Practical Parametricity for GADTs

ANONYMOUS AUTHOR(S)

Abstract goes here

Maybe develop our theory for *any* $\lambda \geq \omega_1$, and then specialize to ω_1 when discussing GADTs? Can we do that? It seems we really use properties of ωCPO to get that interpretations of Nat-types are well-defined.

1 THE CALCULUS

1.1 Types

For each $k \geq 0$, we assume countable sets $\mathbb{T}^k$ of *type constructor variables of arity* k and $\mathbb{F}^k$ of *functorial variables of arity* k, all mutually disjoint. The sets of all type constructor variables and functorial variables are $\mathbb{T} = \bigcup_{k\geq 0} \mathbb{T}^k$ and $\mathbb{F} = \bigcup_{k\geq 0} \mathbb{F}^k$, respectively, and a *type variable* is any element of $\mathbb{T} \cup \mathbb{F}$. We use lower case Greek letters for type variables, writing ϕ^k to indicate that $\phi \in \mathbb{T}^k \cup \mathbb{F}^k$, and omitting the arity indicator k when convenient, unimportant, or clear from context. We reserve letters from the beginning of the alphabet to denote type variables of arity 0, i.e., elements of $\mathbb{T}^0 \cup \mathbb{F}^0$. We write $\overline{\zeta}$ for either a set $\{\zeta_1, ..., \zeta_n\}$ of type constructor variables or a set of functorial variables when the cardinality n of the set is unimportant or clear from context. If P is a set of type variables we write $P, \overline{\phi}$ for $P \cup \overline{\phi}$ when $P \cap \overline{\phi} = \emptyset$. We omit the vector notation for a singleton set, thus writing ϕ, instead of $\overline{\phi}$, for $\{\phi\}$.

Definition 1. *Let V be a finite subset of $\mathbb{T}$, P be a finite subset of $\mathbb{F}$, $\overline{\alpha}$ be a finite subset of $\mathbb{F}^0$ disjoint from P, and $\phi^k \in \mathbb{F}^k \setminus P$. The set $\mathcal{F}^P(V)$ of* functorial expressions *over P and V are given by*

$$\begin{aligned}\mathcal{F}^P(V) \;::=\; & \mathbb{0} \mid \mathbb{1} \mid \mathsf{Nat}^P \mathcal{F}^P(V)\, \mathcal{F}^P(V) \mid P\,\overline{\mathcal{F}^P(V)} \mid V\,\overline{\mathcal{F}^P(V)} \mid \mathcal{F}^P(V) + \mathcal{F}^P(V) \\ & \mid \mathcal{F}^P(V) \times \mathcal{F}^P(V) \mid \left(\mu\phi^k.\lambda\alpha_1...\alpha_k.\mathcal{F}^{P,\alpha_1,\ldots,\alpha_k,\phi}(V)\right)\overline{\mathcal{F}^P(V)} \\ & \mid (\mathsf{Lan}^{\overline{\alpha}}_{\overline{\mathcal{F}^{\overline{\alpha}}}} \mathcal{F}^{P,\overline{\alpha}})\overline{\mathcal{F}^P}\end{aligned}$$

A *type* over P and V is any element of $\mathcal{F}^P(V)$. The difference with [Johann et al. 2020] here lies solely in the incorporation of functorial expressions constructed from Lan.

The notation for types entails that an application $FF_1...F_k$ is allowed only when F is a type variable of arity k, or F is a subexpression of the form $\mu\phi^k.\lambda\alpha_1...\alpha_k.F'$ or $\mathsf{Lan}^{\overline{\alpha}}_{\overline{K}} F'$. Moreover, if F has arity k then F must be applied to exactly k arguments. Accordingly, an overbar indicates a sequence of subexpressions whose length matches the arity of the type applied to it. The fact that types are always in *η-long normal form* avoids having to consider β-conversion at the level of types. In a subexpression $\mathsf{Nat}^{\Phi} F\, G$, the Nat operator binds all occurrences of the variables in Φ in F and G. Note that, by contrast with [Johann et al. 2020], variables of arity greater than 0 are allowed in Φ; this is necessary to construct well-typed terms of Lan types. In a subexpression $\mu\phi^k.\lambda\overline{\alpha}.F$, the μ operator binds all occurrences of the variable ϕ, and the λ operator binds all occurrences of the variables in $\overline{\alpha}$, in the body F. And in a subexpression $(\mathsf{Lan}^{\overline{\alpha}}_{\overline{K}} F)\overline{A}$, the Lan operator binds all occurrences of the variables in $\overline{\alpha}$ in every element of $\overline{K}$, as well as in F.

2020. 2475-1421/2020/1-ART1 $15.00
https://doi.org/

A *type constructor context* is a finite set Γ of type constructor variables, and a *functorial context* is a finite set Φ of functorial variables. In Definition 2, a judgment of the form $\Gamma;\Phi \vdash F$ indicates that the type F is intended to be functorial in the variables in Φ but not necessarily in those in Γ.

Definition 2. *The formation rules for the set* $\mathcal{F} \subseteq \bigcup_{V\subseteq\mathbb{T},P\subseteq\mathbb{F}} \mathcal{F}^P(V)$ *of* well-formed types *are*

$$\frac{}{\Gamma;\Phi \vdash \mathbb{0}} \qquad \frac{}{\Gamma;\Phi \vdash \mathbb{1}}$$

$$\frac{\Gamma;\Phi \vdash F \qquad \Gamma;\Phi \vdash G}{\Gamma;\Phi \vdash F+G} \qquad \frac{\Gamma;\Phi \vdash F \qquad \Gamma;\Phi \vdash G}{\Gamma;\Phi \vdash F\times G}$$

$$\frac{\Gamma;\Phi \vdash F \qquad \Gamma;\Phi \vdash G}{\Gamma;\emptyset \vdash \mathsf{Nat}^{\Phi} F\, G}$$

$$\frac{\phi^k \in \Gamma\cup\Phi \qquad \overline{\Gamma;\Phi \vdash F}}{\Gamma;\Phi \vdash \phi^k\overline{F}}$$

$$\frac{\Gamma;\overline{\gamma^0},\overline{\alpha^0},\phi^k \vdash F \qquad \overline{\Gamma;\Phi,\overline{\gamma^0} \vdash G}}{\Gamma;\Phi,\overline{\gamma^0} \vdash (\mu\phi^k.\lambda\overline{\alpha^0}.\,F)\,\overline{G}}$$

$$\frac{\Gamma;\Phi,\overline{\alpha}^0 \vdash F \qquad \overline{\Gamma;\overline{\alpha}^0 \vdash K} \qquad \overline{\Gamma;\Phi \vdash A}}{\Gamma;\Phi \vdash (\mathsf{Lan}^{\overline{\alpha^0}}_{\overline{K}} F)\,\overline{A}}$$

In addition to textual replacement, we also have a proper substitution operation on types. If F is a type over P and V, if P and V contain only type variables of arity 0, and if $k = 0$ for every occurrence of ϕ^k bound by μ in F, then we say that F is *first-order*; otherwise we say that F is *second-order*. Substitution for first-order types is the usual capture-avoiding textual substitution. We write $F[\alpha := \sigma]$ for the result of substituting σ for α in F, and $F[\alpha_1 := F_1, ..., \alpha_k := F_k]$, or $F[\overline{\alpha := F}]$ when convenient, for $F[\alpha_1 := F_1][\alpha_2 := F_2, ..., \alpha_k := F_k]$. Substitution for second-order types is defined below, where we adopt a similar notational convention for vectors of types. Note that it is not correct to substitute along non-functorial variables.

Definition 3. *If* $\Gamma;\Phi,\phi^k \vdash H$ *and if* $\Gamma;\Phi,\overline{\alpha} \vdash F$ *with* $|\overline{\alpha}| = k$, *then* $\Gamma;\Phi \vdash H[\phi :=_{\overline{\alpha}} F]$. *Similarly, if* $\Gamma,\phi^k;\Phi \vdash H$, *and if* $\Gamma;\overline{\psi},\overline{\alpha} \vdash F$ *with* $|\overline{\alpha}| = k$ *and* $\Phi\cap\overline{\psi} = \emptyset$, *then* $\Gamma,\overline{\psi}';\Phi \vdash H[\phi :=_{\overline{\alpha}} F[\overline{\psi :== \psi'}]]$. *Here, the operation* $(\cdot)[\phi :=_{\overline{\alpha}} F]$ *of* second-order type substitution along $\overline{\alpha}$ *is defined by:*

$$\begin{array}{lcl}
\mathbb{0}[\phi :=_{\overline{\alpha}} F] & = & \mathbb{0}\\
\mathbb{1}[\phi :=_{\overline{\alpha}} F] & = & \mathbb{1}\\
(\mathsf{Nat}^{\overline{\beta}} G\,K)[\phi :=_{\overline{\alpha}} F] & = & \mathsf{Nat}^{\overline{\beta}}\,(G[\phi :=_{\overline{\alpha}} F])\,(K[\phi :=_{\overline{\alpha}} F])\\
(\psi\overline{G})[\phi :=_{\overline{\alpha}} F] & = & \begin{cases} \psi\,\overline{G[\phi :=_{\overline{\alpha}} F]} & \text{if } \psi \neq \phi\\ F[\overline{\alpha := G[\phi :=_{\overline{\alpha}} F]}] & \text{if } \psi = \phi \end{cases}\\
(G+K)[\phi :=_{\overline{\alpha}} F] & = & G[\phi :=_{\overline{\alpha}} F] + K[\phi :=_{\overline{\alpha}} F]\\
(G\times K)[\phi :=_{\overline{\alpha}} F] & = & G[\phi :=_{\overline{\alpha}} F] \times K[\phi :=_{\overline{\alpha}} F]\\
((\mu\psi.\lambda\overline{\beta}.\,G)\overline{K})[\phi :=_{\overline{\alpha}} F] & = & (\mu\psi.\lambda\overline{\beta}.\,G[\phi :=_{\overline{\alpha}} F])\,\overline{K[\phi :=_{\overline{\alpha}} F]}\\
((\mathsf{Lan}^{\overline{\beta}}_{\overline{H}} G)\overline{K})[\phi :=_{\overline{\alpha}} F] & = & (\mathsf{Lan}^{\overline{\beta}}_{\overline{H}} G[\phi :=_{\overline{\alpha}} F])\,\overline{K[\phi :=_{\overline{\alpha}} F]}
\end{array}$$

We note that $(\cdot)[\phi^0 :=_{\emptyset} F]$ coincides with first-order substitution. We also omit $\overline{\alpha}$ when convenient.

1.2 Terms

We now define our term calculus. To do so we assume an infinite set $\mathcal{V}$ of term variables disjoint from $\mathbb{T}$ and $\mathbb{F}$. If Γ is a type constructor context and Φ is a functorial context, then a *term context for* Γ *and* Φ is a finite set of bindings of the form $x : F$, where $x \in \mathcal{V}$ and $\Gamma;\Phi \vdash F$. We adopt the same conventions for denoting disjoint unions and for vectors in term contexts as for type constructor contexts and functorial contexts.

Definition 4. *Let Δ be a term context for Γ and Φ. The formation rules for the set of* well-formed terms over Δ *are*

$$\frac{\Gamma;\Phi \vdash F}{\Gamma;\Phi \,|\, \Delta, x : F \vdash x : F} \qquad \frac{\Gamma;\Phi \,|\, \Delta \vdash t : \mathbb{0} \qquad \Gamma;\Phi \vdash F}{\Gamma;\Phi \,|\, \Delta \vdash \bot_F t : F} \qquad \frac{}{\Gamma;\Phi \,|\, \Delta \vdash \top : \mathbb{1}}$$

$$\frac{\Gamma;\Phi \,|\, \Delta \vdash s : F}{\Gamma;\Phi \,|\, \Delta \vdash \mathsf{inl}\, s : F + G} \qquad \frac{\Gamma;\Phi \,|\, \Delta \vdash t : G}{\Gamma;\Phi \,|\, \Delta \vdash \mathsf{inr}\, t : F + G}$$

$$\frac{\Gamma;\Phi \vdash F, G \qquad \Gamma;\Phi \,|\, \Delta \vdash t : F + G \qquad \Gamma;\Phi \,|\, \Delta, x : F \vdash l : K \qquad \Gamma;\Phi \,|\, \Delta, y : G \vdash r : K}{\Gamma;\Phi \mid \Delta \vdash \mathsf{case}\, t\, \mathsf{of}\, \{x \mapsto l;\, y \mapsto r\} : K}$$

$$\frac{\Gamma;\Phi \,|\, \Delta \vdash s : F \qquad \Gamma;\Phi \,|\, \Delta \vdash t : G}{\Gamma;\Phi \,|\, \Delta \vdash (s, t) : F \times G} \qquad \frac{\Gamma;\Phi \,|\, \Delta \vdash t : F \times G}{\Gamma;\Phi \,|\, \Delta \vdash \pi_1 t : F} \qquad \frac{\Gamma;\Phi \,|\, \Delta \vdash t : F \times G}{\Gamma;\Phi \,|\, \Delta \vdash \pi_2 t : G}$$

$$\frac{\Gamma;\Phi \vdash F \qquad \Gamma;\Phi \vdash G \qquad \Gamma;\Phi \,|\, \Delta, x : F \vdash t : G}{\Gamma;\emptyset \,|\, \Delta \vdash L_\Phi x.t : \mathsf{Nat}^{\Phi}\, F\, G}$$

$$\frac{\overline{\Gamma;\Phi, \overline{\beta} \vdash K} \qquad \Gamma;\emptyset \,|\, \Delta \vdash t : \mathsf{Nat}^{\overline{\psi}}\, F\, G \qquad \Gamma;\Phi \,|\, \Delta \vdash s : F[\overline{\psi :=_{\overline{\beta}} K}]}{\Gamma;\Phi \,|\, \Delta \vdash t_{\overline{K}} s : G[\overline{\psi :=_{\overline{\beta}} K}]}$$

$$\frac{\Gamma;\Phi, \overline{\phi} \vdash H \qquad \overline{\Gamma;\Phi, \overline{\beta} \vdash F} \qquad \overline{\Gamma;\Phi, \overline{\beta} \vdash G}}{\Gamma;\emptyset \mid \emptyset \vdash \mathsf{map}_H^{\overline{F},\overline{G}} : \mathsf{Nat}^{\emptyset}\, (\overline{\mathsf{Nat}^{\Phi,\overline{\beta}}\, F\, G})\, (\mathsf{Nat}^{\Phi}\, H[\overline{\phi :=_{\overline{\beta}} F}]\; H[\overline{\phi :=_{\overline{\beta}} G}])}$$

$$\frac{\Gamma;\Phi, \phi, \overline{\alpha} \vdash H}{\Gamma;\emptyset \,|\, \emptyset \vdash \mathsf{in}_H : \mathsf{Nat}^{\Phi,\overline{\beta}} H[\phi :=_{\overline{\beta}} (\mu\phi.\lambda\overline{\alpha}.H)\overline{\beta}][\overline{\alpha := \beta}]\, (\mu\phi.\lambda\overline{\alpha}.H)\overline{\beta}}$$

$$\frac{\Gamma;\phi, \Phi, \overline{\alpha} \vdash H \qquad \Gamma;\Phi, \overline{\beta} \vdash F}{\Gamma;\emptyset \,|\, \emptyset \vdash \mathsf{fold}_H^F : \mathsf{Nat}^{\emptyset}\, (\mathsf{Nat}^{\Phi,\overline{\beta}}\, H[\phi :=_{\overline{\beta}} F][\overline{\alpha := \beta}]\, F)\, (\mathsf{Nat}^{\Phi,\overline{\beta}}\, (\mu\phi.\lambda\overline{\alpha}.H)\overline{\beta}\, F)}$$

$$\frac{\Gamma;\Phi, \overline{\alpha} \vdash F \qquad \overline{\Gamma;\overline{\alpha} \vdash K} \qquad \overline{\Gamma;\Phi \vdash A} \qquad \Gamma;\Phi \mid \Delta \vdash t : F[\overline{\alpha := A}]}{\Gamma;\Phi \mid \Delta \vdash \int_{\overline{K},F} t : (\mathsf{Lan}_{\overline{K}}^{\overline{\alpha}}\, F)\, \overline{K[\overline{\alpha := A}]}}$$

$$\frac{\Gamma;\emptyset \mid \Delta \vdash \eta : \mathsf{Nat}^{\Phi,\overline{\alpha}}\, F\, G[\overline{\beta := K}] \qquad \overline{\Gamma;\Phi \vdash A} \qquad \Gamma;\Phi \mid \Delta \vdash t : (\mathsf{Lan}_{\overline{K}}^{\overline{\alpha}} F)\, \overline{A}}{\Gamma;\Phi \mid \Delta \vdash \partial_F^{G,\overline{K}} \eta\, t : G[\overline{\beta := A}]}$$

In the rule for $L_{\overline{\alpha}}x.t$, the L operator binds all occurrences of the type variables in $\overline{\alpha}$ in the type of the term variable x and in the body t, as well as all occurrences of x in t. In the rule for $t_{\overline{K}}s$ there is one functorial expression in $\overline{K}$ for every functorial variable in $\overline{\alpha}$. In the rule for $\mathsf{map}_H^{\overline{F},\overline{G}}$ there is one functorial expression F and one functorial expression G for each functorial variable in $\overline{\phi}$. Moreover, for each ϕ^k in $\overline{\phi}$ the number of functorial variables in $\overline{\beta}$ in the judgments for its corresponding functorial expresssions F and G is k. In the rules for in_H and fold_H^F, the functorial variables in $\overline{\beta}$ are fresh with respect to H, and there is one β for every α. (Recall from above that, in order for the types of in_H and fold_H^F to be well-formed, the length of α must equal the arity of ϕ.) In the rule for $\int_{\overline{K},H} t$, there is one functorial expression A for every functorial variable in $\overline{\alpha}$, and in the rule for $\partial_F^{G,\overline{K}}\,\eta\,t$, there is one functorial expression A for every functorial expression in $\overline{K}$ (and hence for every functorial variable in $\overline{\beta}$).

Substitution for terms is the obvious extension of the usual capture-avoiding textual substitution, and Definition 4 ensures that the expected weakening rules for well-formed terms hold.

Sum and product intro and elim rules should be annotated with constituent types for consistency?

We should have a computation rule along the lines of: If $\eta : \mathsf{Nat}^{\overline{\alpha}} F\, G[\overline{\beta := K}]$ then

$$\begin{array}{ll} & (\partial_F^{G,\overline{K}}\eta)_{\overline{K[\overline{\alpha := A}]}} \circ (\textstyle\int_{K,F})_{\overline{A}} \;\rightarrow\; \eta_{\overline{A}} \\ : & F[\overline{\alpha := A}] \to G[\overline{\beta := K[\overline{\alpha := A}]}] \\ = & F[\overline{\alpha := A}] \to G[\overline{\beta := K}][\overline{\alpha := A}] \end{array}$$

This will appear as a computational property of the term interpretations.

2 INTERPRETING TYPES

The fundamental idea underlying Reynolds' parametricity is to give each type $F(\alpha)$ with one free variable α both an *object interpretation* F_0 taking sets to sets and a *relational interpretation* F_1 taking relations $R : \mathsf{Rel}(A, B)$ to relations $F_1(R) : \mathsf{Rel}(F_0(A), F_0(B))$, and to interpret each term $t(\alpha, x) : F(\alpha)$ with one free term variable $x : G(\alpha)$ as a map t_0 associating to each set A a homomorphism $t_0(A) : G_0(A) \to F_0(A)$, and to each relation R a morphism $t_1(R) : G_1(R) \to F_1(R)$. These interpretations are to be given inductively on the structures of F and t in such a way that they imply two fundamental theorems. The first is an *Identity Extension Lemma*, which states that $F_1(\mathsf{Eq}_A) = \mathsf{Eq}_{F_0(A)}$, and is the essential property that makes a model relationally parametric rather than just induced by a logical relation. The second is an *Abstraction Theorem*, which states that, for any $R : \mathsf{Rel}(A, B)$, $(t_0(A), t_0(B))$ is a morphism in Rel from $(G_0(A), G_0(B), G_1(R))$ to $(F_0(A), F_0(B), F_1(R))$. The Identity Extension Lemma is similar to the Abstraction Theorem except that it holds for *all* elements of a type's interpretation, not just those that are interpretations of terms. Similar theorems are expected to hold for types and terms with any number of free variables.

To accommodate GADTs, we will need to transition Reynolds' approach from a Set-based semantics to a semantics based on ω-complete partial orders. We denote the category of ω-complete partial orders (ωCPOs) and their sup-preserving morphisms by ωCPO. The underlying set of an ω-CPO A is denoted $|A|$. The category ωCPORel of relations on ωCPOs has as its objects triples (A, B, R), where $A, B : \omega\mathrm{CPO}$ and $R : \mathsf{Rel}(|A|, |B|)$, and has as its morphisms from (A, B, R) to (A', B', R') pairs $(f : A \to A', g : B \to B')$ of morphisms in ωCPO such that $(f\,a, g\,b) \in R'$

whenever $(a, b) \in R$. We write $R : \omega\mathsf{CPORel}(A, B)$ in place of $(A, B, R) : \omega\mathsf{CPORel}$ when convenient. If $R : \omega\mathsf{CPORel}(A, B)$ then we write $\pi_1 R$ and $\pi_2 R$ for the *domain* A of R and the *codomain* B of R, respectively. If $A : \omega\mathsf{CPO}$, then we write $\mathsf{Eq}_A = (A, A, \mathsf{Eq}_{|A|})$ for the *equality relation* on A.

To adapt Reynolds' approach, we first inductively define, for each type, an object interpretation in ωCPO and a relational interpretation in ωCPORel. Next, we show that these interpretations satisfy both an Identity Extension Lemma (Theorem **??**) and an Abstraction Theorem (Theorem **??**) appropriate to the ωCPO setting. The key to proving our Identity Extension Lemma is a familiar "cutting down" of the interpretations of universally quantified types to include only the "parametric" elements; as in [Johann et al. 2020], the relevant types of the calculus defined above are the (now richer) Nat-types. The requisite cutting down requires that the object interpretations of our types in ωCPO are defined simultaneously with their relational interpretations in ωCPORel. We give the object interpretations for our types in Section 2.1 and give their relational interpretations in Section 2.2. While the former are relatively straightforward, the latter are less so, mainly because of the cocontinuity conditions, adapted from the Set-based setting of [Johann et al. 2020], that must hold if they are to be well-defined. We develop these conditions in Section 2.2, which separates Definitions 8 and **??** in space, but otherwise has no impact on the fact that they are given by mutual induction.

2.1 Object Interpretations of Types

The object interpretations of the types in our calculus will be ω_1-cocontinuous functors between categories of ω_1-cocontinuous functors on categories constructed from the locally ω_1-presentable category ωCPO. We therefore begin by recording some important facts about locally ω_1-presentable categories and functors on them, and verifying verify the properties needed to interpret our syntax.

2.1.1 Preliminaries. Perhaps have as preliminaries to entire paper. Do everything for λ*CPOs*? Define these; investigate their properties.

A category is *small* if its collection of morphisms is a set. It is *locally small* if, for any two objects A and B, the collection of morphisms from A to B is a set. A *small (co)limit* in a category C is a (co)limit of a diagram $F : \mathcal{A} \to C$, where $\mathcal{A}$ is a small category. A category C is *(co)complete* if it has all small (co)limits.

A poset $\mathcal{D} = (D, \leq)$ is ω_1*-directed* if every countable subset of D has a supremum. When $\mathcal{D}$ is considered as a category, we write $d \in \mathcal{D}$ to indicate that d is an object of $\mathcal{D}$ (i.e., $d \in D$). An ω_1*-directed colimit* in a category C is a colimit of a diagram $F : \mathcal{D} \to C$, where $\mathcal{D}$ is an ω_1-directed poset. A category C is ω_1*-cocomplete* if it has all ω_1-directed colimits; a *cocomplete* category is one that has all colimits.

If $\mathcal{A}$ and C are ω_1-cocomplete, then the functor $F : \mathcal{A} \to C$ is ω_1*-cocontinuous* if it preserves ω_1-directed colimits. We write $[\mathcal{A}, C]_{\omega_1}$ for the category of ω_1-cocontinuous functors from $\mathcal{A}$ to C, and $C^{\mathcal{A}}$ for the category of *all* functors from $\mathcal{A}$ to C. Since (co)limits of functors are computed pointwise, $C^{\mathcal{A}}$ has all (co)limits that C has, and (co)limits of (co)continuous functors are again (co)continuous. It follows that $[\mathcal{A}, C]_{\omega_1}$ is $(\omega_1 1)$(co)complete whenever C is.

If $\mathcal{A}$ is locally small, then an object A of $\mathcal{A}$ is ω_1*-presentable* if the functor $\mathsf{Hom}_{\mathcal{A}}(A, -) : \mathcal{A} \to \mathsf{Set}$ preserves ω_1-directed colimits, i.e., if for every ω_1-directed poset $\mathcal{D}$ and every functor $F : \mathcal{D} \to C$, there is a canonical isomorphism $\varinjlim_{d \in \mathcal{D}} \mathsf{Hom}_{\mathcal{A}}(A, Fd) \simeq \mathsf{Hom}_{\mathcal{A}}(A, \varinjlim_{d \in \mathcal{D}} Fd)$. A locally small category $\mathcal{A}$ is ω_1*-accessible* if it is ω_1-cocomplete and has a set $\mathcal{A}_0$ of ω_1-presentable objects such that every object is an ω_1-directed colimit of objects in $\mathcal{A}_0$; a locally small category is *locally* ω_1*-presentable* if it is ω_1-accessible and cocomplete.

The category ωCPO is locally ω_1-presentable (but not locally finitely presentable); see Examples 1.18(2) of [Adámek and Rosický 1994]). Its ω_1-presentable objects are precisely the ωCPOs cardinality less than ω_1, i.e., the countable ωCPOs. In the next subsection we will interpret type variables in $\mathbb{T}^k \cup \mathbb{F}^k$ as elements of $[\omega\mathsf{CPO}^k, \omega\mathsf{CPO}]_{\omega_1}$; the following special cases of standard results (see, e.g., [Adámek and Rosický 1994]) will therefore be critical to deducing important properties of our object interpretations of types:

PROPOSITION 5. (1) *If $C_1, \ldots, C_n$ are locally ω_1-presentable categories then so is $C_1 \times \ldots \times C_n$. Moreover, the presentable objects of $C_1 \times \ldots \times C_n$ are exactly the tuples of the form $(P_1, \ldots, P_n)$, where, for each $i = 1, \ldots, n$, the object P_i is presentable in C_i.*
(2) *If $\mathcal{A}$ is ω_1-accessible and C is λ-cocomplete, then the category $[\mathcal{A}, C]_{\omega_1}$ is naturally equivalent to the category $C^{\mathcal{A}_0}$.*
(3) *If C is locally ω_1-presentable and $\mathcal{A}_0$ is essentially small, then $C^{\mathcal{A}_0}$ is locally ω_1-presentable.*

Together, the statements in Proposition 5 give that if $\mathcal{A}$ and C are locally ω_1-presentable, then $[\mathcal{A}, C]_{\omega_1}$ is naturally equivalent to $C^{\mathcal{A}_0}$, and thus is ω_1-presentable. Thus, for all $k_1, \ldots, k_n \in \mathbb{N}^n$, $[\mathcal{A}, C]^{k_1}_{\omega_1} \times \ldots \times [\mathcal{A}, C]^{k_n}_{\omega_1}$ is locally ω_1-presentable, and therefore $[[\mathcal{A}, C]^{k_1}_{\omega_1} \times \ldots \times [\mathcal{A}, C]^{k_n}_{\omega_1}, C]_{\omega_1}$ is as well. Taking both $\mathcal{A}$ and C to be ωCPO — as we will to ensure that the fixpoints interpreting μ-types in ωCPO exist — we have

PROPOSITION 6. *For all $k_1, \ldots, k_n \in \mathbb{N}^n$,*

$$[[\omega\mathsf{CPO}, \omega\mathsf{CPO}]^{k_1}_{\omega_1} \times \ldots \times [\omega\mathsf{CPO}, \omega\mathsf{CPO}]^{k_n}_{\omega_1}, \omega\mathsf{CPO}]_{\omega_1}$$

is locally ω_1-presentable.

2.1.2 *Object Interpretations.*

To define the object interpretations of the types in Definition 2 we must first interpret their variables. We have:

DEFINITION 7. *A* ωCPO environment *maps each type variable in $\mathbb{T}^k \cup \mathbb{F}^k$ to an element of $[\omega\mathsf{CPO}^k, \omega\mathsf{CPO}]_{\omega_1}$. A morphism $f : \rho \to \rho'$ for set environments ρ and ρ' with $\rho|_{\mathbb{T}} = \rho'|_{\mathbb{T}}$ maps each type constructor variable $\psi^k \in \mathbb{T}$ to the identity natural transformation on $\rho\psi^k = \rho'\psi^k$ and each functorial variable $\phi^k \in \mathbb{F}$ to a natural transformation from the k-ary functor $\rho\phi^k$ on ωCPO to the k-ary functor $\rho'\phi^k$ on ωCPO. Composition of morphisms on set environments is given componentwise, with the identity morphism mapping each set environment to itself. This gives a category of set environments and morphisms between them, which we denote $\omega\mathsf{CPOEnv}$.*

When convenient we identify a functor in $[\omega\mathsf{CPO}^0, \omega\mathsf{CPO}]_{\omega_1}$ with its value on $*$ and consider a ωCPO environment to map a type variable of arity 0 to an ω_1-cocontinuous functor from $\omega\mathsf{CPO}^0$ to ωCPO, i.e., to an ωCPO. If $\Phi = \{\phi_1^{k_1}, \ldots, \phi_n^{k_n}\}$ and $\overline{K} = \{K_1, \ldots, K_n\}$ where $K_i : [\omega\mathsf{CPO}^{k_i}, \omega\mathsf{CPO}]_{\omega_1}$ for $i = 1, \ldots, n$, then we write $\rho[\overline{\Phi := K}]$ for the ωCPO environment ρ' such that $\rho'\phi_i = K_i$ for $i = 1, \ldots, n$ and $\rho'\phi = \rho\phi$ if $\phi \notin \Phi$. If ρ is an ωCPO environment, we write Eq_ρ for the $\omega\mathsf{CPORel}$ environment (see Definition ??) such that $\mathsf{Eq}_\rho v = \mathsf{Eq}_{\rho v}$ for every type variable v. The categories $\omega\mathsf{CPORT}_k$ and relational interpretations appearing in the third clause of Definition 8 are given in full in Section ??.

Definition 8. *The* object interpretation $[\![\cdot]\!]^{\omega\mathsf{CPO}} : \mathcal{F} \to [\omega\mathsf{CPOEnv}, \omega\mathsf{CPO}]_{\omega_1}$ *is defined by*

$$
\begin{aligned}
[\![\Gamma;\Phi \vdash \mathbb{0}]\!]^{\omega\mathsf{CPO}}\rho &= 0\\
[\![\Gamma;\Phi \vdash \mathbb{1}]\!]^{\omega\mathsf{CPO}}\rho &= 1\\
[\![\Gamma;\emptyset \vdash \mathsf{Nat}^{\Phi}\,F\,G]\!]^{\omega\mathsf{CPO}}\rho &= \{\eta : \lambda\overline{K}.\,[\![\Gamma;\Phi \vdash F]\!]^{\omega\mathsf{CPO}}\rho[\overline{\Phi := K}] \Rightarrow \lambda\overline{K}.\,[\![\Gamma;\Phi \vdash G]\!]^{\omega\mathsf{CPO}}\rho[\overline{\Phi := K}]\\
&\qquad \mid \forall\overline{K = (K_0, K_1, K^*) : \omega\mathsf{CPORT}_k}.\\
&\qquad\quad (\eta_{\overline{K_0}}, \eta_{\overline{K_1}}) : [\![\Gamma;\Phi \vdash F]\!]^{\omega\mathsf{CPORel}}\mathsf{Eq}_\rho[\overline{\Phi := K}] \to [\![\Gamma;\Phi \vdash G]\!]^{\omega\mathsf{CPORel}}\mathsf{Eq}_\rho[\overline{\Phi :=}\\
[\![\Gamma;\Phi \vdash \phi\overline{F}]\!]^{\omega\mathsf{CPO}}\rho &= (\rho\phi)\,\overline{[\![\Gamma;\Phi \vdash F]\!]^{\omega\mathsf{CPO}}\rho}\\
[\![\Gamma;\Phi \vdash F + G]\!]^{\omega\mathsf{CPO}}\rho &= [\![\Gamma;\Phi \vdash F]\!]^{\omega\mathsf{CPO}}\rho + [\![\Gamma;\Phi \vdash G]\!]^{\omega\mathsf{CPO}}\rho\\
[\![\Gamma;\Phi \vdash F \times G]\!]^{\omega\mathsf{CPO}}\rho &= [\![\Gamma;\Phi \vdash F]\!]^{\omega\mathsf{CPO}}\rho \times [\![\Gamma;\Phi \vdash G]\!]^{\omega\mathsf{CPO}}\rho\\
[\![\Gamma;\Phi,\overline{\gamma} \vdash (\mu\phi.\lambda\overline{\alpha}.H)\overline{G}]\!]^{\omega\mathsf{CPO}}\rho &= (\mu T^{\omega\mathsf{CPO}}_{H,\rho})\overline{[\![\Gamma;\Phi,\overline{\gamma} \vdash G]\!]^{\omega\mathsf{CPO}}\rho}\\
\textit{where } T^{\omega\mathsf{CPO}}_{H,\rho}\, F &= \lambda\overline{A}.[\![\Gamma;\overline{\gamma},\phi,\overline{\alpha} \vdash H]\!]^{\omega\mathsf{CPO}}\rho[\phi := F][\overline{\alpha := A}]\\
\textit{and } T^{\omega\mathsf{CPO}}_{H,\rho}\, \eta &= \lambda\overline{A}.[\![\Gamma;\overline{\gamma},\phi,\overline{\alpha} \vdash H]\!]^{\omega\mathsf{CPO}} id_\rho[\phi := \eta][\overline{\alpha := id_A}]\\
[\![\Gamma;\Phi \vdash (\mathsf{Lan}^{\overline{\alpha}}_{\overline{K}}F)\overline{A}]\!]^{\omega\mathsf{CPO}}\rho &= (Lan_{\overline{[\![\Gamma;\overline{\alpha}\vdash K]\!]^{\omega\mathsf{CPO}}\rho[\overline{\alpha := \_}]}}[\![\Gamma;\Phi,\overline{\alpha} \vdash F]\!]^{\omega\mathsf{CPO}}\rho[\overline{\alpha := \_}])\overline{[\![\Gamma;\Phi \vdash A]\!]^{\omega\mathsf{CPO}}\rho}
\end{aligned}
$$

If $\rho \in \omega\mathsf{CPOEnv}$ and $\vdash F$ then we write $[\![\vdash F]\!]^{\omega\mathsf{CPO}}$ instead of $[\![\vdash F]\!]^{\omega\mathsf{CPO}}\rho$ since the environment is immaterial. The third clause of Definition 8 does indeed define an ωCPO. First, we have

Lemma 9. *The collection of all natural transformations*

$$\eta : \lambda\overline{K}.[\![\Gamma;\Phi \vdash F]\!]^{\mathsf{Set}}\rho[\overline{\Phi := K}] \Rightarrow \lambda\overline{K}.[\![\Gamma;\Phi \vdash G]\!]^{\mathsf{Set}}\rho[\overline{\Phi := K}]$$

defines a set.

Proof. We first note that $\lambda\overline{K}.[\![\Gamma;\Phi \vdash F]\!]^{\mathsf{Set}}\rho[\overline{\Phi := K}]$ and $\lambda\overline{K}.[\![\Gamma;\Phi \vdash G]\!]^{\mathsf{Set}}\rho[\overline{\Phi := K}]$ are both in $[[\omega\mathsf{CPO}^k, \omega\mathsf{CPO}]_{\omega_1}, \omega\mathsf{CPO}]_{\omega_1}$. By Proposition 6, $[[\omega\mathsf{CPO}^k, \omega\mathsf{CPO}]_{\omega_1}, \omega\mathsf{CPO}]_{\omega_1}$ is locally ω_1-presentable. It is therefore locally small, so there are only Set-many morphisms (i.e., natural transformations) between any two functors in $[[\omega\mathsf{CPO}^k, \omega\mathsf{CPO}]_{\omega_1}, \omega\mathsf{CPO}]_{\omega_1}$. In particular, there are only Set-many natural transformations from $\lambda\overline{K}.[\![\Gamma;\Phi \vdash F]\!]^{\mathsf{Set}}\rho[\overline{\Phi := K}]$ to $\lambda\overline{K}.[\![\Gamma;\Phi \vdash G]\!]^{\mathsf{Set}}\rho[\overline{\Phi := K}]$. □

Need to modify to take equality conditions into account. If C is any category, then given functors $F, G : C \to \omega\mathsf{CPO}$ and natural transformations $\eta, \eta' : F \to G$, we define $\eta \leq \eta'$ iff $\eta_c \leq \eta'_c$ for all $c \in C$, i.e., iff $\eta_c\,x \leq \eta'_c\,x$ in Gc for all $c \in C$ and $x \in Fc$. If $(\eta_i)_{i<\omega}$ is a chain in $\{\eta : F \to G\}$, then the family of morphisms $(\bigvee_{i<\omega}\eta_i)_c = \lambda x.\,\bigvee_{i<\omega}(\eta_i)_c\,x : Fc \to Gc$ defines a natural transformation $\bigvee_{i<\omega}\eta_i : F \to G$, and this natural transformation is clearly the supremum of $(\eta_i)_{i<\omega}$ in $\{\eta : F \to G\}$. We therefore have that $\{\eta : F \to G\}$ is itself an ωCPO. Letting $F = \lambda\overline{K}.[\![\Gamma;\Phi \vdash F]\!]^{\mathsf{Set}}\rho[\overline{\Phi := K}]$ and $G = \lambda\overline{K}.[\![\Gamma;\Phi \vdash G]\!]^{\mathsf{Set}}\rho[\overline{\Phi := K}]$, we have that $[\![\Gamma;\emptyset \vdash \mathsf{Nat}^{\Phi}F\,G]\!]^{\omega\mathsf{CPO}}$ is indeed an ωCPO, i.e., is a constantly-valued functor into ωCPO.

Interpretations of Nat-types ensure that $[\![\Gamma \vdash F \to G]\!]^{\omega\mathsf{CPO}}$ and $[\![\Gamma \vdash \forall\overline{\alpha}.F]\!]^{\omega\mathsf{CPO}}$ are as expected in any parametric model.

To make sense of the last clause in Definition 8, we need to know that, for each $\rho \in \omega\mathsf{CPOEnv}$, $T^{\omega\mathsf{CPO}}_{H,\rho}$ is an ω_1-cocontinuous endofunctor on $[\omega\mathsf{CPO}^k, \omega\mathsf{CPO}]_{\omega_1}$, and thus admits a fixpoint. Since $T^{\omega\mathsf{CPO}}_{H,\rho}$ is defined in terms of $[\![\Gamma;\overline{\gamma},\phi,\overline{\alpha} \vdash H]\!]^{\omega\mathsf{CPO}}$, this means that interpretations of types must be such functors, which in turn means that the actions of set interpretations of types on objects and on morphisms in ωCPOEnv are intertwined. Fortunately, we know from [Johann and Polonsky 2019] that, for every $\Gamma;\Phi \vdash G$, $[\![\Gamma;\Phi \vdash G]\!]^{\omega\mathsf{CPO}}$ is actually in $[[\omega\mathsf{CPO}^k, \omega\mathsf{CPO}]_{\omega_1}, \omega\mathsf{CPO}]_{\omega_1}$, or, equivalently, $[\omega\mathsf{CPOEnv}, \omega\mathsf{CPO}]_{\omega_1}$. Therefore, for each $[\![\Gamma;\overline{\gamma}, \phi^k, \overline{\alpha} \vdash H]\!]^{\omega\mathsf{CPO}}$, the corresponding operator $T_H^{\omega\mathsf{CPO}}$ can be extended to a *functor* from ωCPOEnv to $[[\omega\mathsf{CPO}^k, \omega\mathsf{CPO}]_{\omega_1}, [\omega\mathsf{CPO}^k, \omega\mathsf{CPO}]_{\omega_1}]_{\omega_1}$. The action of $T_H^{\omega\mathsf{CPO}}$ on an object $\rho \in \omega\mathsf{CPOEnv}$ is given by the higher-order functor $T_{H,\rho}^{\omega\mathsf{CPO}}$, whose actions on objects (functors in $[\omega\mathsf{CPO}^k, \omega\mathsf{CPO}]_{\omega_1}$) and on morphisms (natural transformations between such functors) are given in Definition 8. The action of $T_H^{\omega\mathsf{CPO}}$ on a morphism $f : \rho \to \rho'$ is the higher-order natural transformation $T_{H,f}^{\omega\mathsf{CPO}} : T_{H,\rho}^{\omega\mathsf{CPO}} \to T_{H,\rho'}^{\omega\mathsf{CPO}}$ whose action on $F : [\omega\mathsf{CPO}^k, \omega\mathsf{CPO}]_{\omega_1}$ is the natural transformation $T_{H,f}^{\omega\mathsf{CPO}}\, F : T_{H,\rho}^{\omega\mathsf{CPO}}\, F \to T_{H,\rho'}^{\omega\mathsf{CPO}}\, F$ whose component at $\overline{A}$ is $(T_{H,f}^{\omega\mathsf{CPO}}\, F)_{\overline{A}} = [\![\Gamma;\overline{\gamma}, \phi, \overline{\alpha} \vdash H]\!]^{\omega\mathsf{CPO}} f[\phi := id_F][\overline{\alpha := id_A}]$.

In addition, for each $\overline{K}$, we have that $Lan_{\overline{K}}$ is itself a (higher-order) functor. Specifically, given functors $F, F' : \mathcal{C} \to \mathcal{D}$, a sequence of functors $\overline{K} = K_1, \ldots, K_n$ with $K_i : \mathcal{C} \to \mathcal{C}_i$ for $i = 1, \ldots, n$, and a natural transformation $\alpha : F \to G$, the functorial action $Lan_{\overline{K}}\alpha : Lan_{\overline{K}}F \to Lan_{\overline{K}}F'$ of $Lan_{\overline{K}}$ on α is defined to be the unique natural transformation such that $((Lan_{\overline{K}}\alpha) \circ \langle K_1, \ldots, K_n\rangle) \circ \eta_F = \eta_{F'} \circ \alpha$. Here, $\eta_F : F \to (Lan_{\overline{K}}F) \circ \langle K_1, \ldots, K_n\rangle$ and $\eta_{F'} : F' \to (Lan_{\overline{K}}F') \circ \langle K_1, \ldots, K_n\rangle$ are the natural transformations associated with the functors $Lan_{\overline{K}}F$ and $Lan_{\overline{K}}F'$ from $\Pi_{i\in\{1,\ldots,n\}}\mathcal{C}_i$ to $\mathcal{D}$, respectively. It is not hard to see that $Lan_{\overline{K}}$ is a (higher-order) functor under this definition.

The next definition uses the functors $T_H^{\omega\mathsf{CPO}}$ and Lan_K to define the actions of functors interpreting types on morphisms between set environments.

Definition 10. *Let $f : \rho \to \rho'$ be a morphism between ω-CPO environments ρ and ρ' (so that $\rho|_{\mathbb{T}} = \rho'|_{\mathbb{T}}$). The action $[\![\Gamma;\Phi \vdash F]\!]^{\omega\mathsf{CPO}} f$ of $[\![\Gamma;\Phi \vdash F]\!]^{\omega\mathsf{CPO}}$ on the morphism f is given as follows:*

- *If $\Gamma;\Phi \vdash \mathbb{0}$ then $[\![\Gamma;\Phi \vdash \mathbb{0}]\!]^{\omega\mathsf{CPO}} f = id_0$*
- *If $\Gamma;\Phi \vdash \mathbb{1}$ then $[\![\Gamma;\Phi \vdash \mathbb{1}]\!]^{\omega\mathsf{CPO}} f = id_1$*
- *If $\Gamma;\emptyset \vdash \mathsf{Nat}^\Phi F\, G$ then $[\![\Gamma;\emptyset \vdash \mathsf{Nat}^\Phi F\, G]\!]^{\omega\mathsf{CPO}} f = id_{[\![\Gamma;\emptyset\vdash\mathsf{Nat}^\Phi F\, G]\!]^{\omega\mathsf{CPO}}\rho}$*
- *If $\Gamma;\Phi \vdash \phi\overline{F}$ then*

$$[\![\Gamma;\Phi \vdash \phi\overline{F}]\!]^{\omega\mathsf{CPO}} f : [\![\Gamma;\Phi \vdash \phi\overline{F}]\!]^{\omega\mathsf{CPO}}\rho \to [\![\Gamma;\Phi \vdash \phi\overline{F}]\!]^{\omega\mathsf{CPO}}\rho' = (\rho\phi)\overline{[\![\Gamma;\Phi \vdash F]\!]^{\omega\mathsf{CPO}}\rho} \to (\rho'\phi)\overline{[\![\Gamma;\Phi \vdash F]\!]^{\omega\mathsf{CPO}}\rho'}$$

is defined by

$$\begin{aligned}[\![\Gamma;\Phi \vdash \phi\overline{F}]\!]^{\omega\mathsf{CPO}} f &= (f\phi)_{\overline{[\![\Gamma;\Phi\vdash F]\!]^{\omega\mathsf{CPO}}\rho'}} \circ (\rho\phi)\overline{[\![\Gamma;\Phi \vdash F]\!]^{\omega\mathsf{CPO}} f} \\ &= (\rho'\phi)\overline{[\![\Gamma;\Phi \vdash F]\!]^{\omega\mathsf{CPO}} f} \circ (f\phi)_{\overline{[\![\Gamma;\Phi\vdash F]\!]^{\omega\mathsf{CPO}}\rho}}\end{aligned}$$

The latter equality holds because $\rho\phi$ and $\rho'\phi$ are functors and $f\phi : \rho\phi \to \rho'\phi$ is a natural transformation, so the following naturality square commutes:

$$\begin{tikzcd}
(\rho\phi)\overline{[\![\Gamma;\Phi \vdash F]\!]^{\omega\mathsf{CPO}}\rho} \arrow[r, "(f\phi)_{\overline{[\![\Gamma;\Phi\vdash F]\!]^{\omega\mathsf{CPO}}\rho}}"] \arrow[d, "(\rho\phi)\overline{[\![\Gamma;\Phi\vdash F]\!]^{\omega\mathsf{CPO}} f}"'] & (\rho'\phi)\overline{[\![\Gamma;\Phi \vdash F]\!]^{\omega\mathsf{CPO}}\rho} \arrow[d, "(\rho'\phi)\overline{[\![\Gamma;\Phi\vdash F]\!]^{\omega\mathsf{CPO}} f}"'] \\
(\rho\phi)\overline{[\![\Gamma;\Phi \vdash F]\!]^{\omega\mathsf{CPO}}\rho'} \arrow[r, "(f\phi)_{\overline{[\![\Gamma;\Phi\vdash F]\!]^{\omega\mathsf{CPO}}\rho'}}"] & (\rho'\phi)\overline{[\![\Gamma;\Phi \vdash F]\!]^{\omega\mathsf{CPO}}\rho'}
\end{tikzcd} \tag{1}$$

- *If $\Gamma;\Phi \vdash F + G$ then $[\![\Gamma;\Phi \vdash F + G]\!]^{\omega\mathsf{CPO}} f$ is defined by*

$$[\![\Gamma;\Phi \vdash F + G]\!]^{\omega\mathsf{CPO}} f(\mathsf{inl}\, x) = \mathsf{inl}\,([\![\Gamma;\Phi \vdash F]\!]^{\omega\mathsf{CPO}} f x)$$

$$[\![\Gamma;\Phi \vdash F + G]\!]^{\omega\mathsf{CPO}} f(\mathsf{inr}\, y) = \mathsf{inr}\,([\![\Gamma;\Phi \vdash G]\!]^{\omega\mathsf{CPO}} f y)$$

- *If* $\Gamma;\Phi \vdash F \times G$ *then* $[\![\Gamma;\Phi \vdash F \times G]\!]^{\omega\mathrm{CPO}} f = [\![\Gamma;\Phi \vdash F]\!]^{\omega\mathrm{CPO}} f \times [\![\Gamma;\Phi \vdash G]\!]^{\omega\mathrm{CPO}} f$
- *If* $\Gamma;\Phi,\overline{\gamma} \vdash (\mu\phi.\lambda\overline{\alpha}.H)\overline{G}$ *then*

$$\begin{array}{rcl}[\![\Gamma;\Phi,\overline{\gamma} \vdash (\mu\phi.\lambda\overline{\alpha}.H)\overline{G}]\!]^{\omega\mathrm{CPO}} f & : & [\![\Gamma;\Phi,\overline{\gamma} \vdash (\mu\phi.\lambda\overline{\alpha}.H)\overline{G}]\!]^{\omega\mathrm{CPO}}\rho \to [\![\Gamma;\Phi,\overline{\gamma} \vdash (\mu\phi.\lambda\overline{\alpha}.H)\overline{G}]\!]^{\omega\mathrm{CPO}}\rho' \\ & = & (\mu T^{\omega\mathrm{CPO}}_{H,\rho})\overline{[\![\Gamma;\Phi,\overline{\gamma} \vdash G]\!]^{\omega\mathrm{CPO}}\rho} \to (\mu T^{\omega\mathrm{CPO}}_{H,\rho'})\overline{[\![\Gamma;\Phi,\overline{\gamma} \vdash G]\!]^{\omega\mathrm{CPO}}\rho'}\end{array}$$

is defined by

$$\begin{array}{rl} & (\mu T^{\omega\mathrm{CPO}}_{H,f})_{\overline{[\![\Gamma;\Phi,\overline{\gamma}\vdash G]\!]^{\omega\mathrm{CPO}}\rho'}} \circ (\mu T^{\omega\mathrm{CPO}}_{H,\rho})\overline{[\![\Gamma;\Phi,\overline{\gamma} \vdash G]\!]^{\omega\mathrm{CPO}} f} \\ = & (\mu T^{\omega\mathrm{CPO}}_{H,\rho'})\overline{[\![\Gamma;\Phi,\overline{\gamma} \vdash G]\!]^{\omega\mathrm{CPO}} f} \circ (\mu T^{\omega\mathrm{CPO}}_{H,f})_{\overline{[\![\Gamma;\Phi,\overline{\gamma}\vdash G]\!]^{\omega\mathrm{CPO}}\rho}}\end{array}$$

The latter equality holds because $\mu T^{\omega\mathrm{CPO}}_{H,\rho}$ *and* $\mu T^{\omega\mathrm{CPO}}_{H,\rho'}$ *are functors and* $\mu T^{\omega\mathrm{CPO}}_{H,f} : \mu T^{\omega\mathrm{CPO}}_{H,\rho} \to \mu T^{\omega\mathrm{CPO}}_{H,\rho'}$ *is a natural transformation, so the following naturality square commutes:*

$$\begin{tikzcd}[column sep=huge] (\mu T^{\omega\mathrm{CPO}}_{H,\rho})\overline{[\![\Gamma;\Phi,\overline{\gamma} \vdash G]\!]^{\omega\mathrm{CPO}}\rho} \arrow[r, "(\mu T^{\omega\mathrm{CPO}}_{H,f})_{\overline{[\![\Gamma;\Phi,\overline{\gamma}\vdash G]\!]^{\omega\mathrm{CPO}}\rho}}"] \arrow[d, "(\mu T^{\omega\mathrm{CPO}}_{H,\rho})\overline{[\![\Gamma;\Phi,\overline{\gamma}\vdash G]\!]^{\omega\mathrm{CPO}} f}"'] & (\mu T^{\omega\mathrm{CPO}}_{H,\rho'})\overline{[\![\Gamma;\Phi,\overline{\gamma} \vdash G]\!]^{\omega\mathrm{CPO}}\rho} \arrow[d, "(\mu T^{\omega\mathrm{CPO}}_{H,\rho'})\overline{[\![\Gamma;\Phi,\overline{\gamma}\vdash G]\!]^{\omega\mathrm{CPO}} f}"'] \\ (\mu T^{\omega\mathrm{CPO}}_{H,\rho})\overline{[\![\Gamma;\Phi,\overline{\gamma} \vdash G]\!]^{\omega\mathrm{CPO}}\rho'} \arrow[r, "(\mu T^{\omega\mathrm{CPO}}_{H,f})_{\overline{[\![\Gamma;\Phi,\overline{\gamma}\vdash G]\!]^{\omega\mathrm{CPO}}\rho'}}"] & (\mu T^{\omega\mathrm{CPO}}_{H,\rho'})\overline{[\![\Gamma;\Phi,\overline{\gamma} \vdash G]\!]^{\omega\mathrm{CPO}}\rho'} \end{tikzcd} \tag{2}$$

- *If* $\Gamma;\Phi \vdash (\mathrm{Lan}^{\overline{\alpha}}_{\overline{K}} F)\overline{A}$ *then*

$$[\![\Gamma;\Phi \vdash (\mathrm{Lan}^{\overline{\alpha}}_{\overline{K}} F)\overline{A}]\!]^{\omega\mathrm{CPO}} f : [\![\Gamma;\Phi \vdash (\mathrm{Lan}^{\overline{\alpha}}_{\overline{K}} F)\overline{A}]\!]^{\omega\mathrm{CPO}}\rho \to [\![\Gamma;\Phi \vdash (\mathrm{Lan}^{\overline{\alpha}}_{\overline{K}} F)\overline{A}]\!]^{\omega\mathrm{CPO}}\rho'$$

is defined by

$$\begin{array}{rl} & \big(Lan_{\overline{[\![\Gamma;\overline{\alpha}\vdash K]\!]^{\omega\mathrm{CPO}}\rho[\overline{\alpha:=\_}]}}[\![\Gamma;\Phi,\overline{\alpha} \vdash F]\!]^{\omega\mathrm{CPO}} f[\overline{\alpha := id\_}]\big)\overline{[\![\Gamma;\Phi \vdash A]\!]^{\omega\mathrm{CPO}}\rho'} \\ & \circ\ \big(Lan_{\overline{[\![\Gamma;\overline{\alpha}\vdash K]\!]^{\omega\mathrm{CPO}}\rho[\overline{\alpha:=\_}]}}[\![\Gamma;\Phi,\overline{\alpha} \vdash F]\!]^{\omega\mathrm{CPO}}\rho[\overline{\alpha := \_}]\big)\overline{[\![\Gamma;\Phi \vdash A]\!]^{\omega\mathrm{CPO}} f} \\ = & \big(Lan_{\overline{[\![\Gamma;\overline{\alpha}\vdash K]\!]^{\omega\mathrm{CPO}}\rho[\overline{\alpha:=\_}]}}[\![\Gamma;\Phi,\overline{\alpha} \vdash F]\!]^{\omega\mathrm{CPO}}\rho'[\overline{\alpha := \_}]\big)\overline{[\![\Gamma;\Phi \vdash A]\!]^{\omega\mathrm{CPO}} f} \\ & \circ\ \big(Lan_{\overline{[\![\Gamma;\overline{\alpha}\vdash K]\!]^{\omega\mathrm{CPO}}\rho[\overline{\alpha:=\_}]}}[\![\Gamma;\Phi,\overline{\alpha} \vdash F]\!]^{\omega\mathrm{CPO}} f[\overline{\alpha := id\_}]\big)\overline{[\![\Gamma;\Phi \vdash A]\!]^{\omega\mathrm{CPO}}\rho}\end{array}$$

where the above equality holds by naturality of $Lan_{\overline{[\![\Gamma;\overline{\alpha}\vdash K]\!]^{\omega\mathrm{CPO}}\rho[\overline{\alpha:=\_}]}}[\![\Gamma;\Phi,\overline{\alpha} \vdash F]\!]^{\omega\mathrm{CPO}} f[\overline{\alpha := id\_}]$.

Definitions 8 and 10 respect weakening, i.e., ensure that a type and its weakenings have the same set interpretations.

2.2 Relational Interpretations of Types

Some questions/issues:

- Can we write zipBush and appendBush with ∂ and $\int$? We could already represent the uncurried type of appendBush (although not its curried type), but couldn't recurse over both input bushes because folds take natural transformations as inputs.
- More generally, how do we compute with ∂ and $\int$? Can we use the colimit formulation of Lans (see Lemma 6.3.7 of [Riehl 2016]) to get a handle on this?

- What is the connection between exponentials and natural transformations? (Should we assume only small objects are exponentiable?) Do we want the former or the latter for computational purposes? (I suspect the latter.)
  [From nlab: In a functor category D^C , a natural transformation $\alpha : F \to G$ is exponentiable if (though probably not "only if") it is cartesian and each component $\alpha_c : Fc \to Gc$ is exponentiable in D. Given $H \to F$ we define $(\Pi_\alpha H)c = \Pi_{\alpha_c}(Hc)$; then for $u : c \to c'$ to obtain a map $\Pi_{\alpha_c}(Hc) \to \Pi_{\alpha_{c'}}(Hc')$ we need a map $\alpha^*_{c'}(\Pi_{\alpha_c}(Hc)) \to Hc'$. But since α is cartesian, $\alpha^*_{c'}(\Pi_{\alpha_c}(Hc)) \cong \alpha^*_c(\Pi_{\alpha_c}(Hc))$, so we have the counit $\alpha^*_c(\Pi_{\alpha_c}(Hc)) \to Hc$ that we can compose with Hu.]
- After we understand what we can do with Lans and folds on GADTs we might want to try to extend calculus with term-level fixpoints. This would give a categorical analogue for GADTs of [Pitts 1998, 2000] for ADTs. Would it also more accurately reflect how GADTs are used in practice, or are functions over GADTs usually folds? Investigate applications in the literature and/or in implementations.
- ωCPO is a natural choice for modeling general recursion. We know $(Lan^{\gamma}_C \mathbb{1})D$ is $C \to D$ for any closed type C. (Also for select classes of open types?) So can model $\mathrm{Nat} \to \gamma$. But the functor $NX = \mathrm{Nat} \to X$ isn't ω-cocontinuous. It also doesn't preserve ω_1-presentable objects, i.e., countable ωCPOs since $\mathrm{Nat} \to \mathrm{Nat}$ is not countable. So we cannot have a functor like N as the subscript to Lan and expect the resulting Lan to be ω_1-cocontinuous.
- What functors can be subscripts to Lan and produce ω_1-cocontinuous functors? We can use functors that preserve presentable objects by theorem in [Johann and Polonsky 2019], and possibly others as well. These include polynomial functors, ADTs and nested types seen as functors, certain (which?) GADTs seen as functors? How big can GADTs get?

REFERENCES

J. Adámek and J. Rosický. 1994. *Locally Presentable and Accessible Categories*. Cambridge University Press.

R. Atkey. 2012. Relational Parametricity for Higher Kinds. In *Computer Science Logic*. 46–61.

E. S. Bainbridge, P. J. Freyd, A. Scedrov, and P. J. Scott. 1990. Functorial Polymorphism. *Theoretical Computer Science* 70 (1990), 35–64.

N. Ghani, P. Johann, F. Nordvall Forsberg, F. Orsanigo, and T. Revell. 2015. Bifibrational Functorial Semantics for Parametric Polymorphism. In *Mathematical Foundations of Program Semantics*. 165–181.

P. Johann, E. Ghiorzi, and D. Jeffries. 2020. Free Theorems for Nested Types. (2020).

P. Johann and A. Polonsky. 2019. Higher-kinded Data Types: Syntax and Semantics. In *Logic in Computer Science*. 1–13. https://doi.org/10.1109/LICS.2019.8785657

A. Pitts. 1998. Parametric polymorphism, recursive types, and operational equivalence. (1998).

A. Pitts. 2000. Parametric polymorphism and operational equivalence. *Mathematical Structures in Computer Science* 10 (2000), 1–39.

J. C. Reynolds. 1983. Types, abstraction, and parametric polymorphism. *Information Processing* 83(1) (1983), 513–523.

E. Riehl. 2016. *Category Theory in Context*. Dover.