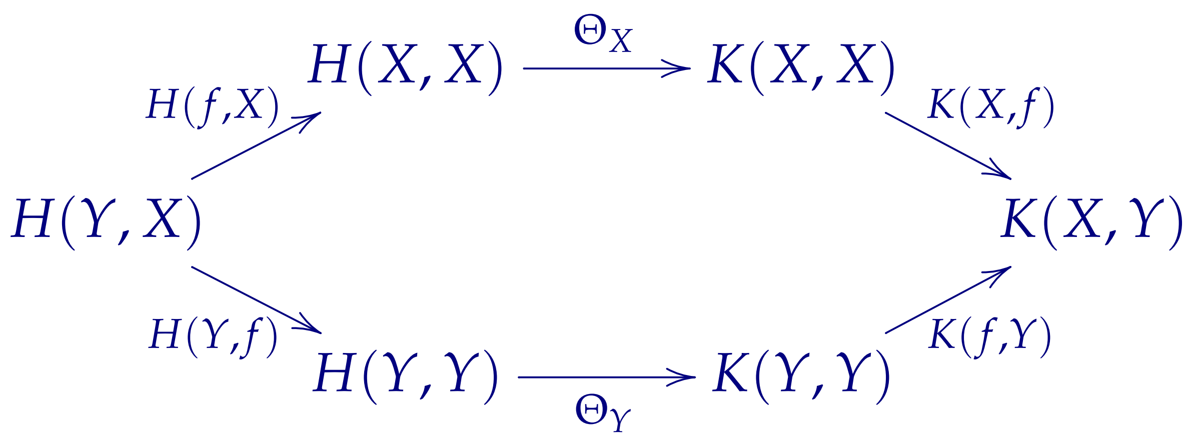
- **Prop.** Let C be a category and $F : \mathcal{C} \to \mathcal{C}$ be a functor. If there is a U_F-limit $(\mu^* F, \mathsf{fold}^*_F)$, then $\mu^* F$ is a carrier of an initial F-algebra.
- Define

$$\mathsf{in}^*_F \quad =_{\mathrm{df}} \quad \mathsf{build}^*_{F,F(\mu^* F)} \mathsf{infold}^*_F \quad : \quad F\,\mu^* F \to \mu^* F$$



From U_F-cones to strong dinaturals

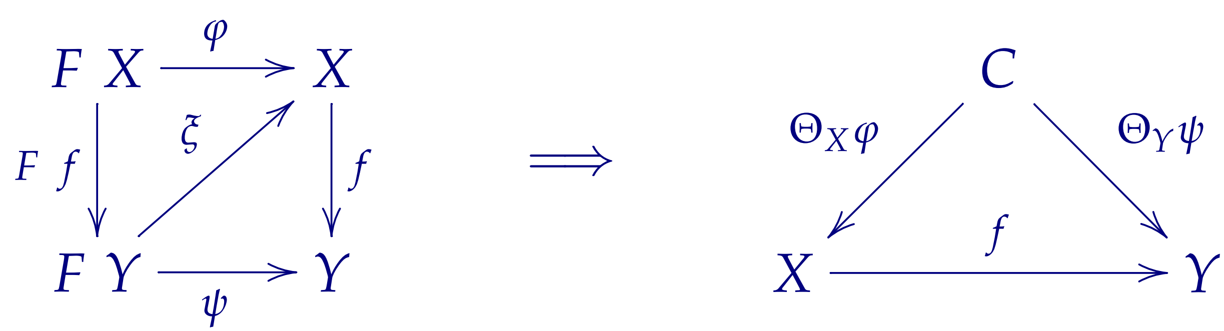
- Let $H, K : \mathcal{C}^{\mathrm{op}} \times \mathcal{C} \to \mathcal{D}$ be functors. A dinatural transformation $\Theta : H \to K$ is a family of maps $\Theta_X : H(X, X) \to K(X, X)$ for all objects X in $\mathcal{C}$ such that, for every map $f : X \to Y$ in $\mathcal{C}$, the following hexagon commutes:



From U_F-cones to strong dinaturals

In our case:

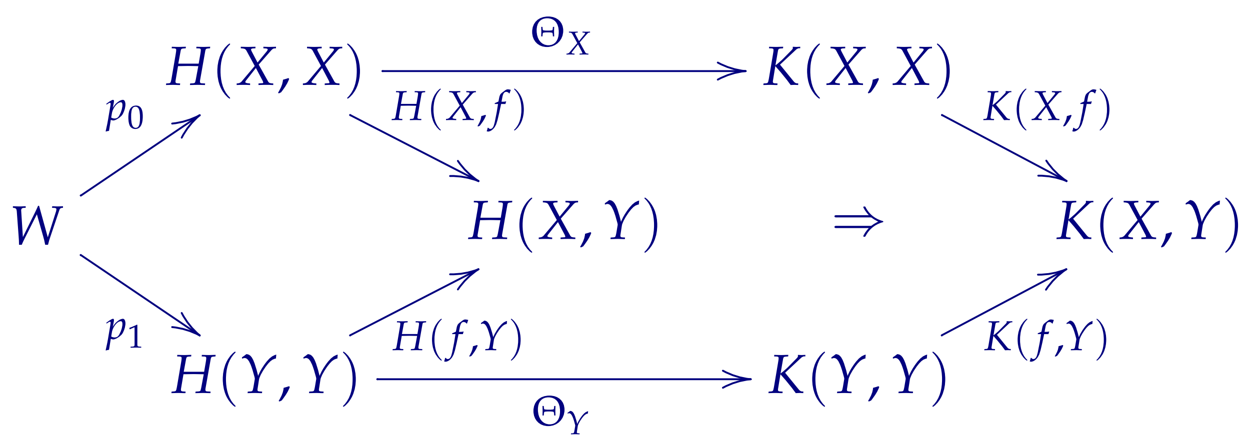
- $\mathcal{C}$ is locally small category and $\mathcal{D} = \mathbf{Set}$
- $H = \mathsf{Hom}(F\ -, -) : \mathcal{C}^{\mathrm{op}} \times \mathcal{C} \to \mathbf{Set}$ for some functor $F : \mathcal{C} \to \mathcal{C}$
- $K = \mathsf{Hom}(C, -) : \mathcal{C} \to \mathbf{Set}$ for some object C in $\mathcal{C}$.
- Dinaturality says: for any maps $f : X \to Y$, $\xi : F\ Y \to X$, $\varphi : F\ X \to X$, $\psi : F\ Y \to Y$



- Not quite right !?

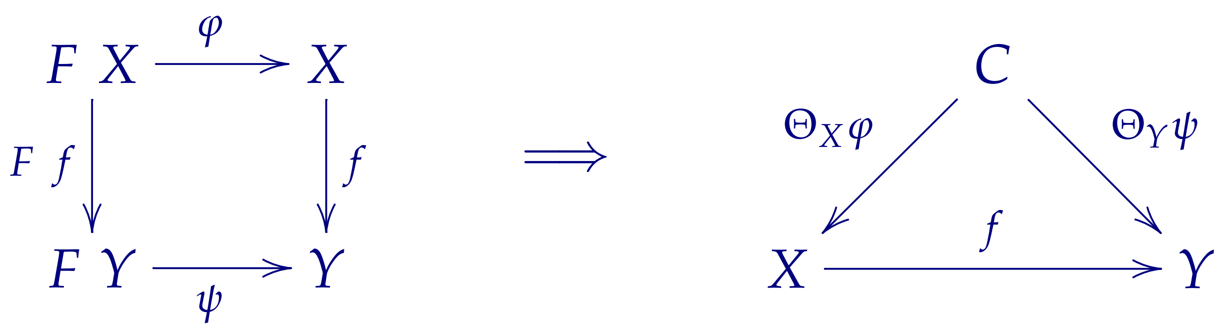
From U_F-cones to strong dinaturals

- Let $H, K : \mathcal{C}^{op} \times \mathcal{C} \to \mathcal{D}$ be functors. A strongly dinatural transformation $\Theta : H \to K$ is a family of maps $\Theta_X : H(X,X) \to K(X,X)$ for all objects X in $\mathcal{C}$ such that, for every map $f : X \to Y$, object W in $\mathcal{D}$ and maps $p_0 : W \to H(X,X)$, $p_1 : W \to H(Y,Y)$, if the square in the following diagram commutes, then so does the hexagon:



From U_F-cones to strong dinaturals

- $\mathcal{C}$ is locally small category and $\mathcal{D} = \mathbf{Set}$
- $H = \mathsf{Hom}(F\ -, -) : \mathcal{C}^{\mathrm{op}} \times \mathcal{C} \to \mathbf{Set}$ for some functor $F : \mathcal{C} \to \mathcal{C}$
- $K = \mathsf{Hom}(C, -) : \mathcal{C} \to \mathbf{Set}$ for some object C in $\mathcal{C}$.
- Strong dinaturality says: for any maps $f : X \to Y$, $\varphi : F\ X \to X$, $\psi : F\ Y \to Y$



Conclusions and future work

- Done: Alternative semantics of inductive types as limits of forgetful functor.
- Also: Derivation and generalization of `augment` combinator.
- Dualizes for coinductive types.
- To do: Parametricity in terms of strong dinaturals for languages supporting interleaved inductive and coinductive types