
PARAMETRICITY FOR PRIMITIVE NESTED TYPES AND GADTS

PATRICIA JOHANN, ENRICO GHIORZI, AND DANIEL JEFFRIES

Appalachian State University
e-mail address: johannp@appstate.edu

Appalachian State University
e-mail address: ghiorzie@appstate.edu

Appalachian State University
e-mail address: jeffriesd@appstate.edu

ABSTRACT. This paper considers parametricity and its consequent free theorems for nested data types. Rather than representing nested types via their Church encodings in a higher-kinded or dependently typed extension of System F, we adopt a functional programming perspective and design a Hindley-Milner-style calculus with primitives for constructing nested types directly as fixpoints. Our calculus can express all nested types appearing in the literature, including truly nested types. At the level of terms, it supports primitive pattern matching, map functions, and fold combinators for nested types. Our main contribution is the construction of a parametric model for our calculus. This is both delicate and challenging. In particular, to ensure the existence of semantic fixpoints interpreting nested types, and thus to establish a suitable Identity Extension Lemma for our calculus, our type system must explicitly track functoriality of types, and cocontinuity conditions on the functors interpreting them must be appropriately threaded throughout the model construction. We also prove that our model satisfies an appropriate Abstraction Theorem, as well as that it verifies all standard consequences of parametricity in the presence of primitive nested types. We give several concrete examples illustrating how our model can be used to derive useful free theorems, including a short cut fusion transformation, for programs over nested types. Finally, we consider generalizing our results to GADTs, and argue that no extension of our parametric model for nested types can give a functorial interpretation of GADTs in terms of left Kan extensions and still be parametric.

1. INTRODUCTION

Algebraic data types (ADTs), both built-in and user-defined, have long been at the core of functional languages such as Haskell, ML, Agda, Epigram, and Idris. ADTs, such as that of natural numbers, can be unindexed. But they can also be indexed over other types. For example, the ADT of lists (here coded in Agda)

Key words and phrases: Nested types, GADTs, categorical semantics, parametricity.
Supported by NSF awards CCR-1906388 and 1420175.

```

data List (A : Set) : Set where
  nil : List A
  cons : A → List A → List A

```

is indexed over its element type A . The instance of `List` at index A depends only on itself, and so is independent of `List B` for any other index B . That is, `List`, like all other ADTs, defines a *family of inductive types*, one for each index type.

Over time, there has been a notable trend toward data types whose non-regular indexing can capture invariants and other sophisticated properties that can be used for program verification and other applications. A simple example of such a type is given by the type

```

data PTree (A : Set) : Set where
  pleaf : A → PTree A
  pnode : PTree (A × A) → PTree A

```

of perfect trees, which can be thought of as constraining lists to have lengths that are powers of 2. A similar data type `Nest` is given in the canonical paper [BM98] on nested types. The above code makes clear that perfect trees at index type A are defined in terms of perfect trees at index type $A \times A$. This is typical of nested types, one type instance of which can depend on others, so that the entire family of types must actually be defined at once. A nested type thus defines not a family of inductive types, but rather an *inductive family of types*. Nested types include simple nested types, like perfect trees, none of whose recursive occurrences occur below another type constructor; “deep” nested types [JP20], such as the nested type

```

data PForest (A : Set) : Set where
  fempty : PForest A
  fnode : A → PTree (PForest A) → PForest A

```

of perfect forests, whose recursive occurrences appear below type constructors for other nested types; and truly nested types, such as the nested type

```

data Bush (A : Set) : Set where
  bn timer : Bush A
  bcons : A → Bush (Bush A) → Bush A

```

of bushes, whose recursive occurrences appear below their own type constructors.

Relational parametricity encodes a powerful notion of type-uniformity, or representation independence, for data types in polymorphic languages. It formalizes the intuition that a polymorphic program must act uniformly on all of its possible type instantiations by requiring that every such program preserves all relations between pairs of types at which it is instantiated. Parametricity was originally put forth by Reynolds [Rey83] for System F [Gir72], the calculus at the core of all polymorphic functional languages. It was later popularized as Wadler’s “theorems for free” [Wad89], so called because it can deduce properties of programs in such languages solely from their types, i.e., with no knowledge whatsoever of the text of the programs involved. Most of Wadler’s free theorems are consequences of naturality for polymorphic list-processing functions. However, parametricity can also derive results that go beyond just naturality, such as correctness for ADTs of the program optimization known as *short cut fusion* [GLP93, Joh02].

But what about nested types? Does parametricity still hold if such types are added to polymorphic calculi? More practically, can we justifiably reason type-independently about (functions over) nested types in functional languages?

Type-independent reasoning about ADTs in functional languages is usually justified by first representing ADTs by their Church encodings, and then reasoning type-independently about these

encodings. This is typically justified by constructing a parametric model — i.e., a model in which polymorphic functions preserve relations *à la* Reynolds — for a suitable fragment of System F, demonstrating that an initial algebra exists for the positive type constructor corresponding to the functor underlying an ADT of interest, and showing that each such initial algebra is suitably isomorphic to its corresponding Church encoding. In fact, this isomorphism of initial algebras and their Church encodings is one of the “litmus tests” for the goodness of a parametric model.

This approach works well for ADTs, which are always fixpoints of *first-order* functors, and whose Church encodings, which involve quantification over only type variables, are always expressible in System F. For example, `List A` is the fixpoint of the first-order functor $F X = 1 + A \times X$ and has Church encoding $\forall \alpha. \alpha \rightarrow (A \rightarrow \alpha \rightarrow \alpha) \rightarrow \alpha$. But despite Cardelli’s [Car84] claim that “virtually any basic type of interest can be encoded within F_2 ” — i.e., within System F — non-ADT nested types cannot. Not even our prototypical nested type of perfect trees has a Church encoding expressible in System F! Indeed, `PTree A` cannot be represented as the fixpoint of any *first-order* functor. However, it can be seen as the instance at index `A` of the fixpoint of the *higher-order* functor $H F A = (A \rightarrow F A) \rightarrow (F (A \times A) \rightarrow F A) \rightarrow F A$. It thus has Church encoding $\forall f. (\forall \alpha. \alpha \rightarrow f \alpha) \rightarrow (\forall \alpha. f(\alpha \times \alpha) \rightarrow f \alpha) \rightarrow \forall \alpha. f \alpha$, which requires quantification at the higher kind $* \rightarrow *$ for f . A similar situation obtains for any (non-ADT) nested type. Unfortunately, higher-kinded quantification is not available in System F, so if we want to reason type-independently about nested types in a language based on it we have only two options: *i*) move to an extension of System F, such as the higher-kinded calculus F_ω or a dependent type theory, and reason via their Church encodings in a known parametric model for that extension, or *ii*) add nested types to System F as primitives — i.e., as primitive type-level fixpoints — and construct a parametric model for the result.

Since the type systems of F_ω and dependent type theories are designed to extend System F with far more than non-ADT data types, it seems like serious overkill to pass to their parametric models to reason about nested types in System F. Indeed, such calculi support fundamentally new features — e.g., the full hierarchy of higher-kinds or term-indexed types — adding complexity to their models that is entirely unnecessary for reasoning about nested types. In addition, to our knowledge no such models represent data types as primitive fixpoints. (See the past paragraph of this section for a more detailed discussion in the case of F_ω .) Whether or not the results of this paper even hold for such calculi is an open question.

This paper therefore pursues the second option above. We first design a Hindley-Milner-style calculus supporting primitive nested types, together with primitive types of natural transformations representing morphisms between them. Our calculus can express all nested types appearing in the literature, including truly nested types. At the term-level, it supports primitive pattern matching, map functions, and fold combinators for nested types.¹ Our main contribution is the construction of a parametric model for our calculus. This is both delicate and challenging. To ensure the existence of semantic fixpoints interpreting nested types, and thus to establish a suitable Identity Extension Lemma, our type system must explicitly track functoriality of types, and cocontinuity conditions on the functors interpreting them must be appropriately threaded throughout the model construction. Our model validates all standard consequences of parametricity in the presence of primitive nested types, including the isomorphism of primitive ADTs and their Church encodings, and correctness of short cut fusion for nested types. The relationship between naturality and parametricity has long been of interest, and our inclusion of a primitive type of natural transformations allows us to clearly

¹We leave incorporating general term-level recursion to future work because, as Pitts [Pit00] reminds us, “it is hard to construct models of both impredicative polymorphism and fixpoint recursion”. In fact, as the development in this paper shows, constructing a parametric model even for our predicative calculus with primitive nested types — and even without term-level fixpoints — is already rather involved. On the other hand, our calculus is strongly normalizing, so it perhaps edges us toward the kind of provably total practical programming language proposed in [Wad89].

delineate those consequences of parametricity that follow from naturality, from those, such as short cut fusion for nested types, that require the full power of parametricity.

Structure of this Paper We introduce our calculus in Section 2. Its type system is based on the level-2-truncation of the higher-kinded grammar from [JP19], augmented with a primitive type of natural transformations. (Since [JP19] contains no term calculus, the issue of parametricity could not even be raised there.) In Section 3 we give set and relational interpretations of our types. Set interpretations are possible precisely because our calculus is predicative — as ensured by our primitive natural transformation types — and [JP19] guarantees that local finite presentability of **Set** makes it suitable for interpreting nested types. As is standard in categorical models, types are interpreted as functors from environments interpreting their type variable contexts to sets or relations, as appropriate. To ensure that these functors satisfy the cocontinuity properties needed for the semantic fixpoints interpreting nested types to exist, set environments must map k -ary type constructor variables to appropriately cocontinuous k -ary functors on sets, relation environments must map k -ary type constructor variables to appropriately cocontinuous k -ary relation transformers, and these cocontinuity conditions must be threaded through our type interpretations in such a way that an Identity Extension Lemma (Theorem 4.4) can be proved. Properly propagating the cocontinuity conditions requires considerable care, and Section 4, where it is done, is (apart from tracking functoriality in the calculus so that it is actually possible) where the bulk of the work in constructing our model lies.

In Section 5, we give set and relational interpretations for the terms of our calculus. As usual in categorical models, terms are interpreted as natural transformations from interpretations of their term contexts to interpretations of their types, and these must cohere in what is essentially a fibred way. In Section 6.1 we prove a scheme deriving free theorems that are consequences of naturality of polymorphic functions over nested types. This scheme is very general, and is parameterized over both the data type and the type of the polymorphic function at hand. It has, for example, analogues for nested types of Wadler’s map-rearrangement free theorems as instances. In Section 6 we prove that our model satisfies an Abstraction Theorem (Theorem 6.4), which we use to derive, in Section 7, other parametricity results that go beyond mere naturality. In Section 8 we show that the results of this paper do not extend to arbitrary GADTs, and argue that no extension of the parametric model we construct here for nested types can give a functorial interpretation of GADTs in terms of left Kan extensions as proposed in [JP19] and still be parametric. We conclude and offer some directions for future work in Section 9.

This paper extends the conference paper [JGJ21] by including more exposition, proofs of all theorems, and more examples. In addition, the discussion of parametricity for GADTs in Section 8 is entirely new.

Related Work There is a long line of work on categorical models of parametricity for System F; see, e.g., [BFSS90, BM05, DR04, GJF⁺15, Has94, Jac99, MR92, RR94]. To our knowledge, all such models treat ADTs via their Church encodings, verifying in the just-constructed parametric model that each Church encoding is a solution for the fixpoint equation associated with its corresponding data type. The present paper draws on this rich tradition of categorical models of parametricity for System F, but modifies them to treat nested types (and therefore ADTs) as primitive data types.

The only other extensions we know of System F with primitive data types are those in [Mat11, MG01, Pit98, Pit00, Wad89]. Wadler [Wad89] treats full System F, and sketches parametricity for its extension with lists. Martin and Gibbons [MG01] outline a semantics for a grammar of primitive nested types similar to that in [JP19], but treat only polynomial nested types, i.e., nested types that are fixpoints of polynomial higher-order functors. Unfortunately, the model suggested in [MG01] is not entirely correct (see [JP19]), and parametricity is nowhere mentioned. Matthes [Mat11] treats System F with non-polynomial ADTs and nested types, but his focus is on expressivity of generalized Mendler iteration for them. He gives no semantics whatsoever.

In [Pit00], Pitts adds list ADTs to full System F with a term-level fixpoint primitive. Other ADTs are included in [Pit98], but nested types are not expressible in either syntax. Pitts constructs parametric models for his calculi based on operational, rather than categorical, semantics. A benefit of using operational semantics to build parametric models is that it avoids needing to work in a suitable metatheory to accommodate System F’s impredicativity. It is well-known that there are no set-based parametric models of System F [Rey84], so parametric models for it and its extensions are often constructed in a syntactic metatheory such as the impredicative Calculus of Inductive Constructions (iCIC). By adding primitive nested types to a Hindley-Milner-style calculus and working in a categorical setting we side-step such metatheoretic distractions.

Atkey [Atk12] treats parametricity for arbitrary higher kinds, constructing a parametric model for System F_ω within iCIC, rather than in a semantic category. His construction is in some ways similar to ours, but he represents (now higher-kinded) data types using Church encodings rather than as primitives. Moreover, the *fmap* functions associated to Atkey’s functors must be *given*, presumably by the programmer, together with their underlying type constructors. This absolves him of imposing cocontinuity conditions on his model to ensure that fixpoints of his functors exist, but, unfortunately, he does not indicate which type constructors support *fmap* functions. We suspect explicitly spelling out which types can be interpreted as strictly positive functors would result in a full higher-kinded extension of a calculus akin to that presented here.

2. THE CALCULUS

2.1. 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. Letters from the beginning of the alphabet denote type variables of arity 0, i.e., elements of $\mathbb{T}^0 \cup \mathbb{F}^0$. We write $\bar{\phi}$ for either a set $\{\phi_1, \dots, \phi_n\}$ of type constructor variables or a set of functorial variables when the cardinality n of the set is unimportant or clear from context. If V is a set of type variables we write $V, \bar{\phi}$ for $V \cup \bar{\phi}$ when $V \cap \bar{\phi} = \emptyset$. We omit the vector notation for a singleton set, thus writing ϕ , instead of $\bar{\phi}$, for $\{\phi\}$.

If Γ is a finite subset of \mathbb{T} , Φ is a finite subset of \mathbb{F} , $\bar{\alpha}$ is a finite subset of \mathbb{F}^0 disjoint from Φ , and $\phi^k \in \mathbb{F}^k \setminus \Phi$, then the set \mathcal{F} of well-formed types is given in Definition 2.1. The notation there entails that an application $FF_1 \dots F_k$ is allowed only when F is a type variable of arity k , or F is a subexpression of the form $\mu \phi^k. \lambda \alpha_1 \dots \alpha_k. F'$. Accordingly, an overbar indicates a sequence of types whose length matches the arity of the type applied to it. Requiring that types are always in such *η -long normal form* avoids having to consider β -conversion at the level of types. In a type $\text{Nat}^{\bar{\alpha}} F G$, the Nat operator binds all occurrences of the variables in $\bar{\alpha}$ in F and G ; intuitively, $\text{Nat}^{\bar{\alpha}} F G$ represents the type of a natural transformation in $\bar{\alpha}$ from the functor F to the functor G . In a subexpression $\mu \phi^k. \lambda \bar{\alpha}. F$, the μ operator binds all occurrences of the variable ϕ , and the λ operator binds all occurrences of the variables in $\bar{\alpha}$, in the body F .

A *type constructor context*, or *non-functorial context*, is a finite set Γ of type constructor variables, and a *functorial context* is a finite set Φ of functorial variables. In Definition 2.1, 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 2.1. The formation rules for the set \mathcal{F} of (*well-formed*) *types* are

$$\frac{}{\Gamma; \Phi \vdash 0} \quad \frac{}{\Gamma; \Phi \vdash 1}$$

$$\begin{array}{c}
\frac{\Gamma; \bar{\alpha}^0 \vdash F \quad \Gamma; \bar{\alpha}^0 \vdash G}{\Gamma; \emptyset \vdash \text{Nat}^{\bar{\alpha}^0} F G} \\
\\
\frac{\phi^k \in \Gamma \cup \Phi \quad \bar{\Gamma}; \bar{\Phi} \vdash \bar{F}}{\Gamma; \Phi \vdash \phi^k \bar{F}} \\
\\
\frac{\Gamma; \bar{\alpha}^0, \phi^k \vdash F \quad \bar{\Gamma}; \bar{\Phi} \vdash \bar{G}}{\Gamma; \Phi \vdash (\mu \phi^k. \lambda \bar{\alpha}^0. F) \bar{G}} \\
\\
\frac{\Gamma; \Phi \vdash F \quad \Gamma; \Phi \vdash G}{\Gamma; \Phi \vdash F + G} \\
\\
\frac{\Gamma; \Phi \vdash F \quad \Gamma; \Phi \vdash G}{\Gamma; \Phi \vdash F \times G}
\end{array}$$

We write $\vdash F$ for $\emptyset; \emptyset \vdash F$. Definition 2.1 ensures that the expected weakening rules for well-formed types hold, i.e., if $\Gamma; \Phi \vdash F$ is well-formed, then both $\Gamma; \bar{\psi}; \Phi \vdash F$ and $\Gamma; \Phi; \bar{\phi} \vdash F$ are also well-formed. Note, however, that weakening does not change the contexts in which types can be formed. For example, the functorial context in which a type $\text{Nat}^0 F G$ is formed must be empty, although it can be weakened by a functorial variable β *after* the type is formed; by contrast, we cannot *first* weaken the functorial contexts of F and G to $\Gamma; \beta \vdash F$ and $\Gamma; \beta \vdash G$, and *then* form $\text{Nat}^0 F G$ in the weakened context $\Gamma; \beta$. If $\Gamma; \emptyset \vdash F$ and $\Gamma; \emptyset \vdash G$, then our rules allow formation of the type $\Gamma; \emptyset \vdash \text{Nat}^0 F G$, which represents the System F type $\Gamma \vdash F \rightarrow G$. Similarly, if $\Gamma; \bar{\alpha} \vdash F$, then our rules allow formation of the type $\Gamma; \emptyset \vdash \text{Nat}^{\bar{\alpha}} \mathbb{1} F$, which represents the System F type $\Gamma; \emptyset \vdash \forall \bar{\alpha}. F$. However, some System F types, such as $\forall \alpha. (\alpha \rightarrow \alpha) \rightarrow \alpha$, are not representable in our calculus. Note that, in the rule for μ -types, no variables in Φ appear in the body F of $\mu \phi. \lambda \bar{\alpha}. F$. This will be critical to proving the Identity Extension Lemma for our calculus.

Definition 2.1 allows the formation of all of the types from Section 1:

$$\begin{array}{ll}
\text{List } \alpha &= \mu \beta. \mathbb{1} + \alpha \times \beta \text{ or } (\mu \phi. \lambda \beta. \mathbb{1} + \beta \times \phi \beta) \alpha \\
\text{PTree } \alpha &= (\mu \phi. \lambda \beta. \beta + \phi(\beta \times \beta)) \alpha \\
\text{Forest } \alpha &= (\mu \phi. \lambda \beta. \mathbb{1} + \beta \times \text{PTree}(\phi \beta)) \alpha \\
\text{Bush } \alpha &= (\mu \phi. \lambda \beta. \mathbb{1} + \beta \times \phi(\phi \beta)) \alpha
\end{array}$$

Note that since the body F of a type $(\mu \phi. \lambda \bar{\alpha}. F) \bar{G}$ can only be functorial in ϕ and the variables in $\bar{\alpha}$, the representation of $\text{List } \alpha$ as the ADT $\mu \beta. \mathbb{1} + \alpha \times \beta$ cannot be functorial in α . By contrast, if $\text{List } \alpha$ is represented as the nested type $(\mu \phi. \lambda \beta. \mathbb{1} + \beta \times \phi \beta) \alpha$ then we can choose α to be a functorial variable or not when forming the type. This observation holds for other ADTs as well; for example, if $\text{Tree } \alpha \gamma = \mu \beta. \alpha + \beta \times \gamma \times \beta$, then $\alpha, \gamma; \emptyset \vdash \text{Tree } \alpha \gamma$ is well-formed, but $\emptyset; \alpha, \gamma \vdash \text{Tree } \alpha \gamma$ is not. And it also applies to some non-ADT types, such as $\text{GRose } \phi \alpha = \mu \beta. \mathbb{1} + \alpha \times \phi \beta$, in which ϕ and α must both be non-functorial variables. It is in fact possible to allow “extra” 0-ary functorial variables in the body of μ -types (functorial variables of higher arity are the real problem), which would allow the first-order representations of ADTs to be functorial. However, since doing this requires some changes to the formation rule for μ -types, as well as the delicate threading of some additional conditions throughout our model construction, we do not pursue this line of investigation here.

Definition 2.1 allows well-formed types to be functorial in no variables. Functorial variables can also be demoted to non-functorial status: if $F[\phi := \psi]$ is the textual replacement of ϕ in F , then $\Gamma, \psi^k; \Phi \vdash F[\phi^k := \psi^k]$ is derivable whenever $\Gamma; \Phi, \phi^k \vdash F$ is. The proof is by induction on the structure of F . In addition to textual replacement, we also have a substitution operation on types.

If $\Gamma; \Phi \vdash F$ is a type, if Γ and Φ 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 $H[\alpha := F]$ for the result of substituting F for α in H , and $H[\alpha_1 := F_1, \dots, \alpha_k := F_k]$, or $H[\bar{\alpha} := \bar{F}]$ when convenient, for $H[\alpha_1 := F_1][\alpha_2 := F_2, \dots, \alpha_k := F_k]$. The operation of *second-order type substitution along $\bar{\alpha}$* is given in Definition 2.2, where we adopt a similar notational convention for vectors of types. Of course, $(\cdot)[\phi^0 :=_{\emptyset} F]$ coincides with first-order substitution. We omit $\bar{\alpha}$ when convenient, but note that it is not correct to substitute along non-functorial variables.

Definition 2.2. If $\Gamma; \Phi, \phi^k \vdash H$ and $\Gamma; \Phi, \bar{\alpha} \vdash F$ with $|\bar{\alpha}| = k$, then the operation $H[\phi :=_{\bar{\alpha}} F]$ of *second-order type substitution along $\bar{\alpha}$* is defined by induction on H as follows:

$$\begin{aligned}
\mathbb{0}[\phi :=_{\bar{\alpha}} F] &= \mathbb{0} \\
\mathbb{1}[\phi :=_{\bar{\alpha}} F] &= \mathbb{1} \\
(\text{Nat}^{\bar{\beta}} G K)[\phi :=_{\bar{\alpha}} F] &= \text{Nat}^{\bar{\beta}} G K \\
(\psi \bar{G})[\phi :=_{\bar{\alpha}} F] &= \begin{cases} \psi \bar{G}[\phi :=_{\bar{\alpha}} F] & \text{if } \psi \neq \phi \\ F[\alpha := G[\phi :=_{\bar{\alpha}} F]] & \text{if } \psi = \phi \end{cases} \\
(G + K)[\phi :=_{\bar{\alpha}} F] &= G[\phi :=_{\bar{\alpha}} F] + K[\phi :=_{\bar{\alpha}} F] \\
(G \times K)[\phi :=_{\bar{\alpha}} F] &= G[\phi :=_{\bar{\alpha}} F] \times K[\phi :=_{\bar{\alpha}} F] \\
((\mu\psi.\lambda\bar{\beta}.G)\bar{K})[\phi :=_{\bar{\alpha}} F] &= (\mu\psi.\lambda\bar{\beta}.G) \bar{K}[\phi :=_{\bar{\alpha}} F]
\end{aligned}$$

The idea is that the arguments to ϕ get substituted for the variables in $\bar{\alpha}$ in each F replacing an occurrence of ϕ . It is not hard to see that $\Gamma; \Phi \vdash H[\phi :=_{\bar{\alpha}} F]$. **REMOVE? In addition, if $\Gamma, \phi^k; \Phi \vdash H$, $\Gamma; \bar{\psi}, \bar{\alpha} \vdash F$, $|\bar{\alpha}| = k$, and $\Phi \cap \bar{\psi} = \emptyset$, then $\Gamma, \bar{\psi}'; \Phi \vdash H[\phi :=_{\bar{\alpha}} F[\psi :=_{\bar{\psi}'} \psi']]$. Ask Daniel.**

2.2. Terms. To define our term calculus 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 above conventions for denoting disjoint unions and vectors in term contexts. If Δ is a term context for Γ and Φ , then the formation rules for the set of *well-formed terms over Δ* are as in Figure 1. In the rule there for $L_{\bar{\alpha}} x.t$, the L operator binds all occurrences of the type variables in $\bar{\alpha}$ in the types of x and t , as well as all occurrences of x in $\bar{\alpha}$. In the rule for $t_{\bar{K}} s$ there is one type in \bar{K} for every functorial variable in $\bar{\alpha}$. In the rule for $\text{map}_H^{\bar{F}, \bar{G}}$ there is one type F and one type G for each functorial variable in $\bar{\phi}$. Moreover, for each ϕ^k in $\bar{\phi}$ the number of functorial variables in $\bar{\beta}$ in the judgments for its corresponding type F and G is k . In the rules for in_H and fold_H^F , the functorial variables in $\bar{\beta}$ are fresh with respect to H , and there is one β for every α . Substitution for terms is the obvious extension of the usual capture-avoiding textual substitution, and the rules of Figure 1 ensure that weakening is respected.

The “extra” functorial variables $\bar{\gamma}$ in the rules for $\text{map}_H^{\bar{F}, \bar{G}}$ (i.e., those variables not affected by the substitution of ϕ) deserve comment. They allow us to map polymorphic functions over nested types. Suppose, for example, we want to map the polymorphic function $\text{flatten} : \text{Nat}^{\beta}(\text{PTree } \beta) (\text{List } \beta)$ over a list. Even in the absence of extra variables the instance of map required to map each non-functorial monomorphic instantiation of flatten over a list of perfect trees is well-formed:

$$\frac{\Gamma; \alpha \vdash \text{List } \alpha \quad \Gamma; \emptyset \vdash \text{PTree } F \quad \Gamma; \emptyset \vdash \text{List } G}{\Gamma; \emptyset \mid \emptyset \vdash \text{map}_{\text{List } \alpha}^{\text{PTree } F, \text{List } G} : \text{Nat}^{\emptyset}(\text{Nat}^{\emptyset}(\text{PTree } F) (\text{List } G)) (\text{Nat}^{\emptyset}(\text{List } (\text{PTree } F)) (\text{List } (\text{List } G)))}$$

But in the absence of $\bar{\gamma}$, the instance

$$\Gamma; \emptyset \mid \emptyset \vdash \text{map}_{\text{List } \alpha}^{\text{PTree } \beta, \text{List } \beta} : \text{Nat}^{\emptyset}(\text{Nat}^{\beta}(\text{PTree } \beta) (\text{List } \beta)) (\text{Nat}^{\beta}(\text{List } (\text{PTree } \beta)) (\text{List } (\text{List } \beta)))$$

$$\begin{array}{c}
\frac{\Gamma; \Phi \vdash F}{\Gamma; \Phi \mid \Delta, x : F \vdash x : F} \quad \frac{\Gamma; \Phi \mid \Delta \vdash t : \mathbb{0} \quad \Gamma; \Phi \vdash F}{\Gamma; \Phi \mid \Delta \vdash \perp_F t : F} \quad \frac{}{\Gamma; \Phi \mid \Delta \vdash \top : \mathbb{1}} \\
\frac{\Gamma; \Phi \mid \Delta \vdash s : F}{\Gamma; \Phi \mid \Delta \vdash \text{inl } s : F + G} \quad \frac{\Gamma; \Phi \mid \Delta \vdash t : G}{\Gamma; \Phi \mid \Delta \vdash \text{inr } t : F + G} \\
\frac{\Gamma; \Phi \vdash F, G \quad \Gamma; \Phi \mid \Delta \vdash t : F + G \quad \Gamma; \Phi \mid \Delta, x : F \vdash l : K \quad \Gamma; \Phi \mid \Delta, y : G \vdash r : K}{\Gamma; \Phi \mid \Delta \vdash \text{case } t \text{ of } \{x \mapsto l; y \mapsto r\} : K} \\
\frac{\Gamma; \Phi \mid \Delta \vdash s : F \quad \Gamma; \Phi \mid \Delta \vdash t : G}{\Gamma; \Phi \mid \Delta \vdash (s, t) : F \times G} \quad \frac{\Gamma; \Phi \mid \Delta \vdash t : F \times G}{\Gamma; \Phi \mid \Delta \vdash \pi_1 t : F} \quad \frac{\Gamma; \Phi \mid \Delta \vdash t : F \times G}{\Gamma; \Phi \mid \Delta \vdash \pi_2 t : G} \\
\frac{\Gamma; \bar{\alpha} \vdash F \quad \Gamma; \bar{\alpha} \vdash G \quad \Gamma; \bar{\alpha} \mid \Delta, x : F \vdash t : G}{\Gamma; \emptyset \mid \Delta \vdash L_{\bar{\alpha}} x. t : \text{Nat}^{\bar{\alpha}} F G} \\
\frac{\Gamma; \bar{\Phi} \vdash \bar{K} \quad \Gamma; \emptyset \mid \Delta \vdash t : \text{Nat}^{\bar{\alpha}} F G \quad \Gamma; \Phi \mid \Delta \vdash s : F[\bar{\alpha} := \bar{K}]}{\Gamma; \Phi \mid \Delta \vdash t_{\bar{K}} s : G[\bar{\alpha} := \bar{K}]} \\
\frac{\Gamma; \bar{\phi}, \bar{\gamma} \vdash H \quad \Gamma; \bar{\beta}, \bar{\gamma} \vdash F \quad \Gamma; \bar{\beta}, \bar{\gamma} \vdash G}{\Gamma; \emptyset \mid \emptyset \vdash \text{map}_{H^{\bar{F}, \bar{G}}}^{\bar{F}, \bar{G}} : \text{Nat}^{\emptyset} (\text{Nat}^{\bar{\beta}, \bar{\gamma}} F G) (\text{Nat}^{\bar{\gamma}} H[\bar{\phi} :=_{\bar{\beta}} \bar{F}] H[\bar{\phi} :=_{\bar{\beta}} \bar{G}])} \\
\frac{\Gamma; \phi, \bar{\alpha} \vdash H}{\Gamma; \emptyset \mid \emptyset \vdash \text{in}_H : \text{Nat}^{\bar{\beta}} H[\phi :=_{\bar{\beta}} (\mu \phi. \lambda \bar{\alpha}. H) \bar{\beta}][\bar{\alpha} := \bar{\beta}] (\mu \phi. \lambda \bar{\alpha}. H) \bar{\beta}} \\
\frac{\Gamma; \phi, \bar{\alpha} \vdash H \quad \Gamma; \bar{\beta} \vdash F}{\Gamma; \emptyset \mid \emptyset \vdash \text{fold}_H^F : \text{Nat}^{\emptyset} (\text{Nat}^{\bar{\beta}} H[\phi :=_{\bar{\beta}} F][\bar{\alpha} := \bar{\beta}] F) (\text{Nat}^{\bar{\beta}} (\mu \phi. \lambda \bar{\alpha}. H) \bar{\beta} F)}
\end{array}$$

Figure 1: Well-formed terms

required to map the *polymorphic flatten* function over a list of perfect trees is not: indeed, the functorial contexts for F and G in the rule for $\text{map}_{H^{\bar{F}, \bar{G}}}^{\bar{F}, \bar{G}}$ would have to be empty, but because the polymorphic *flatten* function is natural in β it cannot possibly have a type of the form $\text{Nat}^{\emptyset} F G$ as would be required for it to be the function input to **map**. Untypeability of this instance of **map** is unsatisfactory in a polymorphic calculus, where we naturally expect to be able to manipulate entire polymorphic functions rather than just their monomorphic instances, but the “extra” variables $\bar{\gamma}$ remedy the situation, ensuring that the instance of **map** needed to map the polymorphic *flatten* function is typeable as follows:

$$\frac{\Gamma; \alpha, \gamma \vdash \text{List } \alpha \quad \Gamma; \gamma \vdash \text{PTree } \gamma \quad \Gamma; \gamma \vdash \text{List } \gamma}{\Gamma; \emptyset \mid \emptyset \vdash \text{map}_{\text{List } \alpha}^{\text{PTree } \gamma, \text{List } \gamma} : \text{Nat}^{\emptyset} (\text{Nat}^{\gamma} (\text{PTree } \gamma) (\text{List } \gamma)) (\text{Nat}^{\gamma} (\text{List } (\text{PTree } \gamma)) (\text{List } (\text{List } \gamma)))}$$

Our calculus is expressive enough to define a function $\text{reversePTree} : \text{Nat}^{\alpha} (\text{PTree } \alpha) (\text{PTree } \alpha)$ that reverses the order of the leaves in a perfect tree. This function maps, e.g., the perfect tree

$$\text{pnode} (\text{pnode} (\text{pleaf} ((1, 2), (3, 4))))$$

to the perfect tree

$$\text{pnode} (\text{pnode} (\text{pleaf} ((4, 3), (2, 1))))$$

i.e., maps $((1, 2), (3, 4))$ to $((4, 3), (2, 1))$. It can be defined as

$$\vdash (\text{fold}_{\beta + \phi(\beta \times \beta)}^{\text{PTree } \alpha})_{\emptyset} s : \text{Nat}^{\alpha} (\text{PTree } \alpha) (\text{PTree } \alpha)$$

where

$$\begin{aligned} \text{fold}_{\beta+\phi(\beta \times \beta)}^{PTree \alpha} &: \text{Nat}^0(\text{Nat}^\alpha(\alpha + PTree(\alpha \times \alpha)) (PTree \alpha)) (\text{Nat}^\alpha(PTree \alpha) (PTree \alpha)) \\ \text{in}_{\beta+\phi(\beta \times \beta)} &: \text{Nat}^\alpha(\alpha + PTree(\alpha \times \alpha)) (PTree \alpha) \\ \text{map}_{PTree \alpha}^{\alpha \times \alpha, \alpha \times \alpha} &: \text{Nat}^0(\text{Nat}^\alpha(\alpha \times \alpha) (\alpha \times \alpha)) (\text{Nat}^\alpha(PTree(\alpha \times \alpha)) (PTree(\alpha \times \alpha))) \end{aligned}$$

and *pleaf*, *pnode*, *swap*, and *s* are the terms

$$\begin{aligned} \vdash \text{in}_{\beta+\phi(\beta \times \beta)} \circ (L_\alpha x. \text{inl } x) &: \text{Nat}^\alpha \alpha (PTree \alpha) \\ \vdash \text{in}_{\beta+\phi(\beta \times \beta)} \circ (L_\alpha x. \text{inr } x) &: \text{Nat}^\alpha PTree(\alpha \times \alpha) (PTree \alpha) \\ \vdash L_\alpha p. (\pi_2 p, \pi_1 p) &: \text{Nat}^\alpha(\alpha \times \alpha) (\alpha \times \alpha) \\ \vdash L_\alpha t. \text{case } t \text{ of } \{b \mapsto \text{pleaf } b; t' \mapsto \text{pnode } (((\text{map}_{PTree \alpha}^{\alpha \times \alpha, \alpha \times \alpha})_\emptyset \text{ swap})_\alpha t')\} &: \text{Nat}^\alpha(\alpha + PTree(\alpha \times \alpha)) PTree \alpha \end{aligned}$$

respectively. Our calculus can similarly define a **reverseBush** function that reverses the data in a bush. This function maps, e.g., the bush **Change one 2 to 4? Find a better name for this operation than reverse. Ask Daniel.**

$$\begin{aligned} &bcons \ 0 \ (bcons \ (bcons \ 1 \ (bcons \ (bcons \ 2 \ bnil) \ bnil)) \\ &\quad (bcons \ (bcons \ (bcons \ 2 \ (bcons \ (bcons \ 3 \ bnil) \ bnil)) \ bnil) \ bnil)) \end{aligned}$$

to the bush

$$\begin{aligned} &bcons \ 3 \ (bcons \ (bcons \ 0 \ (bcons \ (bcons \ 2 \ bnil) \ bnil)) \\ &\quad (bcons \ (bcons \ (bcons \ 2 \ (bcons \ (bcons \ 1 \ bnil) \ bnil)) \ bnil) \ bnil)) \end{aligned}$$

It can be defined as

$$\vdash (\text{fold}_{\mathbb{1}+\beta \times \phi(\phi\beta)}^{Bush \alpha})_\emptyset \text{ balg} : \text{Nat}^\alpha(Bush \alpha) (Bush \alpha)$$

where

$$\vdash \text{fold}_{\mathbb{1}+\beta \times \phi(\phi\beta)}^{Bush \alpha} : \text{Nat}^0(\text{Nat}^\alpha(\mathbb{1} + \alpha \times Bush(Bush \alpha)) (Bush \alpha)) (\text{Nat}^\alpha(Bush \alpha) (Bush \alpha))$$

and *bnil*, *bcons*, $\text{in}_{\mathbb{1}+\beta \times \phi(\phi\beta)}^{-1}$, *balg*, and *consalg* are the terms

$$\begin{aligned} \vdash \text{in}_{\mathbb{1}+\beta \times \phi(\phi\beta)} \circ (L_\alpha x. \text{inl } x) &: \text{Nat}^\alpha \mathbb{1} (Bush \alpha) \\ \vdash \text{in}_{\mathbb{1}+\beta \times \phi(\phi\beta)} \circ (L_\alpha x. \text{inr } x) &: \text{Nat}^\alpha(\alpha \times Bush(Bush \alpha)) (Bush \alpha) \\ \vdash (\text{fold}_{\mathbb{1}+\beta \times \phi(\phi\beta)}^{(\mathbb{1}+\beta \times \phi(\phi\beta))[\phi:=Bush \alpha]})_\emptyset ((\text{map}_{\mathbb{1}+\beta \times \phi(\phi\beta)}^{(\mathbb{1}+\beta \times \phi(\phi\beta))[\phi:=Bush \alpha][\beta:=\alpha], Bush \alpha})_\emptyset \text{in}_{\mathbb{1}+\beta \times \phi(\phi\beta)}) & \\ : \text{Nat}^\alpha(Bush \alpha) (\mathbb{1} + \alpha \times Bush(Bush \alpha)) & \\ \vdash L_\alpha s. \text{case } s \text{ of } \{ * \mapsto \text{bnil}_\alpha *; (a, bba) \mapsto \text{consalg}_\alpha(a, bba) \} &: \text{Nat}^\alpha(\mathbb{1} + \alpha \times Bush(Bush \alpha)) (Bush \alpha) \\ \vdash L_\alpha (a, bba). \text{case } ((\text{in}_{\mathbb{1}+\beta \times \phi(\phi\beta)}^{-1})_{Bush \alpha} bba) \text{ of } \{ & \\ * \mapsto \text{bcons}_\alpha(a, \text{bnil}_{Bush \alpha} *) & \\ (ba, bbba) \mapsto \text{case } ((\text{in}_{\mathbb{1}+\beta \times \phi(\phi\beta)}^{-1})_\alpha ba) \text{ of } \{ & \\ * \mapsto \text{bcons}_\alpha(a, \text{bcons}_{Bush \alpha}(\text{bnil}_\alpha *, bbba)); & \\ (a', bba') \mapsto \text{bcons}_\alpha(a', \text{bcons}_{Bush \alpha}(\text{bcons}_\alpha(a, bba'), bbba)) \} & \} \\ : \text{Nat}^\alpha(\alpha \times Bush(Bush \alpha)) (Bush \alpha) & \end{aligned}$$

respectively.

Unfortunately, our calculus cannot express types of recursive functions — such as a concatenation function for perfect trees or a zip function for bushes — that take as inputs a nested type and an argument of another type, both of which are parameterized over the same variable. The fundamental issue is that recursion is expressible only via *fold*, which produces natural transformations in some variables $\bar{\alpha}$ from μ -types to other functors F . The restrictions on **Nat**-types entail that F cannot itself be a **Nat**-type containing $\bar{\alpha}$, so, e.g., $\text{Nat}^\alpha(PTree \alpha) (\text{Nat}^0(PTree \alpha) (PTree(\alpha \times \alpha)))$ is not well-formed. Uncurrying gives $\text{Nat}^\alpha(PTree \alpha \times PTree \alpha) (PTree(\alpha \times \alpha))$, which is well-formed, but *fold* cannot produce a term of this type because $PTree \alpha \times PTree \alpha$ is not a μ -type. Our calculus can, however, express types of recursive functions that take multiple nested types as arguments, provided they are parameterized over disjoint sets of type variables and the return type of the function is parameterized over only the variables occurring in the type of its final argument. Even for ADTs there is a difference

between which folds over them we can type when they are viewed as ADTs (i.e., as fixpoints of first-order functors) versus as proper nested types (i.e., as fixpoints of higher-order functors). This is because, in the return type of `fold`, the arguments of the μ -type must be variables bound by `Nat`. For ADTs, the μ -type takes no arguments, making it possible to write recursive functions, such as a concatenation function for lists of type α ; $\emptyset \vdash \text{Nat}^\emptyset(\mu\beta.\mathbb{1} + \alpha \times \beta)(\text{Nat}^\emptyset(\mu\beta.\mathbb{1} + \alpha \times \beta)(\mu\beta.\mathbb{1} + \alpha \times \beta))$. This is not possible for nested types — even when they are semantically equivalent to ADTs.

Interestingly, even some recursive functions of a single proper nested type — e.g., a reverse function for bushes that is a true involution — cannot be expressed as folds because the algebra arguments needed to define them are again recursive functions with types of the same problematic form as the type of, e.g., a zip function for perfect trees. Expressivity of folds for nested types has long been a vexing issue, and this is naturally inherited by our calculus. Adding more expressive recursion combinators could help, but since this is orthogonal to the issue of parametricity in the presence of primitive nested types we do not consider it further here.

3. INTERPRETING TYPES

We denote the category of sets and functions by `Set`. The category `Rel` has as its objects triples (A, B, R) where R is a relation between the objects A and B in `Set`, i.e., a subset of $A \times B$, and has as its morphisms from (A, B, R) to (A', B', R') pairs $(f : A \rightarrow A', g : B \rightarrow B')$ of morphisms in `Set` such that $(fa, gb) \in R'$ whenever $(a, b) \in R$. We write $R : \text{Rel}(A, B)$ in place of (A, B, R) when convenient. If $R : \text{Rel}(A, B)$ we write $\pi_1 R$ and $\pi_2 R$ for the *domain* A of R and the *codomain* B of R , respectively. If $A : \text{Set}$, then we write $\text{Eq}_A = (A, A, \{(x, x) \mid x \in A\})$ for the *equality relation* on A .

The key 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 $R : \text{Rel}(A, B)$ to relations $F_1(R) : \text{Rel}(F_0(A), F_0(B))$, 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 function $t_0(A) : G_0(A) \rightarrow F_0(A)$. 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 $F_1(\text{Eq}_A) = \text{Eq}_{F_0(A)}$, 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 $R : \text{Rel}(A, B)$, $(t_0(A), t_0(B))$ 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.

The key to proving the Identity Extension Lemma in our setting (Theorem 4.4) is a familiar “cutting down” of the interpretations of universally quantified types to include only the “parametric” elements; the relevant types in our calculus are the `Nat`-types. This cutting down requires, as usual, that the set interpretations of types (Section 3.1) are defined simultaneously with their relational interpretations (Section 3.2). While the set interpretations are relatively straightforward, their relation interpretations are less so, mainly because of the cocontinuity conditions required to ensure that they are well-defined. We develop these conditions in Sections 3.1 and 3.2. This separates our set and relational interpretations in space, but otherwise has no impact on the fact that they are given by mutual induction.

3.1. Interpreting Types as Sets. We interpret types in our calculus as ω -cocontinuous functors on locally finitely presentable categories [AR94]. Both `Set` and `Rel` are locally finitely presentable categories. Since functor categories of locally finitely presentable categories are again locally finitely presentable, the fixpoints interpreting μ -types in `Set` and `Rel` must all exist, and thus the set and relational interpretations of all of the types in Definition 2.1, are well-defined [JP19]. To bootstrap this process, we interpret type variables as ω -cocontinuous functors in Definitions 3.1 and 3.9. If \mathcal{C}

and \mathcal{D} are locally finitely presentable categories, we write $[\mathcal{C}, \mathcal{D}]$ for the category of ω -cocontinuous functors from \mathcal{C} to \mathcal{D} .

Definition 3.1. A *set environment* maps each type variable in $\mathbb{T}^k \cup \mathbb{F}^k$ to an element of $[\mathbf{Set}^k, \mathbf{Set}]$. A morphism $f : \rho \rightarrow \rho'$ 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 \mathbf{Set} to the k -ary functor $\rho'\phi^k$ on \mathbf{Set} . 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, denoted \mathbf{SetEnv} .

When convenient we identify a functor in $[\mathbf{Set}^0, \mathbf{Set}]$ with its value on $*$ and consider a set environment to map a type variable of arity 0 to a set. If $\bar{\alpha} = \{\alpha_1, \dots, \alpha_k\}$ and $\bar{A} = \{A_1, \dots, A_k\}$, then we write $\rho[\bar{\alpha} := \bar{A}]$ for the set environment ρ' such that $\rho'\alpha_i = A_i$ for $i = 1, \dots, k$ and $\rho'\alpha = \rho\alpha$ if $\alpha \notin \{\alpha_1, \dots, \alpha_k\}$. We note that $\lambda\bar{A}. \rho[\bar{\alpha} := \bar{A}]$ is a functor in \bar{A} .

We can now define our set interpretation. Its action on objects of \mathbf{SetEnv} is given in Definition 3.2, and its action on morphisms of \mathbf{SetEnv} is given in Definition 3.3. If ρ is a set environment we write \mathbf{Eq}_ρ for the equality relation environment such that $\mathbf{Eq}_\rho\phi = \mathbf{Eq}_{\rho\phi}$ for every type variable ϕ ; see Definitions 3.4 and 3.9 for the definitions of a relation transformer and a relation environment, and Equation 4.1 for the definition of the relation transformer \mathbf{Eq}_F on a functor F . Equality relation environments appear in the third clause of Definition 3.2. The relational interpretation also appearing in the third clause of Definition 3.2 is given in Definition 3.11.

Definition 3.2. The *set interpretation* $\llbracket \cdot \rrbracket^{\mathbf{Set}} : \mathcal{F} \rightarrow [\mathbf{SetEnv}, \mathbf{Set}]$ is defined by:

$$\begin{aligned}
\llbracket \Gamma; \Phi \vdash 0 \rrbracket^{\mathbf{Set}} \rho &= 0 \\
\llbracket \Gamma; \Phi \vdash 1 \rrbracket^{\mathbf{Set}} \rho &= 1 \\
\llbracket \Gamma; \emptyset \vdash \mathbf{Nat}^{\bar{\alpha}} F G \rrbracket^{\mathbf{Set}} \rho &= \{ \eta : \lambda\bar{A}. \llbracket \Gamma; \bar{\alpha} \vdash F \rrbracket^{\mathbf{Set}} \rho[\bar{\alpha} := \bar{A}] \Rightarrow \lambda\bar{A}. \llbracket \Gamma; \bar{\alpha} \vdash G \rrbracket^{\mathbf{Set}} \rho[\bar{\alpha} := \bar{A}] \\
&\quad | \forall \bar{A}, \bar{B} : \mathbf{Set}. \forall R : \mathbf{Rel}(\bar{A}, \bar{B}). \\
&\quad (\eta_{\bar{A}}, \eta_{\bar{B}}) : \llbracket \Gamma; \bar{\alpha} \vdash F \rrbracket^{\mathbf{Rel}} \mathbf{Eq}_\rho[\bar{\alpha} := \bar{R}] \rightarrow \llbracket \Gamma; \bar{\alpha} \vdash G \rrbracket^{\mathbf{Rel}} \mathbf{Eq}_\rho[\bar{\alpha} := \bar{R}] \} \\
\llbracket \Gamma; \Phi \vdash \phi \bar{F} \rrbracket^{\mathbf{Set}} \rho &= (\rho\phi) \llbracket \Gamma; \Phi \vdash F \rrbracket^{\mathbf{Set}} \rho \\
\llbracket \Gamma; \Phi \vdash F + G \rrbracket^{\mathbf{Set}} \rho &= \llbracket \Gamma; \Phi \vdash F \rrbracket^{\mathbf{Set}} \rho + \llbracket \Gamma; \Phi \vdash G \rrbracket^{\mathbf{Set}} \rho \\
\llbracket \Gamma; \Phi \vdash F \times G \rrbracket^{\mathbf{Set}} \rho &= \llbracket \Gamma; \Phi \vdash F \rrbracket^{\mathbf{Set}} \rho \times \llbracket \Gamma; \Phi \vdash G \rrbracket^{\mathbf{Set}} \rho \\
\llbracket \Gamma; \Phi \vdash (\mu\phi. \lambda\bar{\alpha}. H) \bar{G} \rrbracket^{\mathbf{Set}} \rho &= (\mu T_{H, \rho}^{\mathbf{Set}}) \llbracket \Gamma; \Phi \vdash G \rrbracket^{\mathbf{Set}} \rho \\
&\quad \text{where } T_{H, \rho}^{\mathbf{Set}} F = \lambda\bar{A}. \llbracket \Gamma; \phi, \bar{\alpha} \vdash H \rrbracket^{\mathbf{Set}} \rho[\phi := F][\bar{\alpha} := \bar{A}] \\
&\quad \text{and } T_{H, \rho}^{\mathbf{Set}} \eta = \lambda\bar{A}. \llbracket \Gamma; \phi, \bar{\alpha} \vdash H \rrbracket^{\mathbf{Set}} id_\rho[\phi := \eta][\bar{\alpha} := id_{\bar{A}}]
\end{aligned}$$

If $\rho \in \mathbf{SetEnv}$ and $\vdash F$ then we write $\llbracket \vdash F \rrbracket^{\mathbf{Set}}$ instead of $\llbracket \vdash F \rrbracket^{\mathbf{Set}} \rho$ since the environment is immaterial. The third clause of Definition 3.2 does indeed define a set: local finite presentability of \mathbf{Set} and ω -cocontinuity of $\llbracket \Gamma; \bar{\alpha} \vdash F \rrbracket^{\mathbf{Set}}$ ensure that $\{ \eta : \lambda\bar{A}. \llbracket \Gamma; \bar{\alpha} \vdash F \rrbracket^{\mathbf{Set}} \rho[\bar{\alpha} := \bar{A}] \Rightarrow \lambda\bar{A}. \llbracket \Gamma; \bar{\alpha} \vdash G \rrbracket^{\mathbf{Set}} \rho[\bar{\alpha} := \bar{A}] \}$ (containing $\llbracket \Gamma; \emptyset \vdash \mathbf{Nat}^{\bar{\alpha}} F G \rrbracket^{\mathbf{Set}} \rho$) can be embedded in the set

$$\prod_{\substack{\bar{S} = (S_1, \dots, S_{|\bar{\alpha}|}) \\ S_1, \dots, S_{|\bar{\alpha}|} \text{ are finite cardinals}}} (\llbracket \Gamma; \bar{\alpha} \vdash G \rrbracket^{\mathbf{Set}} \rho[\bar{\alpha} := \bar{S}])^{(\llbracket \Gamma; \bar{\alpha} \vdash F \rrbracket^{\mathbf{Set}} \rho[\bar{\alpha} := \bar{S}])}$$

Also, $\llbracket \Gamma; \emptyset \vdash \mathbf{Nat}^{\bar{\alpha}} F G \rrbracket^{\mathbf{Set}}$ is ω -cocontinuous since it is constant (in particular, on ω -directed sets). This is because Definition 3.1 ensures that restrictions to \mathbb{T}^k of morphisms between set environments are identities. Interpretations of \mathbf{Nat} -types ensure that $\llbracket \Gamma \vdash F \rightarrow G \rrbracket^{\mathbf{Set}}$ and $\llbracket \Gamma \vdash \forall \bar{\alpha}. F \rrbracket^{\mathbf{Set}}$ are as expected in any parametric model.

To make sense of the last clause in Definition 3.2, we need to know that, for each $\rho \in \mathbf{SetEnv}$, $T_{H,\rho}^{\mathbf{Set}}$ is an ω -cocontinuous endofunctor on $[\mathbf{Set}^k, \mathbf{Set}]$, and thus admits a fixpoint. Since $T_{H,\rho}^{\mathbf{Set}}$ is defined in terms of $\llbracket \Gamma; \phi, \bar{\alpha} \vdash H \rrbracket^{\mathbf{Set}}$, 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 \mathbf{SetEnv} are intertwined. Fortunately, we know from [JP19] that, for every $\Gamma; \bar{\alpha} \vdash G$, $\llbracket \Gamma; \bar{\alpha} \vdash G \rrbracket^{\mathbf{Set}}$ is actually in $[\mathbf{Set}^k, \mathbf{Set}]$ where $k = |\bar{\alpha}|$. This means that for each $\llbracket \Gamma; \phi^k, \bar{\alpha} \vdash H \rrbracket^{\mathbf{Set}}$, the corresponding operator $T_H^{\mathbf{Set}}$ can be extended to a *functor* from \mathbf{SetEnv} to $[[\mathbf{Set}^k, \mathbf{Set}], [\mathbf{Set}^k, \mathbf{Set}]]$. The action of $T_H^{\mathbf{Set}}$ on an object $\rho \in \mathbf{SetEnv}$ is given by the higher-order functor $T_{H,\rho}^{\mathbf{Set}}$, whose actions on objects (functors in $[\mathbf{Set}^k, \mathbf{Set}]$) and morphisms (natural transformations) between them are given in Definition 3.2. Its action on a morphism $f : \rho \rightarrow \rho'$ is the higher-order natural transformation $T_{H,f}^{\mathbf{Set}} : T_{H,\rho}^{\mathbf{Set}} \rightarrow T_{H,\rho'}^{\mathbf{Set}}$ whose action on $F : [\mathbf{Set}^k, \mathbf{Set}]$ is the natural transformation $T_{H,f}^{\mathbf{Set}} F : T_{H,\rho}^{\mathbf{Set}} F \rightarrow T_{H,\rho'}^{\mathbf{Set}} F$ whose component at \bar{A} is $(T_{H,f}^{\mathbf{Set}} F)_{\bar{A}} = \llbracket \Gamma; \phi, \bar{\alpha} \vdash H \rrbracket^{\mathbf{Set}} f[\phi := id_F][\bar{\alpha} := id_{\bar{A}}]$. Note that this action of $T_{H,\rho}^{\mathbf{Set}}$ on morphisms appears in the final clause of Definition 3.3 below. It will indeed be well-defined there because the functorial action of $\llbracket \Gamma; \phi, \bar{\alpha} \vdash H \rrbracket^{\mathbf{Set}}$ will already be available from the induction hypothesis on H .

Using $T_H^{\mathbf{Set}}$, we can define the functorial action of set interpretation.

Definition 3.3. Let $f : \rho \rightarrow \rho'$ be a morphism between set environments ρ and ρ' (so that $\rho|_{\mathbb{T}} = \rho'|_{\mathbb{T}}$). The action $\llbracket \Gamma; \Phi \vdash F \rrbracket^{\mathbf{Set}} f$ of $\llbracket \Gamma; \Phi \vdash F \rrbracket^{\mathbf{Set}}$ on f is given by:

- If $\Gamma; \Phi \vdash 0$ then $\llbracket \Gamma; \Phi \vdash 0 \rrbracket^{\mathbf{Set}} f = id_0$
- If $\Gamma; \Phi \vdash 1$ then $\llbracket \Gamma; \Phi \vdash 1 \rrbracket^{\mathbf{Set}} f = id_1$
- If $\Gamma; \emptyset \vdash \text{Nat}^{\bar{\alpha}} F G$ then $\llbracket \Gamma; \emptyset \vdash \text{Nat}^{\bar{\alpha}} F G \rrbracket^{\mathbf{Set}} f = id_{\llbracket \Gamma; \emptyset \vdash \text{Nat}^{\bar{\alpha}} F G \rrbracket^{\mathbf{Set}} \rho}$
- If $\Gamma; \Phi \vdash \phi \bar{F}$ then $\llbracket \Gamma; \Phi \vdash \phi \bar{F} \rrbracket^{\mathbf{Set}} f : \llbracket \Gamma; \Phi \vdash \phi \bar{F} \rrbracket^{\mathbf{Set}} \rho \rightarrow \llbracket \Gamma; \Phi \vdash \phi \bar{F} \rrbracket^{\mathbf{Set}} \rho' = (\rho\phi) \overline{\llbracket \Gamma; \Phi \vdash F \rrbracket^{\mathbf{Set}} \rho} \rightarrow (\rho'\phi) \overline{\llbracket \Gamma; \Phi \vdash F \rrbracket^{\mathbf{Set}} \rho'}$ is defined by $\llbracket \Gamma; \Phi \vdash \phi \bar{F} \rrbracket^{\mathbf{Set}} f = (f\phi) \overline{\llbracket \Gamma; \Phi \vdash F \rrbracket^{\mathbf{Set}} \rho} \circ (\rho\phi) \overline{\llbracket \Gamma; \Phi \vdash F \rrbracket^{\mathbf{Set}} f} = (\rho'\phi) \overline{\llbracket \Gamma; \Phi \vdash F \rrbracket^{\mathbf{Set}} f} \circ (f\phi) \overline{\llbracket \Gamma; \Phi \vdash F \rrbracket^{\mathbf{Set}} \rho}$. The latter equality holds because $\rho\phi$ and $\rho'\phi$ are functors and $f\phi : \rho\phi \rightarrow \rho'\phi$ is a natural transformation, so the following naturality square commutes:

$$\begin{array}{ccc}
 (\rho\phi) \overline{\llbracket \Gamma; \Phi \vdash F \rrbracket^{\mathbf{Set}} \rho} & \xrightarrow{(f\phi) \overline{\llbracket \Gamma; \Phi \vdash F \rrbracket^{\mathbf{Set}} \rho}} & (\rho'\phi) \overline{\llbracket \Gamma; \Phi \vdash F \rrbracket^{\mathbf{Set}} \rho} \\
 \downarrow (\rho\phi) \overline{\llbracket \Gamma; \Phi \vdash F \rrbracket^{\mathbf{Set}} f} & & \downarrow (\rho'\phi) \overline{\llbracket \Gamma; \Phi \vdash F \rrbracket^{\mathbf{Set}} f} \\
 (\rho\phi) \overline{\llbracket \Gamma; \Phi \vdash F \rrbracket^{\mathbf{Set}} \rho'} & \xrightarrow{(f\phi) \overline{\llbracket \Gamma; \Phi \vdash F \rrbracket^{\mathbf{Set}} \rho'}} & (\rho'\phi) \overline{\llbracket \Gamma; \Phi \vdash F \rrbracket^{\mathbf{Set}} \rho'}
 \end{array} \tag{3.1}$$

- If $\Gamma; \Phi \vdash F + G$ then $\llbracket \Gamma; \Phi \vdash F + G \rrbracket^{\mathbf{Set}} f$ is defined by $\llbracket \Gamma; \Phi \vdash F + G \rrbracket^{\mathbf{Set}} f(\text{inl } x) = \text{inl } (\llbracket \Gamma; \Phi \vdash F \rrbracket^{\mathbf{Set}} f x)$ and $\llbracket \Gamma; \Phi \vdash F + G \rrbracket^{\mathbf{Set}} f(\text{inr } y) = \text{inr } (\llbracket \Gamma; \Phi \vdash G \rrbracket^{\mathbf{Set}} f y)$
- If $\Gamma; \Phi \vdash F \times G$ then $\llbracket \Gamma; \Phi \vdash F \times G \rrbracket^{\mathbf{Set}} f = \llbracket \Gamma; \Phi \vdash F \rrbracket^{\mathbf{Set}} f \times \llbracket \Gamma; \Phi \vdash G \rrbracket^{\mathbf{Set}} f$
- If $\Gamma; \Phi \vdash (\mu\phi.\lambda\bar{\alpha}.H)\bar{G}$ then

$$\begin{aligned}
 \llbracket \Gamma; \Phi \vdash (\mu\phi.\lambda\bar{\alpha}.H)\bar{G} \rrbracket^{\mathbf{Set}} f & : \llbracket \Gamma; \Phi \vdash (\mu\phi.\lambda\bar{\alpha}.H)\bar{G} \rrbracket^{\mathbf{Set}} \rho \rightarrow \llbracket \Gamma; \Phi \vdash (\mu\phi.\lambda\bar{\alpha}.H)\bar{G} \rrbracket^{\mathbf{Set}} \rho' \\
 & = (\mu T_{H,\rho}^{\mathbf{Set}}) \overline{\llbracket \Gamma; \Phi \vdash G \rrbracket^{\mathbf{Set}} \rho} \rightarrow (\mu T_{H,\rho'}^{\mathbf{Set}}) \overline{\llbracket \Gamma; \Phi \vdash G \rrbracket^{\mathbf{Set}} \rho'}
 \end{aligned}$$

is defined by

$$\begin{aligned}
 & (\mu T_{H,f}^{\mathbf{Set}}) \overline{\llbracket \Gamma; \Phi \vdash G \rrbracket^{\mathbf{Set}} \rho'} \circ (\mu T_{H,\rho}^{\mathbf{Set}}) \overline{\llbracket \Gamma; \Phi \vdash G \rrbracket^{\mathbf{Set}} f} \\
 & = (\mu T_{H,\rho'}^{\mathbf{Set}}) \overline{\llbracket \Gamma; \Phi \vdash G \rrbracket^{\mathbf{Set}} f} \circ (\mu T_{H,f}^{\mathbf{Set}}) \overline{\llbracket \Gamma; \Phi \vdash G \rrbracket^{\mathbf{Set}} \rho}
 \end{aligned}$$

The latter equality holds because $\mu T_{H,\rho}^{\mathbf{Set}}$ and $\mu T_{H,\rho'}^{\mathbf{Set}}$ are functors and $\mu T_{H,f}^{\mathbf{Set}} : \mu T_{H,\rho}^{\mathbf{Set}} \rightarrow \mu T_{H,\rho'}^{\mathbf{Set}}$ is a natural transformation, so the following naturality square commutes:

$$\begin{array}{ccc}
 (\mu T_{H,\rho}^{\mathbf{Set}}) \overline{\llbracket \Gamma; \Phi \vdash G \rrbracket^{\mathbf{Set}} \rho} & \xrightarrow{(\mu T_{H,f}^{\mathbf{Set}}) \overline{\llbracket \Gamma; \Phi \vdash G \rrbracket^{\mathbf{Set}} \rho}} & (\mu T_{H,\rho'}^{\mathbf{Set}}) \overline{\llbracket \Gamma; \Phi \vdash G \rrbracket^{\mathbf{Set}} \rho} \\
 \downarrow (\mu T_{H,\rho}^{\mathbf{Set}}) \overline{\llbracket \Gamma; \Phi \vdash G \rrbracket^{\mathbf{Set}} f} & & \downarrow (\mu T_{H,\rho'}^{\mathbf{Set}}) \overline{\llbracket \Gamma; \Phi \vdash G \rrbracket^{\mathbf{Set}} f} \\
 (\mu T_{H,\rho}^{\mathbf{Set}}) \overline{\llbracket \Gamma; \Phi \vdash G \rrbracket^{\mathbf{Set}} \rho'} & \xrightarrow{(\mu T_{H,f}^{\mathbf{Set}}) \overline{\llbracket \Gamma; \Phi \vdash G \rrbracket^{\mathbf{Set}} \rho'}} & (\mu T_{H,\rho'}^{\mathbf{Set}}) \overline{\llbracket \Gamma; \Phi \vdash G \rrbracket^{\mathbf{Set}} \rho'}
 \end{array} \tag{3.2}$$

Definitions 3.2 and 3.3 respect weakening, i.e., ensure that a type and its weakenings have the same set interpretations.

3.2. Interpreting Types as Relations.

Definition 3.4. A k -ary relation transformer F is a triple (F^1, F^2, F^*) , where

- $F^1, F^2 : [\text{Set}^k, \text{Set}]$ and $F^* : [\text{Rel}^k, \text{Rel}]$ are functors
- If $R_1 : \text{Rel}(A_1, B_1), \dots, R_k : \text{Rel}(A_k, B_k)$, then $F^* \overline{R} : \text{Rel}(F^1 \overline{A}, F^2 \overline{B})$
- If $(\alpha_1, \beta_1) \in \text{Hom}_{\text{Rel}}(R_1, S_1), \dots, (\alpha_k, \beta_k) \in \text{Hom}_{\text{Rel}}(R_k, S_k)$ then $F^*(\overline{\alpha}, \overline{\beta}) = (F^1 \overline{\alpha}, F^2 \overline{\beta})$

We define $F \overline{R}$ to be $F^* \overline{R}$ and $F(\overline{\alpha}, \overline{\beta})$ to be $F^*(\overline{\alpha}, \overline{\beta})$.

The last clause of Definition 3.4 expands to: if $(a, b) \in \overline{R}$ implies $(\overline{\alpha} a, \overline{\beta} b) \in \overline{S}$ then $(c, d) \in F^* \overline{R}$ implies $(F^1 \overline{\alpha} c, F^2 \overline{\beta} d) \in F^* \overline{S}$. When convenient we identify a 0-ary relation transformer (A, B, R) with $R : \text{Rel}(A, B)$, and write $\pi_1 F$ for F^1 and $\pi_2 F$ for F^2 . Below we extend these conventions to relation environments in the obvious ways.

Definition 3.5. The category RT_k of k -ary relation transformers is given by the following data:

- An object of RT_k is a relation transformer
- A morphism $\delta : (F^1, F^2, F^*) \rightarrow (G^1, G^2, G^*)$ in RT_k is a pair of natural transformations (δ^1, δ^2) , where $\delta^1 : F^1 \rightarrow G^1$ and $\delta^2 : F^2 \rightarrow G^2$, such that, for all $\overline{R} : \text{Rel}(A, B)$, if $(x, y) \in F^* \overline{R}$ then $(\delta^1_{\overline{A}} x, \delta^2_{\overline{B}} y) \in G^* \overline{R}$
- Identity morphisms and composition are inherited from the category of functors on Set

Definition 3.6. An endofunctor H on RT_k is a triple $H = (H^1, H^2, H^*)$, where

- H^1 and H^2 are functors from $[\text{Set}^k, \text{Set}]$ to $[\text{Set}^k, \text{Set}]$
- H^* is a functor from RT_k to $[\text{Rel}^k, \text{Rel}]$
- For all $\overline{R} : \text{Rel}(A, B)$, $\pi_1((H^*(\delta^1, \delta^2))_{\overline{R}}) = (H^1 \delta^1)_{\overline{A}}$ and $\pi_2((H^*(\delta^1, \delta^2))_{\overline{R}}) = (H^2 \delta^2)_{\overline{B}}$
- The action of H on objects is given by $H(F^1, F^2, F^*) = (H^1 F^1, H^2 F^2, H^*(F^1, F^2, F^*))$
- The action of H on morphisms is given by $H(\delta^1, \delta^2) = (H^1 \delta^1, H^2 \delta^2)$ for $(\delta^1, \delta^2) : (F^1, F^2, F^*) \rightarrow (G^1, G^2, G^*)$

Since the results of applying an endofunctor H to k -ary relation transformers and morphisms between them must again be k -ary relation transformers and morphisms between them, respectively, Definition 3.6 implicitly requires that the following three conditions hold:

- $H^*(F^1, F^2, F^*) \overline{R} : \text{Rel}(H^1 F^1 \overline{A}, H^2 F^2 \overline{B})$ whenever $R_1 : \text{Rel}(A_1, B_1), \dots, R_k : \text{Rel}(A_k, B_k)$
- $H^*(F^1, F^2, F^*) (\overline{\alpha}, \overline{\beta}) = (H^1 F^1 \overline{\alpha}, H^2 F^2 \overline{\beta})$ whenever $(\alpha_1, \beta_1) \in \text{Hom}_{\text{Rel}}(R_1, S_1), \dots, (\alpha_k, \beta_k) \in \text{Hom}_{\text{Rel}}(R_k, S_k)$
- If $(\delta^1, \delta^2) : (F^1, F^2, F^*) \rightarrow (G^1, G^2, G^*)$ in RT_k and $R_1 : \text{Rel}(A_1, B_1), \dots, R_k : \text{Rel}(A_k, B_k)$, then $((H^1 \delta^1)_{\overline{A}} x, (H^2 \delta^2)_{\overline{B}} y) \in H^*(G^1, G^2, G^*) \overline{R}$ whenever $(x, y) \in H^*(F^1, F^2, F^*) \overline{R}$

Note, however, that this last condition is automatically satisfied because it is implied by the third bullet point of Definition 3.6.

Definition 3.7. For endofunctors H and K on RT_k , a *natural transformation* $\sigma : H \rightarrow K$ is a pair $\sigma = (\sigma^1, \sigma^2)$, where $\sigma^1 : H^1 \rightarrow K^1$ and $\sigma^2 : H^2 \rightarrow K^2$ are natural transformations between endofunctors on $[\text{Set}^k, \text{Set}]$ and the component of σ at $F = (F^1, F^2, F^*) \in RT_k$ is given by $\sigma_F = (\sigma^1_{F^1}, \sigma^2_{F^2})$.

Definition 3.7 entails that $\sigma^i_{F^i}$ is natural in $F^i : [\text{Set}^k, \text{Set}]$, and, for every F , both $(\sigma^1_{F^1})_{\overline{A}}$ and $(\sigma^2_{F^2})_{\overline{B}}$ are natural in \overline{A} . Moreover, since the results of applying σ to k -ary relation transformers must be morphisms of k -ary relation transformers, Definition 3.7 implicitly requires that $(\sigma_F)_{\overline{R}} =$

$((\sigma_{F^1}^1)_{\overline{A}}, (\sigma_{F^2}^2)_{\overline{B}})$ is a morphism in Rel for any k -tuple of relations $\overline{R} : \text{Rel}(A, B)$, i.e., that if $(x, y) \in H^* F \overline{R}$, then $((\sigma_{F^1}^1)_{\overline{A}} x, (\sigma_{F^2}^2)_{\overline{B}} y) \in K^* F \overline{R}$.

Critically, we can compute ω -directed colimits in RT_k . Indeed, if \mathcal{D} is an ω -directed set then $\varinjlim_{d \in \mathcal{D}} (F_d^1, F_d^2, F_d^*) = (\varinjlim_{d \in \mathcal{D}} F_d^1, \varinjlim_{d \in \mathcal{D}} F_d^2, \varinjlim_{d \in \mathcal{D}} F_d^*)$. We define an endofunctor $T = (T^1, T^2, T^*)$ on RT_k to be ω -cocontinuous if T^1 and T^2 are ω -cocontinuous endofunctors on $[\text{Set}^k, \text{Set}]$ and T^* is an ω -cocontinuous functor from RT_k to $[\text{Rel}^k, \text{Rel}]$, i.e., is in $[RT_k, [\text{Rel}^k, \text{Rel}]]$. Now, for any k , any $A : \text{Set}$, and any $R : \text{Rel}(A, B)$, let K_A^{Set} be the constantly A -valued functor from Set^k to Set and K_R^{Rel} be the constantly R -valued functor from Rel^k to Rel . Also let 0 denote either the initial object of either Set or Rel , as appropriate. Observing that, for every k , K_0^{Set} is initial in $[\text{Set}^k, \text{Set}]$, and K_0^{Rel} is initial in $[\text{Rel}^k, \text{Rel}]$, we have that, for each k , $K_0 = (K_0^{\text{Set}}, K_0^{\text{Set}}, K_0^{\text{Rel}})$ is initial in RT_k . Thus, if $T = (T^1, T^2, T^*) : RT_k \rightarrow RT_k$ is an endofunctor on RT_k then we can define the relation transformer μT to be $\varinjlim_{n \in \mathbb{N}} T^n K_0$. It is not hard to see that μT is given explicitly as

$$\mu T = (\mu T^1, \mu T^2, \varinjlim_{n \in \mathbb{N}} (T^n K_0)^*) \quad (3.3)$$

Moreover, μT really is a fixpoint for T if T is ω -cocontinuous:

Lemma 3.8. *For any $T : [RT_k, RT_k]$, $\mu T \cong T(\mu T)$.*

The isomorphism is given by the morphisms $(in_1, in_2) : T(\mu T) \rightarrow \mu T$ and $(in_1^{-1}, in_2^{-1}) : \mu T \rightarrow T(\mu T)$ in RT_k . The latter is always a morphism in RT_k , but the former need not be if T is not ω -cocontinuous.

It is worth noting that the third component in Equation (3.3) is the colimit in $[\text{Rel}^k, \text{Rel}]$ of third components of relation transformers, rather than a fixpoint of an endofunctor on $[\text{Rel}^k, \text{Rel}]$. There is thus an asymmetry between the first two components of μT and its third component, which reflects the important conceptual observation that the third component of an endofunctor on RT_k need not be a functor on all of $[\text{Rel}^k, \text{Rel}]$. In particular, although we can define $T_{H, \rho} F$ for a relation transformer F in Definition 3.11 below, it is not clear how we could define it for an arbitrary $F : [\text{Rel}^k, \text{Rel}]$.

Definition 3.9. A *relation environment* maps each type variable in $\mathbb{T} \cup \mathbb{F}^k$ to a k -ary relation transformer. A morphism $f : \rho \rightarrow \rho'$ between relation environments ρ and ρ' with $\rho|_{\mathbb{T}} = \rho'|_{\mathbb{T}}$ maps each type constructor variable $\psi^k \in \mathbb{T}$ to the identity morphism on $\rho\psi^k = \rho'\psi^k$ and each functorial variable $\phi^k \in \mathbb{F}$ to a morphism from the k -ary relation transformer $\rho\phi$ to the k -ary relation transformer $\rho'\phi$. Composition of morphisms on relation environments is given componentwise, with the identity morphism mapping each relation environment to itself. This gives a category of relation environments and morphisms between them, denoted RelEnv .

We identify a 0-ary relation transformer with the relation (transformer) that is its codomain and consider a relation environment to map a type variable of arity 0 to a relation. We write $\rho[\alpha := \overline{R}]$ for the relation environment ρ' such that $\rho'\alpha_i = R_i$ for $i = 1, \dots, k$ and $\rho'\alpha = \rho\alpha$ if $\alpha \notin \{\alpha_1, \dots, \alpha_k\}$. If ρ is a relation environment, we write $\pi_1 \rho$ and $\pi_2 \rho$ for the set environments mapping each type variable ϕ to the functors $(\rho\phi)^1$ and $(\rho\phi)^2$, respectively.

Definition 3.10. For each k , an ω -cocontinuous functor $H \in [\text{RelEnv}, RT_k]$ is a triple $H = (H^1, H^2, H^*)$, where

- H^1 and H^2 are objects in $[\text{SetEnv}, [\text{Set}^k, \text{Set}]]$
- H^* is an object in $[\text{RelEnv}, [\text{Rel}^k, \text{Rel}]]$
- For all $\overline{R} : \text{Rel}(A, B)$ and morphisms f in RelEnv , $\pi_1(H^* f \overline{R}) = H^1(\pi_1 f) \overline{A}$ and $\pi_2(H^* f \overline{R}) = H^2(\pi_2 f) \overline{B}$
- The action of H on ρ in RelEnv is given by $H\rho = (H^1(\pi_1 \rho), H^2(\pi_2 \rho), H^* \rho)$
- The action of H on morphisms $f : \rho \rightarrow \rho'$ in RelEnv is given by $Hf = (H^1(\pi_1 f), H^2(\pi_2 f))$

Spelling out the last two bullet points above gives the following analogues of the three conditions immediately following Definition 3.6:

- $H^*\rho \overline{R} : \text{Rel}(H^1(\pi_1\rho) \overline{A}, H^2(\pi_2\rho) \overline{B})$ whenever $R_1 : \text{Rel}(A_1, B_1), \dots, R_k : \text{Rel}(A_k, B_k)$
- $H^*\rho(\overline{\alpha}, \overline{\beta}) = (H^1(\pi_1\rho) \overline{\alpha}, H^2(\pi_2\rho) \overline{\beta})$ whenever $(\alpha_1, \beta_1) \in \text{Hom}_{\text{Rel}}(R_1, S_1), \dots, (\alpha_k, \beta_k) \in \text{Hom}_{\text{Rel}}(R_k, S_k)$
- If $f : \rho \rightarrow \rho'$ and $R_1 : \text{Rel}(A_1, B_1), \dots, R_k : \text{Rel}(A_k, B_k)$, then $(H^1(\pi_1 f) \overline{A} x, H^2(\pi_2 f) \overline{B} y) \in H^*\rho' \overline{R}$ whenever $(x, y) \in H^*\rho \overline{R}$.

As before, the last condition is automatically satisfied because it is implied by the third bullet point of Definition 3.10.

Considering RelEnv as a subcategory of the product $\prod_{\phi^k \in \mathbb{T} \cup \mathbb{F}} RT_k$, computation of ω -directed colimits in RT_k extends componentwise to RelEnv . Similarly, ω -cocontinuity for endofunctors on RT_k extends componentwise to functors from RelEnv to RT_k . Recalling from the start of this subsection that Definition 3.11 is given mutually inductively with Definition 3.2 we can now define our relational interpretation.

Definition 3.11. The *relational interpretation* $\llbracket \cdot \rrbracket^{\text{Rel}} : \mathcal{F} \rightarrow [\text{RelEnv}, \text{Rel}]$ is defined by

$$\begin{aligned}
\llbracket \Gamma; \Phi \vdash 0 \rrbracket^{\text{Rel}} \rho &= 0 \\
\llbracket \Gamma; \Phi \vdash 1 \rrbracket^{\text{Rel}} \rho &= 1 \\
\llbracket \Gamma; \emptyset \vdash \text{Nat}^{\overline{\alpha}} F G \rrbracket^{\text{Rel}} \rho &= \{ \eta : \lambda \overline{R}. \llbracket \Gamma; \overline{\alpha} \vdash F \rrbracket^{\text{Rel}} \rho[\overline{\alpha} := \overline{R}] \Rightarrow \lambda \overline{R}. \llbracket \Gamma; \overline{\alpha} \vdash G \rrbracket^{\text{Rel}} \rho[\overline{\alpha} := \overline{R}] \} \\
&= \{ (t, t') \in \llbracket \Gamma; \emptyset \vdash \text{Nat}^{\overline{\alpha}} F G \rrbracket^{\text{Set}}(\pi_1 \rho) \times \llbracket \Gamma; \emptyset \vdash \text{Nat}^{\overline{\alpha}} F G \rrbracket^{\text{Set}}(\pi_2 \rho) \mid \\
&\quad \forall R_1 : \text{Rel}(A_1, B_1) \dots R_k : \text{Rel}(A_k, B_k). \\
&\quad (t_{\overline{A}}, t'_{\overline{B}}) \in (\llbracket \Gamma; \overline{\alpha} \vdash G \rrbracket^{\text{Rel}} \rho[\overline{\alpha} := \overline{R}])^{\llbracket \Gamma; \overline{\alpha} \vdash F \rrbracket^{\text{Rel}} \rho[\overline{\alpha} := \overline{R}]} \} \\
\llbracket \Gamma; \Phi \vdash \phi \overline{F} \rrbracket^{\text{Rel}} \rho &= (\rho \phi) \llbracket \Gamma; \Phi \vdash F \rrbracket^{\text{Rel}} \rho \\
\llbracket \Gamma; \Phi \vdash F + G \rrbracket^{\text{Rel}} \rho &= \llbracket \Gamma; \Phi \vdash F \rrbracket^{\text{Rel}} \rho + \llbracket \Gamma; \Phi \vdash G \rrbracket^{\text{Rel}} \rho \\
\llbracket \Gamma; \Phi \vdash F \times G \rrbracket^{\text{Rel}} \rho &= \llbracket \Gamma; \Phi \vdash F \rrbracket^{\text{Rel}} \rho \times \llbracket \Gamma; \Phi \vdash G \rrbracket^{\text{Rel}} \rho \\
\llbracket \Gamma; \Phi \vdash (\mu \phi. \lambda \overline{\alpha}. H) \overline{G} \rrbracket^{\text{Rel}} \rho &= (\mu T_{H, \rho}) \llbracket \Gamma; \Phi \vdash G \rrbracket^{\text{Rel}} \rho \\
&\quad \text{where } T_{H, \rho} = (T_{H, \pi_1 \rho}^{\text{Set}}, T_{H, \pi_2 \rho}^{\text{Set}}, T_{H, \rho}^{\text{Rel}}) \\
&\quad \text{and } T_{H, \rho}^{\text{Rel}} F = \lambda \overline{R}. \llbracket \Gamma; \phi, \overline{\alpha} \vdash H \rrbracket^{\text{Rel}} \rho[\phi := F][\overline{\alpha} := \overline{R}] \\
&\quad \text{and } T_{H, \rho}^{\text{Rel}} \delta = \lambda \overline{R}. \llbracket \Gamma; \phi, \overline{\alpha} \vdash H \rrbracket^{\text{Rel}} id_{\rho}[\phi := \delta][\overline{\alpha} := id_{\overline{R}}]
\end{aligned}$$

The interpretations in Definitions 3.11 and 3.12 below respect weakening, and also ensure that $\llbracket \Gamma \vdash F \rightarrow G \rrbracket^{\text{Rel}}$ and $\llbracket \Gamma \vdash \forall \overline{\alpha}. F \rrbracket^{\text{Rel}}$ are as expected in any parametric model. As for set interpretations, $\llbracket \Gamma; \emptyset \vdash \text{Nat}^{\overline{\alpha}} F G \rrbracket^{\text{Rel}}$ is ω -cocontinuous; indeed, it is constant (on ω -directed sets) because Definition 3.9 ensures that restrictions to \mathbb{T}^k of morphisms between relational environments are identities. If $\rho \in \text{RelEnv}$ and $\vdash F$, then we write $\llbracket \vdash F \rrbracket^{\text{Rel}}$ instead of $\llbracket \vdash F \rrbracket^{\text{Rel}} \rho$. For the last clause in Definition 3.11 to be well-defined we need $T_{H, \rho}$ to be an ω -cocontinuous endofunctor on RT so that, by Lemma 3.8, it admits a fixpoint. Since $T_{H, \rho}$ is defined in terms of $\llbracket \Gamma; \phi, \overline{\alpha} \vdash H \rrbracket^{\text{Rel}}$, this means that relational interpretations of types must be ω -cocontinuous functors from RelEnv to RT_0 , which in turn entails that the actions of relational interpretations of types on objects and on morphisms in RelEnv are intertwined. As for set interpretations, we know from [JP19] that, for every $\Gamma; \overline{\alpha} \vdash F$, $\llbracket \Gamma; \overline{\alpha} \vdash F \rrbracket^{\text{Rel}}$ is actually in $[\text{Rel}^k, \text{Rel}]$ where $k = |\overline{\alpha}|$. We first define the actions of each of these functors on morphisms between environments, and then argue that they are well-defined and have the required properties. To do this, we extend T_H to a *functor* from RelEnv to $[[\text{Rel}^k, \text{Rel}], [\text{Rel}^k, \text{Rel}]]$. Its action on an object $\rho \in \text{RelEnv}$ is given by the higher-order functor $T_{H, \rho}$ whose actions on objects and morphisms are given in Definition 3.12. Its action on a morphism $f : \rho \rightarrow \rho'$ is the higher-order natural transformation $T_{H, f} : T_{H, \rho} \rightarrow T_{H, \rho'}$ whose action on any $F : [\text{Rel}^k, \text{Rel}]$ is the natural transformation $T_{H, f} F : T_{H, \rho} F \rightarrow T_{H, \rho'} F$ whose component at \overline{R} is $(T_{H, f} F)_{\overline{R}} = \llbracket \Gamma; \phi, \overline{\alpha} \vdash H \rrbracket^{\text{Rel}} f[\phi := id_F][\overline{\alpha} := id_{\overline{R}}]$.

Using T_H , we can define the functorial action of relational interpretation.

Definition 3.12. Let $f : \rho \rightarrow \rho'$ for relation environments ρ and ρ' (so that $\rho|_{\mathbb{T}} = \rho'|_{\mathbb{T}}$). The action $\llbracket \Gamma; \Phi \vdash F \rrbracket^{\text{Rel}} f$ of $\llbracket \Gamma; \Phi \vdash F \rrbracket^{\text{Rel}}$ on the morphism f is given exactly as in Definition 3.3, except that all interpretations are relational interpretations and all occurrences of $T_{H,f}^{\text{Set}}$ are replaced by $T_{H,f}$.

For the functors given in Definitions 3.11 and 3.12 to be well-defined we need that, for every H , $T_{H,\rho} F$ is a relation transformer for any relation transformer F , and that $T_{H,f} F : T_{H,\rho} F \rightarrow T_{H,\rho'} F$ is a morphism of relation transformers for every relation transformer F and every morphism $f : \rho \rightarrow \rho'$ in RelEnv . If we define $\llbracket \Gamma; \Phi \vdash F \rrbracket$ to be $(\llbracket \Gamma; \Phi \vdash F \rrbracket^{\text{Set}}, \llbracket \Gamma; \Phi \vdash F \rrbracket^{\text{Set}}, \llbracket \Gamma; \Phi \vdash F \rrbracket^{\text{Rel}})$, then this is an immediate consequence of

Lemma 3.13. *For every $\Gamma; \Phi \vdash F$, $\llbracket \Gamma; \Phi \vdash F \rrbracket \in [\text{RelEnv}, RT_0]$.*

The proof is a straightforward induction on the structure of F , using an appropriate result from [JP19] to deduce ω -cocontinuity of $\llbracket \Gamma; \Phi \vdash F \rrbracket$ in each case, together with Lemma 3.8 and Equation 3.3 for μ -types.

We can prove by simultaneous induction that our interpretations of types interact well with demotion of functorial variables to non-functorial ones, along with other useful identities. Indeed, if $\rho, \rho' : \text{SetEnv}$, $f : \rho \rightarrow \rho'$, $\rho\phi = \rho\psi = \rho'\phi = \rho'\psi$, $f\phi = f\psi = \text{id}_{\rho\phi}$, $\Gamma; \Phi, \phi^k \vdash F$, $\Gamma; \Phi, \bar{\alpha} \vdash G$, $\Gamma; \Phi, \alpha_1 \dots \alpha_k \vdash H$, and $\Gamma; \Phi \vdash K$, then

$$\llbracket \Gamma; \Phi, \phi \vdash F \rrbracket^{\text{Set}} \rho = \llbracket \Gamma, \psi; \Phi \vdash F[\phi := \psi] \rrbracket^{\text{Set}} \rho \quad (3.4)$$

$$\llbracket \Gamma; \Phi, \phi \vdash F \rrbracket^{\text{Set}} f = \llbracket \Gamma, \psi; \Phi \vdash F[\phi := \psi] \rrbracket^{\text{Set}} f \quad (3.5)$$

$$\llbracket \Gamma; \Phi \vdash G[\bar{\alpha} := \bar{K}] \rrbracket^{\text{Set}} \rho = \llbracket \Gamma; \Phi, \bar{\alpha} \vdash G \rrbracket^{\text{Set}} \rho[\bar{\alpha} := \llbracket \Gamma; \Phi \vdash K \rrbracket^{\text{Set}} \rho] \quad (3.6)$$

$$\llbracket \Gamma; \Phi \vdash G[\bar{\alpha} := \bar{K}] \rrbracket^{\text{Set}} f = \llbracket \Gamma; \Phi, \bar{\alpha} \vdash G \rrbracket^{\text{Set}} f[\bar{\alpha} := \llbracket \Gamma; \Phi \vdash K \rrbracket^{\text{Set}} f] \quad (3.7)$$

$$\llbracket \Gamma; \Phi \vdash F[\phi := H] \rrbracket^{\text{Set}} \rho = \llbracket \Gamma; \Phi, \phi \vdash F \rrbracket^{\text{Set}} \rho[\phi := \lambda \bar{A}. \llbracket \Gamma; \Phi, \bar{\alpha} \vdash H \rrbracket^{\text{Set}} \rho[\bar{\alpha} := \bar{A}]] \quad (3.8)$$

$$\llbracket \Gamma; \Phi \vdash F[\phi := H] \rrbracket^{\text{Set}} f = \llbracket \Gamma; \Phi, \phi \vdash F \rrbracket^{\text{Set}} f[\phi := \lambda \bar{A}. \llbracket \Gamma; \Phi, \bar{\alpha} \vdash H \rrbracket^{\text{Set}} f[\bar{\alpha} := \text{id}_{\bar{A}}]] \quad (3.9)$$

Identities analogous to (3.4) through (3.9) hold for relational interpretations as well.

4. THE IDENTITY EXTENSION LEMMA

In most treatments of parametricity, equality relations on sets are taken as *given* — either directly as diagonal relations, or perhaps via reflexive graphs if kinds are also being tracked — and the graph relations used to validate existence of initial algebras are defined in terms of them. We take a different approach here, giving a categorical definition of graph relations for morphisms (i.e., natural transformations) between functors and *constructing* equality relations as particular graph relations. Our definitions specialize to the usual ones for the graph relation for morphisms between sets and equality relations on sets. In light of its novelty, we spell out our construction in detail.

The standard definition of the graph for a morphism $f : A \rightarrow B$ in Set is the relation $\langle f \rangle : \text{Rel}(A, B)$ defined by $(x, y) \in \langle f \rangle$ iff $fx = y$. This definition naturally generalizes to associate to each natural transformation between k -ary functors on Set a k -ary relation transformer as follows:

Definition 4.1. If $F, G : \text{Set}^k \rightarrow \text{Set}$ and $\alpha : F \rightarrow G$ is a natural transformation, then the functor $\langle \alpha \rangle^* : \text{Rel}^k \rightarrow \text{Rel}$ is defined as follows. Given $R_1 : \text{Rel}(A_1, B_1), \dots, R_k : \text{Rel}(A_k, B_k)$, let $\iota_{R_i} : R_i \hookrightarrow A_i \times B_i$, for $i = 1, \dots, k$, be the inclusion of R_i as a subset of $A_i \times B_i$, let $h_{\overline{A \times B}}$ be the unique morphism making the diagram

$$\begin{array}{ccccc} F\bar{A} & \xleftarrow{F\pi_1} & F(\bar{A} \times \bar{B}) & \xrightarrow{F\pi_2} & F\bar{B} & \xrightarrow{\alpha\bar{B}} & G\bar{B} \\ & \searrow \pi_1 & \downarrow h_{\overline{A \times B}} & \nearrow \pi_2 & & & \\ & & F\bar{A} \times G\bar{B} & & & & \end{array}$$

commute, and let $h_{\overline{R}} : F\overline{R} \rightarrow F\overline{A} \times G\overline{B}$ be $h_{\overline{A \times B}} \circ F\overline{\iota_R}$. Further, let $\alpha^\wedge \overline{R}$ be the subobject through which $h_{\overline{R}}$ is factorized by the mono-epi factorization system in **Set**, as shown in the following diagram:

$$\begin{array}{ccc} F\overline{R} & \xrightarrow{h_{\overline{R}}} & F\overline{A} \times G\overline{B} \\ & \searrow q_{\alpha^\wedge \overline{R}} & \nearrow \iota_{\alpha^\wedge \overline{R}} \\ & \alpha^\wedge \overline{R} & \end{array}$$

Then $\alpha^\wedge \overline{R} : \text{Rel}(F\overline{A}, G\overline{B})$ by construction, so the action of $\langle \alpha \rangle^*$ on objects can be given by $\langle \alpha \rangle^*(A, B, R) = (F\overline{A}, G\overline{B}, \iota_{\alpha^\wedge \overline{R}} \alpha^\wedge \overline{R})$. Its action on morphisms is given by $\langle \alpha \rangle^*(\beta, \beta') = (F\overline{\beta}, G\overline{\beta}')$.

The data in Definition 4.1 yield the *graph relation transformer* for α , denoted $\langle \alpha \rangle = (F, G, \langle \alpha \rangle^*)$.

Lemma 4.2. *If $F, G : [\text{Set}^k, \text{Set}]$, and if $\alpha : F \rightarrow G$ is a natural transformation, then $\langle \alpha \rangle$ is in RT_k .*

Proof. Clearly, $\langle \alpha \rangle^*$ is ω -cocontinuous, so $\langle \alpha \rangle^* : [\text{Rel}^k, \text{Rel}]$. Let $\overline{R} : \text{Rel}(A, B)$, $\overline{S} : \text{Rel}(C, D)$, and $(\beta, \beta') : R \rightarrow \overline{S}$. We want to show that there exists a morphism $\epsilon : \alpha^\wedge \overline{R} \rightarrow \alpha^\wedge \overline{S}$ such that the diagram on the left below commutes. Since $(\beta, \beta') : R \rightarrow \overline{S}$, there exist $\gamma : R \rightarrow \overline{S}$ such that each diagram in the middle commutes. Moreover, since both $h_{\overline{C \times D}} \circ F(\beta \times \beta')$ and $(F\overline{\beta} \times G\overline{\beta}') \circ h_{\overline{A \times B}}$ make the diagram on the right commute, they must be equal.

$$\begin{array}{ccc} \alpha^\wedge \overline{R} \xrightarrow{\iota_{\alpha^\wedge \overline{R}}} F\overline{A} \times G\overline{B} & R_i \xrightarrow{\iota_{R_i}} A_i \times B_i & F\overline{C} \xleftarrow{\pi_1} F\overline{C} \times G\overline{D} \xrightarrow{\pi_2} G\overline{D} \\ \epsilon \downarrow & \downarrow \gamma_i & \downarrow F\pi_1 \circ F(\beta \times \beta') \\ \alpha^\wedge \overline{S} \xrightarrow{\iota_{\alpha^\wedge \overline{S}}} F\overline{C} \times G\overline{D} & S_i \xrightarrow{\iota_{S_i}} C_i \times D_i & \downarrow \exists! \\ & & F(A \times B) \end{array}$$

We therefore get that the right-hand square in the diagram on the left below commutes, and thus that the entire diagram does as well. Finally, by the left-lifting property of $q_{F\wedge \overline{R}}$ with respect to $\iota_{F\wedge \overline{S}}$ given by the mono-epi factorization system, there exists an ϵ such that the diagram on the right below commutes as desired.

$$\begin{array}{ccc} & \xrightarrow{h_{\overline{R}}} & \\ F\overline{R} \xrightarrow{F\overline{\iota_R}} F(\overline{A \times B}) \xrightarrow{h_{\overline{A \times B}}} F\overline{A} \times G\overline{B} & & F\overline{R} \xrightarrow{q_{\alpha^\wedge \overline{R}}} \alpha^\wedge \overline{R} \xrightarrow{\iota_{\alpha^\wedge \overline{R}}} F\overline{A} \times G\overline{B} \\ F\overline{\gamma} \downarrow & \downarrow F(\beta \times \beta') & \downarrow F\overline{\beta} \times G\overline{\beta}' \\ F\overline{S} \xrightarrow{F\overline{\iota_S}} F(\overline{C \times D}) \xrightarrow{h_{\overline{C \times D}}} F\overline{C} \times G\overline{D} & & F\overline{S} \xrightarrow{q_{\alpha^\wedge \overline{S}}} \alpha^\wedge \overline{S} \xrightarrow{\iota_{\alpha^\wedge \overline{S}}} F\overline{C} \times G\overline{D} \\ & \xrightarrow{h_{\overline{S}}} & \end{array}$$

□

If $f : A \rightarrow B$ is a morphism in **Set** then the definition of the graph relation transformer $\langle f \rangle$ for f as a natural transformation between 0-ary functors A and B coincides with its standard definition. Graph relation transformers are thus a reasonable extension of graph relations to functors.

The action of a graph relation transformer on a graph relation can be computed explicitly:

Lemma 4.3. *If $\alpha : F \rightarrow G$ is a morphism in $[\text{Set}^k, \text{Set}]$ and $f_1 : A_1 \rightarrow B_1, \dots, f_k : A_k \rightarrow B_k$, then $\langle \alpha \rangle^*(\langle f \rangle) = \langle Gf \circ \alpha_{\overline{A}} \rangle = \langle \alpha_{\overline{B}} \circ Ff \rangle$.*

Proof. Since $h_{\overline{A \times B}}$ is the unique morphism making the bottom triangle of the diagram on the left below commute, and since $h_{\langle \overline{f} \rangle} = h_{\overline{A \times B}} \circ F\overline{\iota_{\langle f \rangle}} = h_{\overline{A \times B}} \circ F\langle id_A, f \rangle$, the universal property of the product depicted in the diagram on the right gives $h_{\langle \overline{f} \rangle} = \langle id_{F\overline{A}}, \alpha_{\overline{B}} \circ Ff \rangle : F\overline{A} \rightarrow F\overline{A} \times G\overline{B}$.

$$\begin{array}{ccc}
& F\bar{A} & \\
& \swarrow \quad \searrow & \\
F\bar{A} & \xleftarrow{F\pi_1} F(\bar{A} \times \bar{B}) \xrightarrow{F\pi_2} F\bar{B} & \xrightarrow{\alpha_{\bar{B}}} G\bar{B} \\
& \swarrow \quad \searrow & \\
& F\bar{A} \times G\bar{B} &
\end{array}
\quad
\begin{array}{ccc}
F\bar{A} & \xleftarrow{\pi_1} F\bar{A} \times G\bar{B} \xrightarrow{\pi_2} G\bar{B} & \\
& \uparrow \exists! & \uparrow \alpha_{\bar{B}} \\
F\bar{A} & \xrightarrow{F\bar{f}} F\bar{B} &
\end{array}$$

Moreover, $\langle id_{F\bar{A}}, \alpha_{\bar{B}} \circ F\bar{f} \rangle$ is a monomorphism in **Set** because $id_{F\bar{A}}$ is, so its mono-epi factorization gives $\iota_{\alpha \wedge \langle \bar{f} \rangle} = \langle id_{F\bar{A}}, \alpha_{\bar{B}} \circ F\bar{f} \rangle$, and thus $\alpha \wedge \langle \bar{f} \rangle = F\bar{A}$. Then $\iota_{\alpha \wedge \langle \bar{f} \rangle} \alpha \wedge \langle \bar{f} \rangle = \langle id_{F\bar{A}}, \alpha_{\bar{B}} \circ F\bar{f} \rangle(F\bar{A}) = \langle \alpha_{\bar{B}} \circ F\bar{f} \rangle^*$, so that

$$\langle \alpha \rangle^* \langle \bar{f} \rangle = (F\bar{A}, G\bar{B}, \iota_{\alpha \wedge \langle \bar{f} \rangle} \alpha \wedge \langle \bar{f} \rangle) = (F\bar{A}, G\bar{B}, \langle \alpha_{\bar{B}} \circ F\bar{f} \rangle^*) = \langle \alpha_{\bar{B}} \circ F\bar{f} \rangle$$

Finally, $\alpha_{\bar{B}} \circ F\bar{f} = G\bar{f} \circ \alpha_{\bar{A}}$ by naturality of α . \square

To prove the IEL, we also need to know that the equality relation transformer preserves equality relations. The *equality relation transformer* on $F : [\mathbf{Set}^k, \mathbf{Set}]$ is defined to be

$$\mathbf{Eq}_F = \langle id_F \rangle = (F, F, \langle id_F \rangle^*) \quad (4.1)$$

Lemma 4.3 then gives that, for all $\bar{A} : \mathbf{Set}$,

$$\mathbf{Eq}_F^* \mathbf{Eq}_{\bar{A}} = \langle id_F \rangle^* \langle id_{\bar{A}} \rangle = \langle F id_{\bar{A}} \circ (id_F)_{\bar{A}} \rangle = \langle id_{F\bar{A}} \circ id_{F\bar{A}} \rangle = \langle id_{F\bar{A}} \rangle = \mathbf{Eq}_{F\bar{A}} \quad (4.2)$$

Graph relation transformers in general, and equality relation transformers in particular, naturally extend to relation environments. Indeed, if $\rho, \rho' : \mathbf{SetEnv}$ and $f : \rho \rightarrow \rho'$, then the *graph relation environment* $\langle f \rangle$ is defined pointwise by $\langle f \rangle \phi = \langle f \phi \rangle$ for every ϕ , which entails that $\pi_1 \langle f \rangle = \rho$ and $\pi_2 \langle f \rangle = \rho'$. In particular, the *equality relation environment* \mathbf{Eq}_ρ is defined to be $\langle id_\rho \rangle$, which entails that $\mathbf{Eq}_\rho \phi = \mathbf{Eq}_{\rho \phi}$ for every ϕ .

With these definitions in hand, we can state and prove both an Identity Extension Lemma and a Graph Lemma for our calculus.

Theorem 4.4 (IEL). *If $\rho : \mathbf{SetEnv}$ and $\Gamma; \Phi \vdash F$ then $\llbracket \Gamma; \Phi \vdash F \rrbracket^{\mathbf{Rel}} \mathbf{Eq}_\rho = \mathbf{Eq}_{\llbracket \Gamma; \Phi \vdash F \rrbracket^{\mathbf{Set}} \rho}$.*

Proof. The proof is by induction on the structure of F . Only the **Nat**, application, and fixpoint cases are non-routine.

- $\llbracket \Gamma; \Phi \vdash 0 \rrbracket^{\mathbf{Rel}} \mathbf{Eq}_\rho = 0_{\mathbf{Rel}} = \mathbf{Eq}_{0_{\mathbf{Set}}} = \mathbf{Eq}_{\llbracket \Gamma; \Phi \vdash 0 \rrbracket^{\mathbf{Set}} \rho}$
- $\llbracket \Gamma; \Phi \vdash 1 \rrbracket^{\mathbf{Rel}} \mathbf{Eq}_\rho = 1_{\mathbf{Rel}} = \mathbf{Eq}_{1_{\mathbf{Set}}} = \mathbf{Eq}_{\llbracket \Gamma; \Phi \vdash 1 \rrbracket^{\mathbf{Set}} \rho}$
- By definition, $\llbracket \Gamma; \emptyset \vdash \mathbf{Nat}^{\bar{\alpha}} F G \rrbracket^{\mathbf{Rel}} \mathbf{Eq}_\rho$ is the relation on $\llbracket \Gamma; \emptyset \vdash \mathbf{Nat}^{\bar{\alpha}} F G \rrbracket^{\mathbf{Set}} \rho$ relating t and t' if, for all $R_1 : \mathbf{Rel}(A_1, B_1), \dots, R_k : \mathbf{Rel}(A_k, B_k)$, $(t_{\bar{A}}, t'_{\bar{B}})$ is a morphism from $\llbracket \Gamma; \bar{\alpha} \vdash F \rrbracket^{\mathbf{Rel}} \mathbf{Eq}_\rho[\bar{\alpha} := \bar{R}]$ to $\llbracket \Gamma; \bar{\alpha} \vdash G \rrbracket^{\mathbf{Rel}} \mathbf{Eq}_\rho[\bar{\alpha} := \bar{R}]$ in **Rel**. To prove that this relation is $\mathbf{Eq}_{\llbracket \Gamma; \emptyset \vdash \mathbf{Nat}^{\bar{\alpha}} F G \rrbracket^{\mathbf{Set}} \rho}$ we need to show that $(t_{\bar{A}}, t'_{\bar{B}})$ is a morphism from $\llbracket \Gamma; \bar{\alpha} \vdash F \rrbracket^{\mathbf{Rel}} \mathbf{Eq}_\rho[\bar{\alpha} := \bar{R}]$ to $\llbracket \Gamma; \bar{\alpha} \vdash G \rrbracket^{\mathbf{Rel}} \mathbf{Eq}_\rho[\bar{\alpha} := \bar{R}]$ in **Rel** for all $R_1 : \mathbf{Rel}(A_1, B_1), \dots, R_k : \mathbf{Rel}(A_k, B_k)$ if and only if $t = t'$. We first show that if $(t_{\bar{A}}, t'_{\bar{B}})$ is a morphism from $\llbracket \Gamma; \bar{\alpha} \vdash F \rrbracket^{\mathbf{Rel}} \mathbf{Eq}_\rho[\bar{\alpha} := \bar{R}]$ to $\llbracket \Gamma; \bar{\alpha} \vdash G \rrbracket^{\mathbf{Rel}} \mathbf{Eq}_\rho[\bar{\alpha} := \bar{R}]$ for all $R_1 : \mathbf{Rel}(A_1, B_1), \dots, R_k : \mathbf{Rel}(A_k, B_k)$ then $t = t'$. By hypothesis, for all $A_1 \dots A_k : \mathbf{Set}$, $(t_{\bar{A}}, t'_{\bar{A}})$ is a morphism from $\llbracket \Gamma; \bar{\alpha} \vdash F \rrbracket^{\mathbf{Rel}} \mathbf{Eq}_\rho[\bar{\alpha} := \mathbf{Eq}_{\bar{A}}]$ to $\llbracket \Gamma; \bar{\alpha} \vdash G \rrbracket^{\mathbf{Rel}} \mathbf{Eq}_\rho[\bar{\alpha} := \mathbf{Eq}_{\bar{A}}]$. By the induction hypothesis, it is therefore a morphism from $\mathbf{Eq}_{\llbracket \Gamma; \bar{\alpha} \vdash F \rrbracket^{\mathbf{Set}} \rho[\bar{\alpha} := \bar{A}]}$ to $\mathbf{Eq}_{\llbracket \Gamma; \bar{\alpha} \vdash G \rrbracket^{\mathbf{Set}} \rho[\bar{\alpha} := \bar{A}]}$ in **Rel**. This means that, for all $x : \mathbf{Eq}_{\llbracket \Gamma; \bar{\alpha} \vdash F \rrbracket^{\mathbf{Set}} \rho[\bar{\alpha} := \bar{A}]}$, $t_{\bar{A}} x = t'_{\bar{A}} x$, so $t = t'$ by extensionality. For the converse, we observe that if $t = t'$ then the desired conclusion is simply the extra condition in the definition of $\llbracket \Gamma; \emptyset \vdash \mathbf{Nat}^{\bar{\alpha}} F G \rrbracket^{\mathbf{Set}} \rho$.

- The application case is proved by the following sequence of equalities, where the second equality is by the induction hypothesis and the definition of the relation environment Eq_ρ , the third is by the definition of application of relation transformers from Definition 9, and the fourth is by Equation 4.2:

$$\begin{aligned}
\llbracket \Gamma; \Phi \vdash \phi \bar{F} \rrbracket^{\text{Rel}} \text{Eq}_\rho &= (\text{Eq}_\rho \phi) \overline{\llbracket \Gamma; \Phi \vdash F \rrbracket^{\text{Rel}} \text{Eq}_\rho} \\
&= \text{Eq}_{\rho \phi} \overline{\text{Eq}_{\llbracket \Gamma; \Phi \vdash F \rrbracket^{\text{Set}}_\rho}} \\
&= (\text{Eq}_{\rho \phi})^* \overline{\text{Eq}_{\llbracket \Gamma; \Phi \vdash F \rrbracket^{\text{Set}}_\rho}} \\
&= \text{Eq}_{(\rho \phi)} \overline{\llbracket \Gamma; \Phi \vdash F \rrbracket^{\text{Set}}_\rho} \\
&= \text{Eq}_{\llbracket \Gamma; \Phi \vdash \phi \bar{F} \rrbracket^{\text{Set}}_\rho}
\end{aligned}$$

- The fixpoint case is proven by the sequence of equalities

$$\begin{aligned}
\llbracket \Gamma; \Phi \vdash (\mu \phi. \lambda \bar{\alpha}. H) \bar{F} \rrbracket^{\text{Rel}} \text{Eq}_\rho &= (\mu T_{H, \text{Eq}_\rho}) \overline{\llbracket \Gamma; \Phi \vdash F \rrbracket^{\text{Rel}} \text{Eq}_\rho} \\
&= \lim_{\rightarrow n \in \mathbb{N}} T_{H, \text{Eq}_\rho}^n K_0 \overline{\llbracket \Gamma; \Phi \vdash F \rrbracket^{\text{Rel}} \text{Eq}_\rho} \\
&= \lim_{\rightarrow n \in \mathbb{N}} T_{H, \text{Eq}_\rho}^n K_0 \overline{\text{Eq}_{\llbracket \Gamma; \Phi \vdash F \rrbracket^{\text{Set}}_\rho}} \\
&= \lim_{\rightarrow n \in \mathbb{N}} (\text{Eq}_{(T_{H, \rho}^{\text{Set}})^n K_0})^* \overline{\text{Eq}_{\llbracket \Gamma; \Phi \vdash F \rrbracket^{\text{Set}}_\rho}} \\
&= \lim_{\rightarrow n \in \mathbb{N}} \text{Eq}_{(T_{H, \rho}^{\text{Set}})^n K_0} \overline{\llbracket \Gamma; \Phi \vdash F \rrbracket^{\text{Set}}_\rho} \\
&= \text{Eq}_{\lim_{\rightarrow n \in \mathbb{N}} (T_{H, \rho}^{\text{Set}})^n K_0} \overline{\llbracket \Gamma; \Phi \vdash F \rrbracket^{\text{Set}}_\rho} \\
&= \text{Eq}_{\llbracket \Gamma; \Phi \vdash (\mu \phi. \lambda \bar{\alpha}. H) \bar{F} \rrbracket^{\text{Set}}_\rho}
\end{aligned}$$

Here, the third equality is by induction hypothesis, the fifth is by Equation 4.2, and the fourth equality is because, for every $n \in \mathbb{N}$, the following two statements can be proved by simultaneous induction: for any H , ρ , and A ,

$$T_{H, \text{Eq}_\rho}^n K_0 \overline{\text{Eq}_A} = (\text{Eq}_{(T_{H, \rho}^{\text{Set}})^n K_0})^* \overline{\text{Eq}_A} \quad (4.3)$$

and for any subformula J of H ,

$$\begin{aligned}
&\llbracket \Gamma; \phi, \bar{\alpha} \vdash J \rrbracket^{\text{Rel}} \text{Eq}_\rho [\phi := T_{H, \text{Eq}_\rho}^n K_0] [\bar{\alpha} := \overline{\text{Eq}_A}] \\
&= \llbracket \Gamma; \phi, \bar{\alpha} \vdash J \rrbracket^{\text{Rel}} \text{Eq}_\rho [\phi := \text{Eq}_{(T_{H, \rho}^{\text{Set}})^n K_0}] [\bar{\alpha} := \overline{\text{Eq}_A}]
\end{aligned} \quad (4.4)$$

We prove Equation 4.3 by induction on n . The case $n = 0$ is trivial, because $T_{H, \text{Eq}_\rho}^0 K_0 = K_0$ and $(T_{H, \rho}^{\text{Set}})^0 K_0 = K_0$; the inductive step is proved by the following sequence of equalities:

$$\begin{aligned}
T_{H, \text{Eq}_\rho}^{n+1} K_0 \overline{\text{Eq}_A} &= T_{H, \text{Eq}_\rho}^{\text{Rel}} (T_{H, \text{Eq}_\rho}^n K_0) \overline{\text{Eq}_A} \\
&= \llbracket \Gamma; \phi, \bar{\alpha} \vdash H \rrbracket^{\text{Rel}} \text{Eq}_\rho [\phi := T_{H, \text{Eq}_\rho}^n K_0] [\bar{\alpha} := \overline{\text{Eq}_A}] \\
&= \llbracket \Gamma; \phi, \bar{\alpha} \vdash H \rrbracket^{\text{Rel}} \text{Eq}_\rho [\phi := \text{Eq}_{(T_{H, \rho}^{\text{Set}})^n K_0}] [\bar{\alpha} := \overline{\text{Eq}_A}] \\
&= \llbracket \Gamma; \phi, \bar{\alpha} \vdash H \rrbracket^{\text{Rel}} \text{Eq}_{\rho[\phi := (T_{H, \rho}^{\text{Set}})^n K_0]} [\bar{\alpha} := \overline{A}] \\
&= \text{Eq}_{\llbracket \Gamma; \phi, \bar{\alpha} \vdash H \rrbracket^{\text{Set}}_\rho [\phi := (T_{H, \rho}^{\text{Set}})^n K_0]} [\bar{\alpha} := \overline{A}] \\
&= \text{Eq}_{(T_{H, \rho}^{\text{Set}})^{n+1} K_0} \overline{A} \\
&= (\text{Eq}_{(T_{H, \rho}^{\text{Set}})^{n+1} K_0})^* \overline{\text{Eq}_A}
\end{aligned}$$

Here, the third equality is by Equation 4.4 for $J = H$, the fifth by the induction hypothesis of the IEL on H , and the last is by Equation 4.2.

We prove Equation 4.4 by structural induction on J . The only interesting cases, though, are when $J = \phi \bar{G}$ and when $J = (\mu \psi. \lambda \bar{\beta}. G) \bar{K}$.

- The case $J = \phi\overline{G}$ is proved by the sequence of equalities:

$$\begin{aligned}
& \llbracket \Gamma; \phi, \bar{\alpha} \vdash \phi\overline{G} \rrbracket^{\text{Rel}} \text{Eq}_\rho[\phi := T_{H, \text{Eq}_\rho}^n K_0][\bar{\alpha} := \overline{\text{Eq}_A}] \\
&= T_{H, \text{Eq}_\rho}^n K_0 \overline{\llbracket \Gamma; \phi, \bar{\alpha} \vdash G \rrbracket^{\text{Rel}} \text{Eq}_\rho[\phi := T_{H, \text{Eq}_\rho}^n K_0][\bar{\alpha} := \overline{\text{Eq}_A}]} \\
&= T_{H, \text{Eq}_\rho}^n K_0 \overline{\llbracket \Gamma; \phi, \bar{\alpha} \vdash G \rrbracket^{\text{Rel}} \text{Eq}_\rho[\phi := \text{Eq}_{(T_{H, \rho}^{\text{Set}})^n K_0}][\bar{\alpha} := \overline{\text{Eq}_A}]} \\
&= T_{H, \text{Eq}_\rho}^n K_0 \overline{\llbracket \Gamma; \phi, \bar{\alpha} \vdash G \rrbracket^{\text{Rel}} \text{Eq}_\rho[\phi := (T_{H, \rho}^{\text{Set}})^n K_0][\bar{\alpha} := \overline{A}]} \\
&= T_{H, \text{Eq}_\rho}^n K_0 \overline{\text{Eq}_{\llbracket \Gamma; \phi, \bar{\alpha} \vdash G \rrbracket^{\text{Set}} \rho[\phi := (T_{H, \rho}^{\text{Set}})^n K_0][\bar{\alpha} := \overline{A}]}} \\
&= (\text{Eq}_{(T_{H, \rho}^{\text{Set}})^n K_0})^* \overline{\text{Eq}_{\llbracket \Gamma; \phi, \bar{\alpha} \vdash G \rrbracket^{\text{Set}} \rho[\phi := (T_{H, \rho}^{\text{Set}})^n K_0][\bar{\alpha} := \overline{A}]}} \\
&= (\text{Eq}_{(T_{H, \rho}^{\text{Set}})^n K_0})^* \overline{\llbracket \Gamma; \phi, \bar{\alpha} \vdash G \rrbracket^{\text{Rel}} \text{Eq}_\rho[\phi := \text{Eq}_{(T_{H, \rho}^{\text{Set}})^n K_0}][\bar{\alpha} := \overline{\text{Eq}_A}]} \\
&= \llbracket \Gamma; \phi, \bar{\alpha} \vdash \phi\overline{G} \rrbracket^{\text{Rel}} \text{Eq}_\rho[\phi := \text{Eq}_{(T_{H, \rho}^{\text{Set}})^n K_0}][\bar{\alpha} := \overline{\text{Eq}_A}]
\end{aligned}$$

Here, the second equality is by the induction hypothesis for Equation 4.4 on the G s, the fourth is by the induction hypothesis for the IEL on the G s, and the fifth is by the induction hypothesis on n for Equation 4.3.

- The case $J = (\mu\psi. \lambda\beta. G)\overline{K}$ is proved by the sequence of equalities

$$\begin{aligned}
& \llbracket \Gamma; \phi, \bar{\alpha} \vdash (\mu\psi. \lambda\beta. G)\overline{K} \rrbracket^{\text{Rel}} \text{Eq}_\rho[\phi := T_{H, \text{Eq}_\rho}^n K_0][\bar{\alpha} := \overline{\text{Eq}_A}] \\
&= (\mu T_{G, \text{Eq}_\rho[\phi := T_{H, \text{Eq}_\rho}^n K_0][\bar{\alpha} := \overline{\text{Eq}_A}]})(\llbracket \Gamma; \phi, \bar{\alpha} \vdash K \rrbracket^{\text{Rel}} \text{Eq}_\rho[\phi := T_{H, \text{Eq}_\rho}^n K_0][\bar{\alpha} := \overline{\text{Eq}_A}]) \\
&= \lim_{\rightarrow m \in \mathbb{N}} T_{G, \text{Eq}_\rho[\phi := T_{H, \text{Eq}_\rho}^n K_0][\bar{\alpha} := \overline{\text{Eq}_A}]}^m K_0 (\llbracket \Gamma; \phi, \bar{\alpha} \vdash K \rrbracket^{\text{Rel}} \text{Eq}_\rho[\phi := T_{H, \text{Eq}_\rho}^n K_0][\bar{\alpha} := \overline{\text{Eq}_A}]) \\
&= \lim_{\rightarrow m \in \mathbb{N}} T_{G, \text{Eq}_\rho[\phi := T_{H, \text{Eq}_\rho}^n K_0][\bar{\alpha} := \overline{\text{Eq}_A}]}^m K_0 (\llbracket \Gamma; \phi, \bar{\alpha} \vdash K \rrbracket^{\text{Rel}} \text{Eq}_\rho[\phi := \text{Eq}_{(T_{H, \rho}^{\text{Set}})^n K_0}][\bar{\alpha} := \overline{\text{Eq}_A}]) \\
&= \lim_{\rightarrow m \in \mathbb{N}} T_{G, \text{Eq}_\rho[\phi := \text{Eq}_{(T_{H, \rho}^{\text{Set}})^n K_0}][\bar{\alpha} := \overline{\text{Eq}_A}]}^m K_0 (\llbracket \Gamma; \phi, \bar{\alpha} \vdash K \rrbracket^{\text{Rel}} \text{Eq}_\rho[\phi := \text{Eq}_{(T_{H, \rho}^{\text{Set}})^n K_0}][\bar{\alpha} := \overline{\text{Eq}_A}]) \\
&= (\mu T_{G, \text{Eq}_\rho[\phi := \text{Eq}_{(T_{H, \rho}^{\text{Set}})^n K_0}][\bar{\alpha} := \overline{\text{Eq}_A}]})(\llbracket \Gamma; \phi, \bar{\alpha} \vdash K \rrbracket^{\text{Rel}} \text{Eq}_\rho[\phi := \text{Eq}_{(T_{H, \rho}^{\text{Set}})^n K_0}][\bar{\alpha} := \overline{\text{Eq}_A}]) \\
&= \llbracket \Gamma; \phi, \bar{\alpha} \vdash (\mu\psi. \lambda\beta. G)\overline{K} \rrbracket^{\text{Rel}} \text{Eq}_\rho[\phi := \text{Eq}_{(T_{H, \rho}^{\text{Set}})^n K_0}][\bar{\alpha} := \overline{\text{Eq}_A}]
\end{aligned}$$

Here, the third equality is by the induction hypothesis for Equation 4.4 on the K s, and the fourth equality holds because we can prove that, for all $m \in \mathbb{N}$,

$$T_{G, \text{Eq}_\rho[\phi := T_{H, \text{Eq}_\rho}^n K_0][\bar{\alpha} := \overline{\text{Eq}_A}]}^m K_0 = T_{G, \text{Eq}_\rho[\phi := \text{Eq}_{(T_{H, \rho}^{\text{Set}})^n K_0}][\bar{\alpha} := \overline{\text{Eq}_A}]}^m K_0 \quad (4.5)$$

Indeed, the base case of Equation 4.5 is trivial because

$$T_{G, \text{Eq}_\rho[\phi := T_{H, \text{Eq}_\rho}^n K_0][\bar{\alpha} := \overline{\text{Eq}_A}]}^0 K_0 = K_0 = T_{G, \text{Eq}_\rho[\phi := \text{Eq}_{(T_{H, \rho}^{\text{Set}})^n K_0}][\bar{\alpha} := \overline{\text{Eq}_A}]}^0 K_0$$

and the inductive case is proved by:

$$\begin{aligned}
& T_{G, \text{Eq}_\rho[\phi := T_{H, \text{Eq}_\rho}^n K_0][\bar{\alpha} := \overline{\text{Eq}_A}]}^{m+1} K_0 \\
&= T_{G, \text{Eq}_\rho[\phi := T_{H, \text{Eq}_\rho}^n K_0][\bar{\alpha} := \overline{\text{Eq}_A}]} (T_{G, \text{Eq}_\rho[\phi := T_{H, \text{Eq}_\rho}^n K_0][\bar{\alpha} := \overline{\text{Eq}_A}]}^m K_0) \\
&= T_{G, \text{Eq}_\rho[\phi := T_{H, \text{Eq}_\rho}^n K_0][\bar{\alpha} := \overline{\text{Eq}_A}]} (T_{G, \text{Eq}_\rho[\phi := \text{Eq}_{(T_{H, \rho}^{\text{Set}})^n K_0}][\bar{\alpha} := \overline{\text{Eq}_A}]}^m K_0) \\
&= \lambda \bar{R}. [\Gamma; \psi, \bar{\beta} \vdash G]^{\text{Rel}} \text{Eq}_\rho[\phi := T_{H, \text{Eq}_\rho}^n K_0][\bar{\alpha} := \overline{\text{Eq}_A}][\psi := T_{G, \text{Eq}_\rho[\phi := \text{Eq}_{(T_{H, \rho}^{\text{Set}})^n K_0}][\bar{\alpha} := \overline{\text{Eq}_A}]}^m K_0][\bar{\beta} := \bar{R}] \\
&= \lambda \bar{R}. [\Gamma; \psi, \bar{\beta} \vdash G]^{\text{Rel}} \text{Eq}_\rho[\phi := \text{Eq}_{(T_{H, \rho}^{\text{Set}})^n K_0}][\bar{\alpha} := \overline{\text{Eq}_A}][\psi := T_{G, \text{Eq}_\rho[\phi := \text{Eq}_{(T_{H, \rho}^{\text{Set}})^n K_0}][\bar{\alpha} := \overline{\text{Eq}_A}]}^m K_0][\bar{\beta} := \bar{R}] \\
&= T_{G, \text{Eq}_\rho[\phi := \text{Eq}_{(T_{H, \rho}^{\text{Set}})^n K_0}][\bar{\alpha} := \overline{\text{Eq}_A}]} (T_{G, \text{Eq}_\rho[\phi := \text{Eq}_{(T_{H, \rho}^{\text{Set}})^n K_0}][\bar{\alpha} := \overline{\text{Eq}_A}]}^m K_0) \\
&= T_{G, \text{Eq}_\rho[\phi := \text{Eq}_{(T_{H, \rho}^{\text{Set}})^n K_0}][\bar{\alpha} := \overline{\text{Eq}_A}]}^{m+1} K_0
\end{aligned}$$

Here, the second equality holds by the induction hypothesis for Equation 4.5 on m . The fourth equality holds because ϕ and the variables in $\bar{\alpha}$ do not appear in G . (We note that this case of the IEL would fail to hold if the functorial context in which the body F of a μ -type $\mu\bar{\alpha}.F$ were formed in Definition 2.1 were extended to contain variables other than ϕ and the α s. This justifies our design of the μ -type formation rule there.)

This concludes the fixpoint case.

- $[\Gamma; \Phi \vdash F + G]^{\text{Rel}} \text{Eq}_\rho = [\Gamma; \Phi \vdash F]^{\text{Rel}} \text{Eq}_\rho + [\Gamma; \Phi \vdash G]^{\text{Rel}} \text{Eq}_\rho = \text{Eq}_{[\Gamma; \Phi \vdash F]^{\text{Set}}_\rho} + \text{Eq}_{[\Gamma; \Phi \vdash G]^{\text{Set}}_\rho} = \text{Eq}_{[\Gamma; \Phi \vdash F]^{\text{Set}}_\rho + [\Gamma; \Phi \vdash G]^{\text{Set}}_\rho} = \text{Eq}_{[\Gamma; \Phi \vdash F + G]^{\text{Set}}_\rho}$
- $[\Gamma; \Phi \vdash F \times G]^{\text{Rel}} \text{Eq}_\rho = [\Gamma; \Phi \vdash F]^{\text{Rel}} \text{Eq}_\rho \times [\Gamma; \Phi \vdash G]^{\text{Rel}} \text{Eq}_\rho = \text{Eq}_{[\Gamma; \Phi \vdash F]^{\text{Set}}_\rho} \times \text{Eq}_{[\Gamma; \Phi \vdash G]^{\text{Set}}_\rho} = \text{Eq}_{[\Gamma; \Phi \vdash F]^{\text{Set}}_\rho \times [\Gamma; \Phi \vdash G]^{\text{Set}}_\rho} = \text{Eq}_{[\Gamma; \Phi \vdash F \times G]^{\text{Set}}_\rho}$

□

With the IEL in hand we can prove a Graph Lemma appropriate to our setting:

Lemma 4.5 (Graph Lemma). *If $\rho, \rho' : \text{SetEnv}$ and $f : \rho \rightarrow \rho'$ then $\langle [\Gamma; \Phi \vdash F]^{\text{Set}} f \rangle = [\Gamma; \Phi \vdash F]^{\text{Rel}} \langle f \rangle$.*

Proof. Applying Lemma 3.13 to the morphisms $(f, id_{\rho'}) : \langle f \rangle \rightarrow \text{Eq}_{\rho'}$ and $(id_\rho, f) : \text{Eq}_\rho \rightarrow \langle f \rangle$ of relation environments gives

$$\begin{aligned}
& ([\Gamma; \Phi \vdash F]^{\text{Set}} f, [\Gamma; \Phi \vdash F]^{\text{Set}} id_{\rho'}) \\
&= [\Gamma; \Phi \vdash F]^{\text{Rel}}(f, id_{\rho'}) : [\Gamma; \Phi \vdash F]^{\text{Rel}} \langle f \rangle \rightarrow [\Gamma; \Phi \vdash F]^{\text{Rel}} \text{Eq}_{\rho'}
\end{aligned}$$

and

$$\begin{aligned}
& ([\Gamma; \Phi \vdash F]^{\text{Set}} id_\rho, [\Gamma; \Phi \vdash F]^{\text{Set}} f) \\
&= [\Gamma; \Phi \vdash F]^{\text{Rel}}(id_\rho, f) : [\Gamma; \Phi \vdash F]^{\text{Rel}} \text{Eq}_\rho \rightarrow [\Gamma; \Phi \vdash F]^{\text{Rel}} \langle f \rangle
\end{aligned}$$

Expanding the first equation gives that if $(x, y) \in [\Gamma; \Phi \vdash F]^{\text{Rel}} \langle f \rangle$ then

$$([\Gamma; \Phi \vdash F]^{\text{Set}} f x, [\Gamma; \Phi \vdash F]^{\text{Set}} id_{\rho'} y) \in [\Gamma; \Phi \vdash F]^{\text{Rel}} \text{Eq}_{\rho'}$$

So $[\Gamma; \Phi \vdash F]^{\text{Set}} id_{\rho'} y = id_{[\Gamma; \Phi \vdash F]^{\text{Set}}_{\rho'}} y = y$ and $[\Gamma; \Phi \vdash F]^{\text{Rel}} \text{Eq}_{\rho'} = \text{Eq}_{[\Gamma; \Phi \vdash F]^{\text{Set}}_{\rho'}}$, and if $(x, y) \in [\Gamma; \Phi \vdash F]^{\text{Rel}} \langle f \rangle$ then $([\Gamma; \Phi \vdash F]^{\text{Set}} f x, y) \in \text{Eq}_{[\Gamma; \Phi \vdash F]^{\text{Set}}_{\rho'}}$, i.e., $[\Gamma; \Phi \vdash F]^{\text{Set}} f x = y$, i.e., $(x, y) \in \langle [\Gamma; \Phi \vdash F]^{\text{Set}} f \rangle$. So, we have $[\Gamma; \Phi \vdash F]^{\text{Rel}} \langle f \rangle \subseteq \langle [\Gamma; \Phi \vdash F]^{\text{Set}} f \rangle$.

Expanding the second equation gives that if $x \in [\Gamma; \Phi \vdash F]^{\text{Set}}_\rho$ then

$$([\Gamma; \Phi \vdash F]^{\text{Set}} id_\rho x, [\Gamma; \Phi \vdash F]^{\text{Set}} f x) \in [\Gamma; \Phi \vdash F]^{\text{Rel}} \langle f \rangle$$

Then $[\Gamma; \Phi \vdash F]^{\text{Set}} id_\rho x = id_{[\Gamma; \Phi \vdash F]^{\text{Set}}_\rho} x = x$, so for any $x \in [\Gamma; \Phi \vdash F]^{\text{Set}}_\rho$ we have that $(x, [\Gamma; \Phi \vdash F]^{\text{Set}} f x) \in [\Gamma; \Phi \vdash F]^{\text{Rel}} \langle f \rangle$. Moreover, $x \in [\Gamma; \Phi \vdash F]^{\text{Set}}_\rho$ if and only if $(x, [\Gamma; \Phi \vdash F]^{\text{Set}} f x) \in \langle [\Gamma; \Phi \vdash F]^{\text{Set}} f \rangle$ and, if $x \in [\Gamma; \Phi \vdash F]^{\text{Set}}_\rho$ then $(x, [\Gamma; \Phi \vdash F]^{\text{Set}} f x) \in [\Gamma; \Phi \vdash F]^{\text{Rel}} \langle f \rangle$. Thus, if $(x, [\Gamma; \Phi \vdash F]^{\text{Set}} f x) \in$

$\llbracket \Gamma; \Phi \mid \Delta, x : F \vdash x : F \rrbracket^{\text{D}} \rho$	$= \pi_{ \Delta +1}$
$\llbracket \Gamma; \emptyset \mid \Delta \vdash L_{\bar{\alpha}} x.t : \text{Nat}^{\bar{\alpha}} F G \rrbracket^{\text{D}} \rho$	$= \text{curry}(\llbracket \Gamma; \bar{\alpha} \mid \Delta, x : F \vdash t : G \rrbracket^{\text{D}} \rho [\bar{\alpha} := _])$
$\llbracket \Gamma; \Phi \mid \Delta \vdash t_{\bar{K}} s : G[\bar{\alpha} := \bar{K}] \rrbracket^{\text{D}} \rho$	$= \text{eval} \circ \langle \lambda d. (\llbracket \Gamma; \emptyset \mid \Delta \vdash t : \text{Nat}^{\bar{\alpha}} F G \rrbracket^{\text{D}} \rho d) \rrbracket^{\text{D}} \rho, \\ \llbracket \Gamma; \Phi \mid \Delta \vdash s : F[\bar{\alpha} := \bar{K}] \rrbracket^{\text{D}} \rho \rangle$
$\llbracket \Gamma; \Phi \mid \Delta \vdash \perp_F t : F \rrbracket^{\text{D}} \rho$	$= !_{\llbracket \Gamma; \Phi \vdash F \rrbracket^{\text{D}} \rho}^0 \circ \llbracket \Gamma; \Phi \mid \Delta \vdash t : \emptyset \rrbracket^{\text{D}} \rho$, where $!_{\llbracket \Gamma; \Phi \vdash F \rrbracket^{\text{D}} \rho}^0$ is the unique morphism from 0 to $\llbracket \Gamma; \Phi \vdash F \rrbracket^{\text{D}} \rho$
$\llbracket \Gamma; \Phi \mid \Delta \vdash \top : 1 \rrbracket^{\text{D}} \rho$	$= !_1^{\llbracket \Gamma; \Phi \vdash \Delta \rrbracket^{\text{D}} \rho}$, where $!_1^{\llbracket \Gamma; \Phi \vdash \Delta \rrbracket^{\text{D}} \rho}$ is the unique morphism from $\llbracket \Gamma; \Phi \vdash \Delta \rrbracket^{\text{D}} \rho$ to 1
$\llbracket \Gamma; \Phi \mid \Delta \vdash (s, t) : F \times G \rrbracket^{\text{D}} \rho$	$= \langle \llbracket \Gamma; \Phi \mid \Delta \vdash s : F \rrbracket^{\text{D}} \rho, \llbracket \Gamma; \Phi \mid \Delta \vdash t : G \rrbracket^{\text{D}} \rho \rangle$
$\llbracket \Gamma; \Phi \mid \Delta \vdash \pi_1 t : F \rrbracket^{\text{D}} \rho$	$= \pi_1 \circ \llbracket \Gamma; \Phi \mid \Delta \vdash t : F \times G \rrbracket^{\text{D}} \rho$
$\llbracket \Gamma; \Phi \mid \Delta \vdash \pi_2 t : G \rrbracket^{\text{D}} \rho$	$= \pi_2 \circ \llbracket \Gamma; \Phi \mid \Delta \vdash t : F \times G \rrbracket^{\text{D}} \rho$
$\llbracket \Gamma; \Phi \mid \Delta \vdash \text{case } t \text{ of } \{x \mapsto l; y \mapsto r\} : K \rrbracket^{\text{D}} \rho$	$= \text{eval} \circ \langle \text{curry}(\llbracket \Gamma; \Phi \mid \Delta, x : F \vdash l : K \rrbracket^{\text{D}} \rho, \\ \llbracket \Gamma; \Phi \mid \Delta, y : G \vdash r : K \rrbracket^{\text{D}} \rho), \\ \llbracket \Gamma; \Phi \mid \Delta \vdash t : F + G \rrbracket^{\text{D}} \rho \rangle$
$\llbracket \Gamma; \Phi \mid \Delta \vdash \text{inl } s : F + G \rrbracket^{\text{D}} \rho$	$= \text{inl} \circ \llbracket \Gamma; \Phi \mid \Delta \vdash s : F \rrbracket^{\text{D}} \rho$
$\llbracket \Gamma; \Phi \mid \Delta \vdash \text{inr } t : F + G \rrbracket^{\text{D}} \rho$	$= \text{inr} \circ \llbracket \Gamma; \Phi \mid \Delta \vdash t : G \rrbracket^{\text{D}} \rho$
$\llbracket \Gamma; \emptyset \mid \emptyset \vdash \text{map}_{\bar{H}}^{\bar{F}, \bar{G}} : \text{Nat}^{\bar{\theta}}(\text{Nat}^{\bar{\beta}, \bar{\gamma}} F G) \\ (\text{Nat}^{\bar{\gamma}} H[\phi := \bar{\beta} \bar{F}] H[\phi := \bar{\beta} \bar{G}]) \rrbracket^{\text{D}} \rho$	$= \lambda d. \bar{\eta} \bar{C}. \llbracket \Gamma; \bar{\phi}, \bar{\gamma} \vdash H \rrbracket^{\text{D}} id_{\rho[\bar{\gamma} := \bar{C}]}[\phi := \lambda \bar{B}. \bar{\eta} \bar{B} \bar{C}]$
$\llbracket \Gamma; \emptyset \mid \emptyset \vdash \text{in}_H : \text{Nat}^{\bar{\beta}} H[\phi := (\mu \phi. \lambda \bar{\alpha}. H) \bar{\beta}] [\bar{\alpha} := \bar{\beta}] \\ (\mu \phi. \lambda \bar{\alpha}. H) \bar{\beta} \rrbracket^{\text{D}} \rho$	$= \lambda d. \text{in}_{T_{H, \rho}^X}$ where X is Set when $D = \text{Set}$ and not present when $D = \text{Rel}$
$\llbracket \Gamma; \emptyset \mid \emptyset \vdash \text{fold}_H^F : \text{Nat}^{\bar{\theta}}(\text{Nat}^{\bar{\beta}} H[\phi := \bar{\beta} F] [\bar{\alpha} := \bar{\beta}] F) \\ (\text{Nat}^{\bar{\beta}} (\mu \phi. \lambda \bar{\alpha}. H) \bar{\beta} F) \rrbracket^{\text{D}} \rho$	$= \lambda d. \text{fold}_{T_{H, \rho}^X}$ where X is as above

Figure 2: Term semantics

$\langle \llbracket \Gamma; \Phi \vdash F \rrbracket^{\text{Set}} f \rangle$ then $(x, \llbracket \Gamma; \Phi \vdash F \rrbracket^{\text{Set}} f x) \in \llbracket \Gamma; \Phi \vdash F \rrbracket^{\text{Rel}} \langle f \rangle$, i.e., $\langle \llbracket \Gamma; \Phi \vdash F \rrbracket^{\text{Set}} f \rangle \subseteq \llbracket \Gamma; \Phi \vdash F \rrbracket^{\text{Rel}} \langle f \rangle$. \square

5. INTERPRETING TERMS

If $\Delta = x_1 : F_1, \dots, x_n : F_n$ is a term context for Γ and Φ , then the interpretations $\llbracket \Gamma; \Phi \vdash \Delta \rrbracket^{\text{Set}}$ and $\llbracket \Gamma; \Phi \vdash \Delta \rrbracket^{\text{Rel}}$ are defined by

$$\begin{aligned} \llbracket \Gamma; \Phi \vdash \Delta \rrbracket^{\text{Set}} &= \llbracket \Gamma; \Phi \vdash F_1 \rrbracket^{\text{Set}} \times \dots \times \llbracket \Gamma; \Phi \vdash F_n \rrbracket^{\text{Set}} \\ \llbracket \Gamma; \Phi \vdash \Delta \rrbracket^{\text{Rel}} &= \llbracket \Gamma; \Phi \vdash F_1 \rrbracket^{\text{Rel}} \times \dots \times \llbracket \Gamma; \Phi \vdash F_n \rrbracket^{\text{Rel}} \end{aligned}$$

Every well-formed term $\Gamma; \Phi \mid \Delta \vdash t : F$ then has, for every $\rho \in \text{SetEnv}$, a set interpretation $\llbracket \Gamma; \Phi \mid \Delta \vdash t : F \rrbracket^{\text{Set}} \rho$ as the component at ρ of a natural transformation from $\llbracket \Gamma; \Phi \vdash \Delta \rrbracket^{\text{Set}}$ to $\llbracket \Gamma; \Phi \vdash F \rrbracket^{\text{Set}}$, and, for every $\rho \in \text{RelEnv}$, a relational interpretation $\llbracket \Gamma; \Phi \mid \Delta \vdash t : F \rrbracket^{\text{Rel}} \rho$ as the component at ρ of a natural transformation from $\llbracket \Gamma; \Phi \vdash \Delta \rrbracket^{\text{Rel}}$ to $\llbracket \Gamma; \Phi \vdash F \rrbracket^{\text{Rel}}$.

Definition 5.1. If ρ is a set (resp., relation) environment and $\Gamma; \Phi \mid \Delta \vdash t : F$ then $\llbracket \Gamma; \Phi \mid \Delta \vdash t : F \rrbracket^{\text{Set}} \rho$ (resp., $\llbracket \Gamma; \Phi \mid \Delta \vdash t : F \rrbracket^{\text{Rel}} \rho$) is defined as in Figure 2, where D is either **Set** or **Rel** as appropriate.

If t is closed, i.e., if $\emptyset; \emptyset \mid \emptyset \vdash t : F$, then we write $\llbracket t : F \rrbracket^{\text{D}}$ instead of $\llbracket \emptyset; \emptyset \mid \emptyset \vdash t : F \rrbracket^{\text{D}}$. The interpretations in Definition 5.1 respect weakening, i.e., a term and its weakenings all have the same set and relational interpretations. Specifically, for any $\rho \in \text{SetEnv}$,

$$\llbracket \Gamma; \Phi \mid \Delta, x : F \vdash t : G \rrbracket^{\text{Set}} \rho = (\llbracket \Gamma; \Phi \mid \Delta \vdash t : G \rrbracket^{\text{Set}} \rho) \circ \pi_{\Delta}$$

where π_{Δ} is the projection $\llbracket \Gamma; \Phi \vdash \Delta, x : F \rrbracket^{\text{Set}} \rightarrow \llbracket \Gamma; \Phi \vdash \Delta \rrbracket^{\text{Set}}$. A similar result holds for relational interpretations.

The return type for the semantic fold is $\llbracket \Gamma; \bar{\beta} \vdash F \rrbracket^D \rho [\bar{\beta} := \bar{B}]$. This interpretation gives

$$\llbracket \Gamma; \emptyset \mid \Delta \vdash \lambda x. t : F \rightarrow G \rrbracket^D \rho = \text{curry}(\llbracket \Gamma; \emptyset \mid \Delta, x : F \vdash t : G \rrbracket^D \rho)$$

and

$$\llbracket \Gamma; \emptyset \mid \Delta \vdash st : G \rrbracket^D \rho = \text{eval} \circ \langle \llbracket \Gamma; \emptyset \mid \Delta \vdash s : F \rightarrow G \rrbracket^D \rho, \llbracket \Gamma; \emptyset \mid \Delta \vdash t : F \rrbracket^D \rho \rangle$$

so it specializes to the standard interpretations for System F terms. Term interpretation also respects substitution for both functorial and non-functorial type variables, as well as term substitution. That is, if $\Gamma, \alpha; \Phi \mid \Delta \vdash t : F$ and $\Gamma; \Phi, \alpha \mid \Delta \vdash t' : F$ and $\Gamma; \Phi \vdash G$ then

$$\llbracket \Gamma; \Phi \mid \Delta[\alpha := G] \vdash t[\alpha := G] : F[\alpha := G] \rrbracket^D \rho = \llbracket \Gamma, \alpha; \Phi \mid \Delta \vdash t : F \rrbracket^D \rho[\alpha := \llbracket \Gamma; \Phi \vdash G \rrbracket^D \rho]$$

and

$$\llbracket \Gamma; \Phi \mid \Delta[\alpha := G] \vdash t'[\alpha := G] : F[\alpha := G] \rrbracket^D \rho = \llbracket \Gamma; \Phi, \alpha \mid \Delta \vdash t' : F \rrbracket^D \rho[\alpha := \llbracket \Gamma; \Phi \vdash G \rrbracket^D \rho]$$

and if $\Gamma; \Phi \mid \Delta, x : G \vdash t : F$ and $\Gamma; \Phi \mid \Delta \vdash s : G$ then

$$\lambda A. \llbracket \Gamma; \Phi \mid \Delta \vdash t[x := s] : F \rrbracket^D \rho A = \lambda A. \llbracket \Gamma; \Phi \mid \Delta, x : G \vdash t : F \rrbracket^D \rho(A, \llbracket \Gamma; \Phi \mid \Delta \vdash s : G \rrbracket^D \rho A)$$

Direct calculation further reveals that term interpretations also satisfy

$$\llbracket \Gamma; \Phi \mid \Delta \vdash (L_{\bar{\alpha}x}.t)_{\bar{K}} s \rrbracket^D = \llbracket \Gamma; \Phi \mid \Delta \vdash t[\bar{\alpha} := \bar{K}][x := s] \rrbracket^D$$

Add if true: $\llbracket \Gamma; \emptyset \mid \Delta \vdash L_{\bar{\alpha}x}.t_{\bar{\alpha}x} : \text{Nat}^{\bar{\alpha}} F G \rrbracket^D = \llbracket \Gamma; \emptyset \mid \Delta \vdash t : \text{Nat}^{\bar{\alpha}} F G \rrbracket^D$

6. NATURALITY AND THE ABSTRACTION THEOREM

6.1. Naturality and Its Consequences. We first show that terms of Nat -type behave as natural transformations with respect to their source and target functorial types, and derive some consequences of this observation. If $\Gamma; \bar{\alpha} \vdash F$ then define the *identity* id_F on F by $id_F = \Gamma; \emptyset \mid \emptyset \vdash L_{\bar{\alpha}x}.x : \text{Nat}^{\bar{\alpha}} F F$, and if $\Gamma; \emptyset \mid \Delta \vdash t : \text{Nat}^{\bar{\alpha}} F G$ and $\Gamma; \emptyset \mid \Delta \vdash s : \text{Nat}^{\bar{\alpha}} G H$ are terms then define the *composition* $s \circ t$ of t and s by $s \circ t = \Gamma; \emptyset \mid \Delta \vdash L_{\bar{\alpha}x}.s_{\bar{\alpha}}(t_{\bar{\alpha}}x) : \text{Nat}^{\bar{\alpha}} F H$. Then

$$\llbracket \Gamma; \emptyset \mid \emptyset \vdash id_F : \text{Nat}^{\bar{\alpha}} F F \rrbracket^{\text{Set}} \rho * = id_{\lambda \bar{A}. \llbracket \Gamma; \bar{\alpha} \vdash F \rrbracket^{\text{Set}} \rho [\bar{\alpha} := \bar{A}]}$$

for any set environment ρ , and

$$\llbracket \Gamma; \emptyset \mid \Delta \vdash s \circ t : \text{Nat}^{\bar{\alpha}} F H \rrbracket^{\text{Set}} = \llbracket \Gamma; \emptyset \mid \Delta \vdash s : \text{Nat}^{\bar{\alpha}} G H \rrbracket^{\text{Set}} \circ \llbracket \Gamma; \emptyset \mid \Delta \vdash t : \text{Nat}^{\bar{\alpha}} F G \rrbracket^{\text{Set}}$$

Naturality of term interpretations is then easily verified:

Theorem 6.1. *If $\Gamma; \emptyset \mid \Delta \vdash s : \text{Nat}^{\bar{\alpha}, \bar{\gamma}} F G$ and $\Gamma; \emptyset \mid \Delta \vdash t : \text{Nat}^{\bar{\gamma}} K H$, then*

$$\begin{aligned} & \llbracket \Gamma; \emptyset \mid \Delta \vdash ((\text{map}_G^{\bar{K}, \bar{H}})_{\emptyset} \bar{t}) \circ (L_{\bar{\gamma}z}.s_{\bar{K}, \bar{\gamma}}z) : \text{Nat}^{\bar{\gamma}} F[\bar{\alpha} := \bar{K}] G[\bar{\alpha} := \bar{H}] \rrbracket^{\text{Set}} \\ &= \llbracket \Gamma; \emptyset \mid \Delta \vdash (L_{\bar{\gamma}z}.s_{\bar{H}, \bar{\gamma}}z) \circ ((\text{map}_F^{\bar{K}, \bar{H}})_{\emptyset} \bar{t}) : \text{Nat}^{\bar{\gamma}} F[\bar{\alpha} := \bar{K}] G[\bar{\alpha} := \bar{H}] \rrbracket^{\text{Set}} \end{aligned}$$

Theorem 6.1 gives rise to an entire family of free theorems that are consequences of naturality, and thus do not require the full power of parametricity. In particular, we can prove all of the standard properties of the initial algebraic constructs map , in , and fold . We have, for example, that the interpretation of every map_H is a functor, i.e., if $\Gamma; \bar{\alpha}, \bar{\gamma} \vdash H$, $\Gamma; \emptyset \mid \Delta \vdash g : \text{Nat}^{\bar{\gamma}} F G$, and $\Gamma; \emptyset \mid \Delta \vdash f : \text{Nat}^{\bar{\gamma}} G K$, then

$$\begin{aligned} & \llbracket \Gamma; \emptyset \mid \Delta \vdash (\text{map}_H^{\bar{F}, \bar{K}})_{\emptyset} (\bar{f} \circ \bar{g}) : \text{Nat}^{\bar{\gamma}} H[\bar{\alpha} := \bar{F}] H[\bar{\alpha} := \bar{K}] \rrbracket^{\text{Set}} \\ &= \llbracket \Gamma; \emptyset \mid \Delta \vdash (\text{map}_H^{\bar{G}, \bar{K}})_{\emptyset} \bar{f} \circ (\text{map}_H^{\bar{F}, \bar{G}})_{\emptyset} \bar{g} : \text{Nat}^{\bar{\gamma}} H[\bar{\alpha} := \bar{F}] H[\bar{\alpha} := \bar{K}] \rrbracket^{\text{Set}} \end{aligned}$$

In fact, we have that if $\Gamma; \bar{\alpha} \vdash H$, $\Gamma; \emptyset \vdash \bar{F}$, and $\Gamma; \emptyset \vdash \bar{G}$, then, for all $f \in \llbracket \Gamma; \emptyset \vdash \text{Nat}^0 F G \rrbracket^{\text{Set}} \rho$,

$$\begin{aligned} & \llbracket \Gamma; \emptyset \mid x : \text{Nat}^0 F G \vdash (\text{map}_{H, \bar{F}, \bar{G}}^{\bar{F}, \bar{G}})_\emptyset \bar{x} : \text{Nat}^0 H[\bar{\alpha} := \bar{F}] H[\bar{\alpha} := \bar{G}] \rrbracket^{\text{Set}} \rho f \\ &= \llbracket \Gamma; \bar{\alpha} \vdash H \rrbracket^{\text{Set}} id_\rho[\bar{\alpha} := f] \\ &= \text{map}_{\lambda \bar{A}. \llbracket \Gamma; \bar{\alpha} \vdash H \rrbracket^{\text{Set}} \rho[\bar{\alpha} := \bar{A}]} \bar{f} \end{aligned}$$

Here, we obtain the first equality from the appropriate instance of Definition 5.1, and the second one by noting that $\lambda \bar{A}. \llbracket \Gamma; \bar{\alpha} \vdash H \rrbracket^{\text{Set}} \rho[\bar{\alpha} := \bar{A}]$ is a functor in $\bar{\alpha}$ and using $\text{map}_G \bar{f}$ to denote the action of the semantic functor G on morphisms \bar{f} . We also have, e.g., that:

- **map** is a higher-order functor, i.e., if $\Gamma; \bar{\psi}, \bar{\gamma} \vdash H$, $\Gamma; \bar{\alpha}, \bar{\gamma}, \bar{\phi} \vdash K$, $\Gamma; \bar{\beta}, \bar{\gamma} \vdash F$, $\Gamma; \bar{\beta}, \bar{\gamma} \vdash G$, and $\xi = \text{Nat}^0(\text{Nat}^{\bar{\alpha}, \bar{\beta}, \bar{\gamma}} F G)(\text{Nat}^{\bar{\gamma}} H[\bar{\psi} := \bar{K}][\bar{\phi} := \bar{F}] H[\bar{\psi} := \bar{K}][\bar{\phi} := \bar{G}])$, then

$$\llbracket \Gamma; \emptyset \mid \emptyset \vdash \text{map}_{H[\bar{\psi} := \bar{K}]}^{\bar{F}, \bar{G}} : \xi \rrbracket^{\text{Set}} = \llbracket \Gamma; \emptyset \mid \emptyset \vdash L_{\emptyset \bar{z}}.(\text{map}_H^{K[\bar{\phi} := \bar{F}], K[\bar{\phi} := \bar{G}]})_\emptyset ((\text{map}_K^{\bar{F}, \bar{G}})_\emptyset \bar{z}) : \xi \rrbracket^{\text{Set}}$$

- **fold** behaves as expected, i.e., if $\xi = \text{Nat}^{\bar{\beta}} H[\phi := (\mu\phi. \lambda \bar{\alpha}. H)\bar{\beta}][\bar{\alpha} := \bar{\beta}] F$, then

$$\begin{aligned} & \llbracket \Gamma; \emptyset \mid x : \text{Nat}^{\bar{\beta}} H[\phi := F][\bar{\alpha} := \bar{\beta}] F \vdash ((\text{fold}_{H, F})_\emptyset x) \circ \text{in}_H : \xi \rrbracket^{\text{Set}} \\ &= \llbracket \Gamma; \emptyset \mid x : \text{Nat}^{\bar{\beta}} H[\phi := F][\bar{\alpha} := \bar{\beta}] F \vdash x \circ ((\text{map}_{H[\bar{\alpha} := \bar{\beta}]}^{(\mu\phi. \lambda \bar{\alpha}. H)\bar{\beta}, F})_\emptyset ((\text{fold}_{H, F})_\emptyset x)) : \xi \rrbracket^{\text{Set}} \end{aligned}$$

- **in** has a right inverse, i.e., if $\xi = \text{Nat}^{\bar{\beta}} (\mu\phi. \lambda \bar{\alpha}. H)\bar{\beta} (\mu\phi. \lambda \bar{\alpha}. H)\bar{\beta}$, then

$$\begin{aligned} & \llbracket \Gamma; \emptyset \mid \emptyset \vdash \text{in}_H \circ (\text{fold}_{H, H[\phi := (\mu\phi. \lambda \bar{\alpha}. H)\bar{\beta}]})_\emptyset ((\text{map}_H^{H[\phi := (\mu\phi. \lambda \bar{\alpha}. H)\bar{\beta}][\bar{\alpha} := \bar{\beta}], (\mu\phi. \lambda \bar{\alpha}. H)\bar{\beta}})_\emptyset \text{in}_H) : \xi \rrbracket^{\text{Set}} \\ &= \llbracket \Gamma; \emptyset \mid \emptyset \vdash id_{(\mu\phi. \lambda \bar{\alpha}. H)\bar{\beta}} : \xi \rrbracket^{\text{Set}} \end{aligned}$$

- **in** has a left inverse, i.e., if $\xi = \text{Nat}^{\bar{\beta}} H[\phi := (\mu\phi. \lambda \bar{\alpha}. H)\bar{\beta}] H[\phi := (\mu\phi. \lambda \bar{\alpha}. H)\bar{\beta}]$, then

$$\begin{aligned} & \llbracket \Gamma; \emptyset \mid \emptyset \vdash (\text{fold}_{H, H[\phi := (\mu\phi. \lambda \bar{\alpha}. H)\bar{\beta}]})_\emptyset ((\text{map}_H^{H[\phi := (\mu\phi. \lambda \bar{\alpha}. H)\bar{\beta}][\bar{\alpha} := \bar{\beta}], (\mu\phi. \lambda \bar{\alpha}. H)\bar{\beta}})_\emptyset \text{in}_H) \circ \text{in}_H : \xi \rrbracket^{\text{Set}} \\ &= \llbracket \Gamma; \emptyset \mid \emptyset \vdash id_{H[\phi := (\mu\phi. \lambda \bar{\alpha}. H)\bar{\beta}]} : \xi \rrbracket^{\text{Set}} \end{aligned}$$

Analogous results hold for relational interpretations of terms and relation environments.

6.2. The Abstraction Theorem. To get consequences of parametricity that are not merely consequences of naturality, we prove an Abstraction Theorem (Theorem 6.4) for our calculus. As is usual for such theorems, we first prove a more general result (Theorem 6.2) for open terms, and recover our Abstraction Theorem as a special case of it for closed terms of closed type.

Theorem 6.2. *Every well-formed term $\Gamma; \Phi \mid \Delta \vdash t : F$ induces a natural transformation from $\llbracket \Gamma; \Phi \vdash \Delta \rrbracket$ to $\llbracket \Gamma; \Phi \vdash F \rrbracket$, i.e., a triple of natural transformations*

$$(\llbracket \Gamma; \Phi \mid \Delta \vdash t : F \rrbracket^{\text{Set}}, \llbracket \Gamma; \Phi \mid \Delta \vdash t : F \rrbracket^{\text{Set}}, \llbracket \Gamma; \Phi \mid \Delta \vdash t : F \rrbracket^{\text{Rel}})$$

where, for $D \in \{\text{Set}, \text{Rel}\}$ and for $\rho \in \text{SetEnv}$ or $\rho \in \text{RelEnv}$ as appropriate,

$$\llbracket \Gamma; \Phi \mid \Delta \vdash t : F \rrbracket^D : \llbracket \Gamma; \Phi \vdash \Delta \rrbracket^D \rightarrow \llbracket \Gamma; \Phi \vdash F \rrbracket^D$$

has as its component at $\rho : \text{SetEnv}$ a morphism

$$\llbracket \Gamma; \Phi \mid \Delta \vdash t : F \rrbracket^{\text{Set}} \rho : \llbracket \Gamma; \Phi \vdash \Delta \rrbracket^{\text{Set}} \rho \rightarrow \llbracket \Gamma; \Phi \vdash F \rrbracket^{\text{Set}} \rho$$

Moreover, for all $\rho : \text{RelEnv}$,

$$\llbracket \Gamma; \Phi \mid \Delta \vdash t : F \rrbracket^{\text{Rel}} \rho = (\llbracket \Gamma; \Phi \mid \Delta \vdash t : F \rrbracket^{\text{Set}} (\pi_1 \rho), \llbracket \Gamma; \Phi \mid \Delta \vdash t : F \rrbracket^{\text{Set}} (\pi_2 \rho)) \quad (6.1)$$

Proof. The proof is by induction on t . It requires showing that set and relational interpretations of term judgments are natural transformations, and that all set interpretations of terms of **Nat** types satisfy the appropriate equality preservation conditions from Definition 3.2. For the interesting cases of abstraction, application, **map**, **in**, and **fold** terms, propagating the naturality conditions is quite involved; the latter two especially require some rather delicate diagram chasing. That it is possible provides strong evidence that our development is sensible, natural, and at an appropriate level of abstraction.

The only interesting cases are the cases for abstraction, application, **map**, **in**, and **fold**. We omit the others.

- $\Gamma; \emptyset \mid \Delta \vdash L_{\bar{\alpha}} x.t : \text{Nat}^{\bar{\alpha}} F G$ To see that $\llbracket \Gamma; \emptyset \mid \Delta \vdash L_{\bar{\alpha}} x.t : \text{Nat}^{\bar{\alpha}} F G \rrbracket^{\text{Set}}$ is a natural transformation from $\llbracket \Gamma; \emptyset \vdash \Delta \rrbracket^{\text{Set}}$ to $\llbracket \Gamma; \emptyset \vdash \text{Nat}^{\bar{\alpha}} F G \rrbracket^{\text{Set}}$ we need to show that, for every $\rho : \text{SetEnv}$, $\llbracket \Gamma; \emptyset \mid \Delta \vdash L_{\bar{\alpha}} x.t : \text{Nat}^{\bar{\alpha}} F G \rrbracket^{\text{Set}} \rho$ is a morphism in **Set** from $\llbracket \Gamma; \emptyset \vdash \Delta \rrbracket^{\text{Set}} \rho$ to $\llbracket \Gamma; \emptyset \vdash \text{Nat}^{\bar{\alpha}} F G \rrbracket^{\text{Set}} \rho$, and that such a family of morphisms is natural. First, we need to show that, for all $\bar{A} : \text{Set}$ and all $d : \llbracket \Gamma; \emptyset \vdash \Delta \rrbracket^{\text{Set}} \rho = \llbracket \Gamma; \bar{\alpha} \vdash \Delta \rrbracket^{\text{Set}} \rho[\bar{\alpha} := \bar{A}]$, we have that

$$\begin{aligned} & (\llbracket \Gamma; \emptyset \mid \Delta \vdash L_{\bar{\alpha}} x.t : \text{Nat}^{\bar{\alpha}} F G \rrbracket^{\text{Set}} \rho d)_{\bar{A}} : \\ & \llbracket \Gamma; \bar{\alpha} \vdash F \rrbracket^{\text{Set}} \rho[\bar{\alpha} := \bar{A}] \rightarrow \llbracket \Gamma; \bar{\alpha} \vdash G \rrbracket^{\text{Set}} \rho[\bar{\alpha} := \bar{A}] \end{aligned}$$

but this follows easily from the induction hypothesis. That these maps comprise a natural transformation $\eta : \llbracket \Gamma; \bar{\alpha} \vdash F \rrbracket^{\text{Set}} \rho[\bar{\alpha} := _] \rightarrow \llbracket \Gamma; \bar{\alpha} \vdash G \rrbracket^{\text{Set}} \rho[\bar{\alpha} := _]$ is clear since $\eta_{\bar{A}} = \text{curry}(\llbracket \Gamma; \bar{\alpha} \mid \Delta, x : F \vdash t : G \rrbracket^{\text{Set}} \rho[\bar{\alpha} := \bar{A}])$ is the component at \bar{A} of the partial specialization to d of the natural transformation

$$\llbracket \Gamma; \bar{\alpha} \mid \Delta, x : F \vdash t : G \rrbracket^{\text{Set}} \rho[\bar{\alpha} := _]$$

To see that the components of η also satisfy the additional condition needed for η to be in $\llbracket \Gamma; \emptyset \vdash \text{Nat}^{\bar{\alpha}} F G \rrbracket^{\text{Set}} \rho$, let $\bar{R} : \text{Rel}(A, B)$ and suppose

$$\begin{aligned} (u, v) & \in \llbracket \Gamma; \bar{\alpha} \vdash F \rrbracket^{\text{Rel}} \text{Eq}_{\rho}[\bar{\alpha} := \bar{R}] \\ & = (\llbracket \Gamma; \bar{\alpha} \vdash F \rrbracket^{\text{Set}} \rho[\bar{\alpha} := \bar{A}], \\ & \quad \llbracket \Gamma; \bar{\alpha} \vdash F \rrbracket^{\text{Set}} \rho[\bar{\alpha} := \bar{B}], (\llbracket \Gamma; \bar{\alpha} \vdash F \rrbracket^{\text{Rel}} \text{Eq}_{\rho}[\bar{\alpha} := \bar{R}])^*) \end{aligned}$$

Then the induction hypothesis and $(d, d) \in \llbracket \Gamma; \emptyset \vdash \Delta \rrbracket^{\text{Rel}} \text{Eq}_{\rho} = \llbracket \Gamma; \emptyset \vdash \Delta \rrbracket^{\text{Rel}} \text{Eq}_{\rho}[\bar{\alpha} := \bar{R}]$ ensure that

$$\begin{aligned} & (\eta_{\bar{A}} u, \eta_{\bar{B}} v) \\ & = (\text{curry}(\llbracket \Gamma; \bar{\alpha} \mid \Delta, x : F \vdash t : G \rrbracket^{\text{Set}} \rho[\bar{\alpha} := \bar{A}]) d u, \\ & \quad \text{curry}(\llbracket \Gamma; \bar{\alpha} \mid \Delta, x : F \vdash t : G \rrbracket^{\text{Set}} \rho[\bar{\alpha} := \bar{B}]) d v) \\ & = \text{curry}(\llbracket \Gamma; \bar{\alpha} \mid \Delta, x : F \vdash t : G \rrbracket^{\text{Rel}} \text{Eq}_{\rho}[\bar{\alpha} := \bar{R}]) (d, d) (u, v) \\ & : \llbracket \Gamma; \bar{\alpha} \vdash G \rrbracket^{\text{Rel}} \text{Eq}_{\rho}[\bar{\alpha} := \bar{R}] \end{aligned}$$

Moreover, $\llbracket \Gamma; \emptyset \mid \Delta \vdash L_{\bar{\alpha}} x.t : \text{Nat}^{\bar{\alpha}} F G \rrbracket^{\text{Set}} \rho$ is trivially natural in ρ , as the functorial action of $\llbracket \Gamma; \emptyset \vdash \Delta \rrbracket^{\text{Set}}$ and $\llbracket \Gamma; \emptyset \vdash \text{Nat}^{\bar{\alpha}} F G \rrbracket^{\text{Set}}$ on morphisms is the identity.

- $\Gamma; \Phi \mid \Delta \vdash t_{\bar{K}} s : G[\bar{\alpha} := \bar{K}]$ To see that $\llbracket \Gamma; \Phi \mid \Delta \vdash t_{\bar{K}} s : G[\bar{\alpha} := \bar{K}] \rrbracket^{\text{Set}}$ is a natural transformation from $\llbracket \Gamma; \Phi \vdash \Delta \rrbracket^{\text{Set}}$ to $\llbracket \Gamma; \Phi \vdash G[\bar{\alpha} := \bar{K}] \rrbracket^{\text{Set}}$ we show that, for every $\rho : \text{SetEnv}$, $\llbracket \Gamma; \Phi \mid \Delta \vdash t_{\bar{K}} s : G[\bar{\alpha} := \bar{K}] \rrbracket^{\text{Set}} \rho$ is a morphism from $\llbracket \Gamma; \Phi \vdash \Delta \rrbracket^{\text{Set}} \rho$ to $\llbracket \Gamma; \Phi \vdash G[\bar{\alpha} := \bar{K}] \rrbracket^{\text{Set}} \rho$, and that this family of morphisms is natural in ρ . Let $d : \llbracket \Gamma; \Phi \vdash \Delta \rrbracket^{\text{Set}} \rho$. Then

$$\begin{aligned} & \llbracket \Gamma; \Phi \mid \Delta \vdash t_{\bar{K}} s : G[\bar{\alpha} := \bar{K}] \rrbracket^{\text{Set}} \rho d \\ & = (\text{eval} \circ \langle (\llbracket \Gamma; \emptyset \mid \Delta \vdash t : \text{Nat}^{\bar{\alpha}} F G \rrbracket^{\text{Set}} \rho _)_{\llbracket \Gamma; \Phi \vdash \bar{K} \rrbracket^{\text{Set}} \rho}, \llbracket \Gamma; \Phi \mid \Delta \vdash s : F[\bar{\alpha} := \bar{K}] \rrbracket^{\text{Set}} \rho \rangle) d \\ & = \text{eval}((\llbracket \Gamma; \emptyset \mid \Delta \vdash t : \text{Nat}^{\bar{\alpha}} F G \rrbracket^{\text{Set}} \rho _)_{\llbracket \Gamma; \Phi \vdash \bar{K} \rrbracket^{\text{Set}} \rho} d, \llbracket \Gamma; \Phi \mid \Delta \vdash s : F[\bar{\alpha} := \bar{K}] \rrbracket^{\text{Set}} \rho d) \\ & = \text{eval}((\llbracket \Gamma; \emptyset \mid \Delta \vdash t : \text{Nat}^{\bar{\alpha}} F G \rrbracket^{\text{Set}} \rho d)_{\llbracket \Gamma; \Phi \vdash \bar{K} \rrbracket^{\text{Set}} \rho}, \llbracket \Gamma; \Phi \mid \Delta \vdash s : F[\bar{\alpha} := \bar{K}] \rrbracket^{\text{Set}} \rho d) \end{aligned}$$

The induction hypothesis ensures that $(\llbracket \Gamma; \emptyset \mid \Delta \vdash t : \text{Nat}^{\bar{\alpha}} F G \rrbracket^{\text{Set}} \rho d)_{\llbracket \Gamma; \Phi \vdash K \rrbracket^{\text{Set}} \rho}$ has type $\llbracket \Gamma; \bar{\alpha} \vdash F \rrbracket^{\text{Set}} \rho [\alpha := \llbracket \Gamma; \Phi \vdash K \rrbracket^{\text{Set}} \rho]$. Since, in addition,

$$\begin{aligned} & \llbracket \Gamma; \Phi \mid \Delta \vdash s : F[\alpha := \bar{K}] \rrbracket^{\text{Set}} \rho d : \llbracket \Gamma; \Phi \vdash F[\alpha := \bar{K}] \rrbracket^{\text{Set}} \rho \\ &= \llbracket \Gamma; \Phi, \bar{\alpha} \vdash F \rrbracket^{\text{Set}} \rho [\alpha := \llbracket \Gamma; \Phi \vdash K \rrbracket^{\text{Set}} \rho] \\ &= \llbracket \Gamma; \bar{\alpha} \vdash F \rrbracket^{\text{Set}} \rho [\alpha := \llbracket \Gamma; \Phi \vdash K \rrbracket^{\text{Set}} \rho] \end{aligned}$$

we have that

$$\begin{aligned} & \llbracket \Gamma; \Phi \mid \Delta \vdash t_{\bar{K}} s : G[\alpha := \bar{K}] \rrbracket^{\text{Set}} \rho d : \llbracket \Gamma; \Phi, \bar{\alpha} \vdash G \rrbracket^{\text{Set}} \rho [\alpha := \llbracket \Gamma; \Phi \vdash K \rrbracket^{\text{Set}} \rho] \\ &= \llbracket \Gamma; \Phi \vdash G[\alpha := \bar{K}] \rrbracket^{\text{Set}} \rho \end{aligned}$$

as desired.

To see that the family of maps comprising $\llbracket \Gamma; \Phi \mid \Delta \vdash t_{\bar{K}} s : G[\alpha := \bar{K}] \rrbracket^{\text{Set}}$ is natural in ρ we need to show that, if $f : \rho \rightarrow \rho'$ in SetEnv , then the following diagram commutes, where $g = \llbracket \Gamma; \emptyset \mid \Delta \vdash t : \text{Nat}^{\bar{\alpha}} F G \rrbracket^{\text{Set}}$ and $h = \llbracket \Gamma; \Phi \mid \Delta \vdash s : F[\alpha := \bar{K}] \rrbracket^{\text{Set}}$:

$$\begin{array}{ccc} \llbracket \Gamma; \Phi \vdash \Delta \rrbracket^{\text{Set}} \rho & \xrightarrow{\llbracket \Gamma; \Phi \vdash \Delta \rrbracket^{\text{Set}} f} & \llbracket \Gamma; \Phi \vdash \Delta \rrbracket^{\text{Set}} \rho' \\ \downarrow \langle g\rho, h\rho \rangle & \llbracket \Gamma; \emptyset \vdash \text{Nat}^{\bar{\alpha}} F G \rrbracket^{\text{Set}} f \times \llbracket \Gamma; \Phi \vdash F[\alpha := \bar{K}] \rrbracket^{\text{Set}} f \downarrow \langle g\rho', h\rho' \rangle & \\ \llbracket \Gamma; \emptyset \vdash \text{Nat}^{\bar{\alpha}} F G \rrbracket^{\text{Set}} \rho \times \llbracket \Gamma; \Phi \vdash F[\alpha := \bar{K}] \rrbracket^{\text{Set}} \rho & \xrightarrow{\quad} & \llbracket \Gamma; \emptyset \vdash \text{Nat}^{\bar{\alpha}} F G \rrbracket^{\text{Set}} \rho' \times \llbracket \Gamma; \Phi \vdash F[\alpha := \bar{K}] \rrbracket^{\text{Set}} \rho' \\ \downarrow \text{evalo}((-)_{\llbracket \Gamma; \Phi \vdash K \rrbracket^{\text{Set}} \rho} \times id) & & \downarrow \text{evalo}((-)_{\llbracket \Gamma; \Phi \vdash K \rrbracket^{\text{Set}} \rho'} \times id) \\ \llbracket \Gamma; \Phi \vdash G[\alpha := \bar{K}] \rrbracket^{\text{Set}} \rho & \xrightarrow{\llbracket \Gamma; \Phi \vdash G[\alpha := \bar{K}] \rrbracket^{\text{Set}} f} & \llbracket \Gamma; \Phi \vdash G[\alpha := \bar{K}] \rrbracket^{\text{Set}} \rho' \end{array}$$

The top diagram commutes because g and h are natural in ρ by the induction hypothesis. To see that the bottom diagram commutes, we need to show that

$$\begin{aligned} & \llbracket \Gamma; \Phi \vdash G[\alpha := \bar{K}] \rrbracket^{\text{Set}} f(\eta_{\llbracket \Gamma; \Phi \vdash K \rrbracket^{\text{Set}} \rho} x) \\ &= (\llbracket \Gamma; \emptyset \vdash \text{Nat}^{\bar{\alpha}} F G \rrbracket^{\text{Set}} f \eta)_{\llbracket \Gamma; \Phi \vdash K \rrbracket^{\text{Set}} \rho'}(\llbracket \Gamma; \Phi \vdash F[\alpha := \bar{K}] \rrbracket^{\text{Set}} f x) \end{aligned}$$

holds for all $\eta \in \llbracket \Gamma; \emptyset \vdash \text{Nat}^{\bar{\alpha}} F G \rrbracket^{\text{Set}} \rho$ and $x \in \llbracket \Gamma; \Phi \vdash F[\alpha := \bar{K}] \rrbracket^{\text{Set}} \rho$, i.e., that

$$\begin{aligned} & \llbracket \Gamma; \bar{\alpha} \vdash G \rrbracket^{\text{Set}} f[\alpha := \llbracket \Gamma; \Phi \vdash K \rrbracket^{\text{Set}} f] \circ \eta_{\llbracket \Gamma; \Phi \vdash K \rrbracket^{\text{Set}} \rho} \\ &= \eta_{\llbracket \Gamma; \Phi \vdash K \rrbracket^{\text{Set}} \rho'} \circ \llbracket \Gamma; \bar{\alpha} \vdash F \rrbracket^{\text{Set}} f[\alpha := \llbracket \Gamma; \Phi \vdash K \rrbracket^{\text{Set}} f] \end{aligned}$$

for all $\eta \in \llbracket \Gamma; \emptyset \vdash \text{Nat}^{\bar{\alpha}} F G \rrbracket^{\text{Set}} \rho$. But this follows from the naturality of η , which indeed ensures the commutativity of

$$\begin{array}{ccc} \llbracket \Gamma; \bar{\alpha} \vdash F \rrbracket^{\text{Set}} \rho [\alpha := \llbracket \Gamma; \Phi \vdash K \rrbracket^{\text{Set}} \rho] & \xrightarrow{\eta_{\llbracket \Gamma; \Phi \vdash K \rrbracket^{\text{Set}} \rho}} & \llbracket \Gamma; \bar{\alpha} \vdash G \rrbracket^{\text{Set}} \rho [\alpha := \llbracket \Gamma; \Phi \vdash K \rrbracket^{\text{Set}} \rho] \\ \downarrow \llbracket \Gamma; \bar{\alpha} \vdash F \rrbracket^{\text{Set}} f[\alpha := \llbracket \Gamma; \Phi \vdash K \rrbracket^{\text{Set}} f] & & \downarrow \llbracket \Gamma; \bar{\alpha} \vdash G \rrbracket^{\text{Set}} f[\alpha := \llbracket \Gamma; \Phi \vdash K \rrbracket^{\text{Set}} f] \\ \llbracket \Gamma; \bar{\alpha} \vdash F \rrbracket^{\text{Set}} \rho' [\alpha := \llbracket \Gamma; \Phi \vdash K \rrbracket^{\text{Set}} \rho'] & \xrightarrow{\eta_{\llbracket \Gamma; \Phi \vdash K \rrbracket^{\text{Set}} \rho'}} & \llbracket \Gamma; \bar{\alpha} \vdash G \rrbracket^{\text{Set}} \rho' [\alpha := \llbracket \Gamma; \Phi \vdash K \rrbracket^{\text{Set}} \rho'] \end{array}$$

- $\Gamma; \emptyset \mid \emptyset \vdash \text{map}_H^{\bar{F}, \bar{G}} : \text{Nat}^{\emptyset} (\text{Nat}^{\bar{\beta}, \bar{\gamma}} F G) (\text{Nat}^{\bar{\gamma}} H[\phi :=_{\bar{\beta}} \bar{F}] H[\phi :=_{\bar{\beta}} \bar{G}])$ To see that

$$\llbracket \Gamma; \emptyset \mid \emptyset \vdash \text{map}_H^{\bar{F}, \bar{G}} : \text{Nat}^{\emptyset} (\text{Nat}^{\bar{\beta}, \bar{\gamma}} F G) (\text{Nat}^{\bar{\gamma}} H[\phi :=_{\bar{\beta}} \bar{F}] H[\phi :=_{\bar{\beta}} \bar{G}]) \rrbracket^{\text{Set}} \rho d \bar{\eta}$$

is in $\llbracket \Gamma; \emptyset \vdash \text{Nat}^{\bar{\gamma}} H[\phi :=_{\bar{\beta}} \bar{F}] H[\phi :=_{\bar{\beta}} \bar{G}] \rrbracket^{\text{Set}} \rho$ for all $\rho : \text{SetEnv}$, $d : \llbracket \Gamma; \emptyset \vdash \emptyset \rrbracket^{\text{Set}} \rho$, and $\bar{\eta} : \llbracket \Gamma; \emptyset \vdash \text{Nat}^{\bar{\beta}, \bar{\gamma}} F G \rrbracket^{\text{Set}} \rho$, we first note that $\llbracket \Gamma; \bar{\phi}, \bar{\gamma} \vdash H \rrbracket^{\text{Set}}$ is a functor from SetEnv to Set and, for any \bar{C} , $id_{\rho[\bar{\gamma} := \bar{C}]}[\phi := \lambda \bar{B}. \eta_{\bar{B}} \bar{C}]$ is a morphism in SetEnv from

$$\rho[\bar{\gamma} := \bar{C}][\phi := \lambda \bar{B}. \llbracket \Gamma; \bar{\gamma}, \bar{\beta} \vdash F \rrbracket^{\text{Set}} \rho[\bar{\gamma} := \bar{C}][\bar{\beta} := \bar{B}]]$$

to

$$\rho[\overline{\gamma := C}][\overline{\phi := \lambda \overline{B}. \llbracket \Gamma; \overline{\gamma}, \overline{\beta} \vdash G \rrbracket^{\text{Set}} \rho[\overline{\gamma := C}][\overline{\beta := \overline{B}}]}]$$

so that

$$\begin{aligned} & (\llbracket \Gamma; \emptyset \mid \emptyset \vdash \text{map}_H^{\overline{F}, \overline{G}} : \text{Nat}^\emptyset (\text{Nat}^{\overline{\beta}, \overline{\gamma}} F G) (\text{Nat}^{\overline{\gamma}} H[\overline{\phi := \overline{\beta}} F] H[\overline{\phi := \overline{\beta}} G]) \rrbracket^{\text{Set}} \rho d \overline{\eta})_{\overline{C}} \\ &= \llbracket \Gamma; \overline{\phi}, \overline{\gamma} \vdash H \rrbracket^{\text{Set}} id_{\rho[\overline{\gamma := C}][\overline{\phi := \lambda \overline{B}. \eta_{\overline{B}} \overline{C}}]} \end{aligned}$$

is indeed a morphism of type

$$\llbracket \Gamma; \overline{\gamma} \vdash H[\overline{\phi := F}] \rrbracket^{\text{Set}} \rho[\overline{\gamma := C}] \rightarrow \llbracket \Gamma; \overline{\gamma} \vdash H[\overline{\phi := G}] \rrbracket^{\text{Set}} \rho[\overline{\gamma := C}]$$

This family of morphisms is natural in \overline{C} : if $\overline{f} : C \rightarrow C'$ then, writing ξ for

$$\llbracket \Gamma; \emptyset \mid \emptyset \vdash \text{map}_H^{\overline{F}, \overline{G}} : \text{Nat}^\emptyset (\text{Nat}^{\overline{\beta}, \overline{\gamma}} F G) (\text{Nat}^{\overline{\gamma}} H[\overline{\phi := \overline{\beta}} F] H[\overline{\phi := \overline{\beta}} G]) \rrbracket^{\text{Set}} \rho d \overline{\eta}$$

the naturality of η , together with the fact that composition of environments is computed componentwise, ensure that the following naturality diagram for ξ commutes:

$$\begin{array}{ccc} \llbracket \Gamma; \overline{\gamma} \vdash H[\overline{\phi := F}] \rrbracket^{\text{Set}} \rho[\overline{\gamma := C}] & \xrightarrow{\xi_{\overline{C}}} & \llbracket \Gamma; \overline{\gamma} \vdash H[\overline{\phi := G}] \rrbracket^{\text{Set}} \rho[\overline{\gamma := C}] \\ \downarrow \llbracket \Gamma; \overline{\gamma} \vdash H[\overline{\phi := F}] \rrbracket^{\text{Set}} id_{\rho[\overline{\gamma := C}]} & & \downarrow \llbracket \Gamma; \overline{\gamma} \vdash H[\overline{\phi := G}] \rrbracket^{\text{Set}} id_{\rho[\overline{\gamma := C}]} \\ \llbracket \Gamma; \overline{\gamma} \vdash H[\overline{\phi := F}] \rrbracket^{\text{Set}} \rho[\overline{\gamma := C'}] & \xrightarrow{\xi_{\overline{C'}}} & \llbracket \Gamma; \overline{\gamma} \vdash H[\overline{\phi := G}] \rrbracket^{\text{Set}} \rho[\overline{\gamma := C'}] \end{array}$$

That, for all $\rho : \text{SetEnv}$ and $d : \llbracket \Gamma; \emptyset \vdash \emptyset \rrbracket^{\text{Set}} \rho$, ξ satisfies the additional condition needed for it to be in $\llbracket \Gamma; \emptyset \vdash \text{Nat}^{\overline{\gamma}} H[\overline{\phi := \overline{\beta}} F] H[\overline{\phi := \overline{\beta}} G] \rrbracket^{\text{Set}} \rho$ follows from the fact that η satisfies the extra condition needed for it to be in its corresponding $\llbracket \Gamma; \emptyset \vdash \text{Nat}^{\overline{\beta}, \overline{\gamma}} F G \rrbracket^{\text{Set}} \rho$. Finally, since $\Phi = \emptyset$, the naturality of

$$\llbracket \Gamma; \emptyset \mid \emptyset \vdash \text{map}_H^{\overline{F}, \overline{G}} : \text{Nat}^\emptyset (\text{Nat}^{\overline{\beta}, \overline{\gamma}} F G) (\text{Nat}^{\overline{\gamma}} H[\overline{\phi := \overline{\beta}} F] H[\overline{\phi := \overline{\beta}} G]) \rrbracket^{\text{Set}} \rho$$

in ρ is trivial.

- $\Gamma; \emptyset \mid \emptyset \vdash \text{in}_H : \text{Nat}^{\overline{\beta}} H[\overline{\phi := (\mu \phi. \lambda \overline{\alpha}. H) \overline{\beta}}][\overline{\alpha := \overline{\beta}}] (\mu \phi. \lambda \overline{\alpha}. H) \overline{\beta}$ To see that if $d : \llbracket \Gamma; \emptyset \vdash \emptyset \rrbracket^{\text{Set}} \rho$ then

$$\llbracket \Gamma; \emptyset \mid \emptyset \vdash \text{in}_H : \text{Nat}^{\overline{\beta}} H[\overline{\phi := (\mu \phi. \lambda \overline{\alpha}. H) \overline{\beta}}][\overline{\alpha := \overline{\beta}}] (\mu \phi. \lambda \overline{\alpha}. H) \overline{\beta} \rrbracket^{\text{Set}} \rho d$$

is in $\llbracket \Gamma; \emptyset \vdash \text{Nat}^{\overline{\beta}} H[\overline{\phi := (\mu \phi. \lambda \overline{\alpha}. H) \overline{\beta}}][\overline{\alpha := \overline{\beta}}] (\mu \phi. \lambda \overline{\alpha}. H) \overline{\beta} \rrbracket^{\text{Set}} \rho$, we first note that, for all \overline{B} ,

$$\begin{aligned} & (\llbracket \Gamma; \emptyset \mid \emptyset \vdash \text{in}_H : \text{Nat}^{\overline{\beta}} H[\overline{\phi := (\mu \phi. \lambda \overline{\alpha}. H) \overline{\beta}}][\overline{\alpha := \overline{\beta}}] (\mu \phi. \lambda \overline{\alpha}. H) \overline{\beta} \rrbracket^{\text{Set}} \rho d)_{\overline{B}} \\ &= (in_{T_{H, \rho}^{\text{Set}}})_{\overline{B}} \end{aligned}$$

maps

$$\begin{aligned} & \llbracket \Gamma; \overline{\beta} \vdash H[\overline{\phi := (\mu \phi. \lambda \overline{\alpha}. H) \overline{\beta}}][\overline{\alpha := \overline{\beta}}] \rrbracket^{\text{Set}} \rho[\overline{\beta := \overline{B}}] \\ &= T_{H, \rho}^{\text{Set}} (\mu T_{H, \rho}^{\text{Set}}) \overline{B} \end{aligned}$$

to $\llbracket \Gamma; \overline{\beta} \vdash (\mu \phi. \lambda \overline{\alpha}. H) \overline{\beta} \rrbracket^{\text{Set}} \rho[\overline{\beta := \overline{B}}] = (\mu T_{H, \rho}^{\text{Set}}) \overline{B}$. Secondly, we observe that

$$\begin{aligned} & \llbracket \Gamma; \emptyset \mid \emptyset \vdash \text{in}_H : \text{Nat}^{\overline{\beta}} H[\overline{\phi := (\mu \phi. \lambda \overline{\alpha}. H) \overline{\beta}}][\overline{\alpha := \overline{\beta}}] (\mu \phi. \lambda \overline{\alpha}. H) \overline{\beta} \rrbracket^{\text{Set}} \rho d \\ &= in_{T_{H, \rho}^{\text{Set}}} \end{aligned}$$

is natural in \overline{B} , since naturality of $in_{T_{H, \rho}^{\text{Set}}}$ ensures that the following diagram commutes for all $\overline{f} : B \rightarrow B'$:

$$\begin{array}{ccc}
T_{H,\rho}^{\text{Set}} (\mu T_{H,\rho}^{\text{Set}}) \overline{B} & \xrightarrow{(in_{T_{H,\rho}^{\text{Set}}}) \overline{B}} & (\mu T_{H,\rho}^{\text{Set}}) \overline{B} \\
\downarrow T_{H,\rho}^{\text{Set}} (\mu T_{H,\rho}^{\text{Set}}) \overline{f} & & \downarrow \mu T_{H,\rho}^{\text{Set}} \overline{f} \\
T_{H,\rho}^{\text{Set}} (\mu T_{H,\rho}^{\text{Set}}) \overline{B'} & \xrightarrow{(in_{T_{H,\rho}^{\text{Set}}}) \overline{B'}} & (\mu T_{H,\rho}^{\text{Set}}) \overline{B'}
\end{array}$$

That, for all $\rho : \text{SetEnv}$ and $d : \llbracket \Gamma; \emptyset \vdash \emptyset \rrbracket^{\text{Set}} \rho$,

$$\llbracket \Gamma; \emptyset \mid \emptyset \vdash in_H : \text{Nat}^{\overline{\beta}} H[\phi := (\mu\phi.\lambda\overline{\alpha}.H)\overline{\beta}][\overline{\alpha} := \overline{\beta}] (\mu\phi.\lambda\overline{\alpha}.H)\overline{\beta} \rrbracket^{\text{Set}} \rho d$$

satisfies the additional property needed for it to be in

$$\llbracket \Gamma; \emptyset \vdash \text{Nat}^{\overline{\beta}} H[\phi := (\mu\phi.\lambda\overline{\alpha}.H)\overline{\beta}][\overline{\alpha} := \overline{\beta}] (\mu\phi.\lambda\overline{\alpha}.H)\overline{\beta} \rrbracket^{\text{Set}} \rho$$

follows from the fact that, for every $\overline{R} : \text{Rel}(B, B')$,

$$\begin{aligned}
& ((\llbracket \Gamma; \emptyset \mid \emptyset \vdash in_H : \text{Nat}^{\overline{\beta}} H[\phi := (\mu\phi.\lambda\overline{\alpha}.H)\overline{\beta}][\overline{\alpha} := \overline{\beta}] (\mu\phi.\lambda\overline{\alpha}.H)\overline{\beta} \rrbracket^{\text{Set}} \rho d)_{\overline{B}}, \\
& (\llbracket \Gamma; \emptyset \mid \emptyset \vdash in_H : \text{Nat}^{\overline{\beta}} H[\phi := (\mu\phi.\lambda\overline{\alpha}.H)\overline{\beta}][\overline{\alpha} := \overline{\beta}] (\mu\phi.\lambda\overline{\alpha}.H)\overline{\beta} \rrbracket^{\text{Set}} \rho d)_{\overline{B'}} \\
& = ((in_{T_{H,\rho}^{\text{Set}}})_{\overline{B}}, (in_{T_{H,\rho}^{\text{Set}}})_{\overline{B'}})
\end{aligned}$$

has type

$$\begin{aligned}
& (T_{H,\rho}^{\text{Set}} (\mu T_{H,\rho}^{\text{Set}}) \overline{B} \rightarrow (\mu T_{H,\rho}^{\text{Set}}) \overline{B}, T_{H,\rho}^{\text{Set}} (\mu T_{H,\rho}^{\text{Set}}) \overline{B'} \rightarrow (\mu T_{H,\rho}^{\text{Set}}) \overline{B'}) \\
& = \llbracket \Gamma; \overline{\beta} \vdash H[\phi := (\mu\phi.\lambda\overline{\alpha}.H)\overline{\beta}][\overline{\alpha} := \overline{\beta}] \rrbracket^{\text{RelEq}_\rho} [\beta := \overline{R}] \rightarrow \\
& \quad \llbracket \Gamma; \overline{\beta} \vdash (\mu\phi.\lambda\overline{\alpha}.H)\overline{\beta} \rrbracket^{\text{RelEq}_\rho} [\beta := \overline{R}]
\end{aligned}$$

Finally, since $\Phi = \emptyset$, naturality of

$$\llbracket \Gamma; \emptyset \mid \emptyset \vdash in_H : \text{Nat}^{\overline{\beta}} H[\phi := (\mu\phi.\lambda\overline{\alpha}.H)\overline{\beta}][\overline{\alpha} := \overline{\beta}] (\mu\phi.\lambda\overline{\alpha}.H)\overline{\beta} \rrbracket^{\text{Set}}$$

in ρ is trivial.

- $\Gamma; \emptyset \mid \emptyset \vdash \text{fold}_H^F : \text{Nat}^{\overline{\beta}} (\text{Nat}^{\overline{\beta}} H[\phi :=_{\overline{\beta}} F][\overline{\alpha} := \overline{\beta}] F) (\text{Nat}^{\overline{\beta}} (\mu\phi.\lambda\overline{\alpha}.H)\overline{\beta} F)$ Since Φ is empty, to see that

$$\llbracket \Gamma; \emptyset \mid \emptyset \vdash \text{fold}_H^F : \text{Nat}^{\overline{\beta}} (\text{Nat}^{\overline{\beta}} H[\phi :=_{\overline{\beta}} F][\overline{\alpha} := \overline{\beta}] F) (\text{Nat}^{\overline{\beta}} (\mu\phi.\lambda\overline{\alpha}.H)\overline{\beta} F) \rrbracket^{\text{Set}}$$

is a natural transformation from $\llbracket \Gamma; \emptyset \vdash \emptyset \rrbracket^{\text{Set}}$ to

$$\llbracket \Gamma; \emptyset \vdash \text{Nat}^{\overline{\beta}} (\text{Nat}^{\overline{\beta}} H[\phi :=_{\overline{\beta}} F][\overline{\alpha} := \overline{\beta}] F) (\text{Nat}^{\overline{\beta}} (\mu\phi.\lambda\overline{\alpha}.H)\overline{\beta} F) \rrbracket^{\text{Set}}$$

we need only show that, for all $\rho : \text{SetEnv}$, the unique $d : \llbracket \Gamma; \emptyset \vdash \emptyset \rrbracket^{\text{Set}} \rho$, and all $\eta : \llbracket \Gamma; \emptyset \vdash \text{Nat}^{\overline{\beta}} H[\phi :=_{\overline{\beta}} F][\overline{\alpha} := \overline{\beta}] F \rrbracket^{\text{Set}} \rho$,

$$\llbracket \Gamma; \emptyset \mid \emptyset \vdash \text{fold}_H^F : \text{Nat}^{\overline{\beta}} (\text{Nat}^{\overline{\beta}} H[\phi :=_{\overline{\beta}} F][\overline{\alpha} := \overline{\beta}] F) (\text{Nat}^{\overline{\beta}} (\mu\phi.\lambda\overline{\alpha}.H)\overline{\beta} F) \rrbracket^{\text{Set}} \rho d \eta$$

has type $\llbracket \Gamma; \emptyset \vdash \text{Nat}^{\overline{\beta}} (\mu\phi.\lambda\overline{\alpha}.H)\overline{\beta} F \rrbracket^{\text{Set}} \rho$, i.e., for any \overline{B} ,

$$(\llbracket \Gamma; \emptyset \mid \emptyset \vdash \text{fold}_H^F : \text{Nat}^{\overline{\beta}} (\text{Nat}^{\overline{\beta}} H[\phi :=_{\overline{\beta}} F][\overline{\alpha} := \overline{\beta}] F) (\text{Nat}^{\overline{\beta}} (\mu\phi.\lambda\overline{\alpha}.H)\overline{\beta} F) \rrbracket^{\text{Set}} \rho d \eta)_{\overline{B}}$$

is a morphism from $\llbracket \Gamma; \overline{\beta} \vdash (\mu\phi.\lambda\overline{\alpha}.H)\overline{\beta} \rrbracket^{\text{Set}} \rho[\overline{\beta} := \overline{B}] = (\mu T_{H,\rho}^{\text{Set}})_{\overline{B}}$ to $\llbracket \Gamma; \overline{\beta} \vdash F \rrbracket^{\text{Set}} \rho[\overline{\beta} := \overline{B}]$. To see this, note that η is a natural transformation from

$$\begin{aligned}
& \lambda \overline{B}. \llbracket \Gamma; \overline{\beta} \vdash H[\phi := F][\overline{\alpha} := \overline{\beta}] \rrbracket^{\text{Set}} \rho[\overline{\beta} := \overline{B}] \\
& = \lambda \overline{B}. T_{H,\rho}^{\text{Set}} (\lambda \overline{A}. \llbracket \Gamma; \overline{\beta} \vdash F \rrbracket^{\text{Set}} \rho[\overline{\beta} := \overline{A}]) \overline{B}
\end{aligned}$$

to

$$\begin{aligned}
& \lambda \overline{B}. (\lambda \overline{A}. \llbracket \Gamma; \overline{\beta} \vdash F \rrbracket^{\text{Set}} \rho[\overline{\beta} := \overline{A}]) \overline{B} \\
& = \lambda \overline{B}. \llbracket \Gamma; \overline{\beta} \vdash F \rrbracket^{\text{Set}} \rho[\overline{\beta} := \overline{B}]
\end{aligned}$$

and thus for each \overline{B} ,

$$(\llbracket \Gamma; \emptyset \mid \emptyset \vdash \text{fold}_H^F : \text{Nat}^\emptyset (\text{Nat}^{\overline{\beta}} H[\phi :=_{\overline{\beta}} F][\overline{\alpha} := \overline{\beta}] F) (\text{Nat}^{\overline{\beta}} (\mu\phi.\lambda\overline{\alpha}.H)\overline{\beta} F) \rrbracket^{\text{Set}} \rho d \eta)_{\overline{B}}$$

is a morphism from $\llbracket \Gamma; \overline{\beta} \vdash (\mu\phi.\lambda\overline{\alpha}.H)\overline{\beta} \rrbracket^{\text{Set}} \rho[\overline{\beta} := \overline{B}] = (\mu T_{H,\rho}^{\text{Set}})_{\overline{B}}$ to $\llbracket \Gamma; \overline{\beta} \vdash F \rrbracket^{\text{Set}} \rho[\overline{\beta} := \overline{B}]$. To see that this family of morphisms is natural in \overline{B} , we observe that the following diagram commutes for all $\overline{f} : \overline{B} \rightarrow \overline{B}'$:

$$\begin{array}{ccc} (\mu T_{H,\rho}^{\text{Set}})_{\overline{B}} & \xrightarrow{(fold_{T_{H,\rho}^{\text{Set}}} \eta)_{\overline{B}}} & \llbracket \Gamma; \overline{\beta} \vdash F \rrbracket^{\text{Set}} \rho[\overline{\beta} := \overline{B}] \\ \downarrow (\mu T_{H,\rho}^{\text{Set}})_{\overline{f}} & & \downarrow \llbracket \Gamma; \overline{\beta} \vdash F \rrbracket^{\text{Set}} id_{\rho[\overline{\beta} := \overline{f}]} \\ (\mu T_{H,\rho}^{\text{Set}})_{\overline{B}'} & \xrightarrow{(fold_{T_{H,\rho}^{\text{Set}}} \eta)_{\overline{B}'}} & \llbracket \Gamma; \overline{\beta} \vdash F \rrbracket^{\text{Set}} \rho[\overline{\beta} := \overline{B}'] \end{array}$$

by naturality of $fold_{T_{H,\rho}^{\text{Set}}} \eta$. To see that, for all $\rho : \text{SetEnv}$, $d \in \llbracket \Gamma; \emptyset \mid \emptyset \rrbracket^{\text{Set}} \rho$, and $\eta : \llbracket \Gamma; \emptyset \vdash \text{Nat}^{\overline{\beta}} H[\phi :=_{\overline{\beta}} F][\overline{\alpha} := \overline{\beta}] F \rrbracket^{\text{Set}} \rho$

$$\llbracket \Gamma; \emptyset \mid \emptyset \vdash \text{fold}_H^F : \text{Nat}^\emptyset (\text{Nat}^{\overline{\beta}} H[\phi :=_{\overline{\beta}} F][\overline{\alpha} := \overline{\beta}] F) (\text{Nat}^{\overline{\beta}} (\mu\phi.\lambda\overline{\alpha}.H)\overline{\beta} F) \rrbracket^{\text{Set}} \rho d \eta$$

satisfies the additional condition needed for it to be in $\llbracket \Gamma; \emptyset \vdash \text{Nat}^{\overline{\beta}} (\mu\phi.\lambda\overline{\alpha}.H)\overline{\beta} F \rrbracket^{\text{Set}} \rho$, let

$R : \text{Rel}(\overline{B}, \overline{B}')$. Since η satisfies the additional condition needed for it to be in $\llbracket \Gamma; \emptyset \vdash \text{Nat}^{\overline{\beta}} (H[\phi := F][\overline{\alpha} := \overline{\beta}]) F \rrbracket^{\text{Set}} \rho$,

$$((fold_{T_{H,\rho}^{\text{Set}}} \eta)_{\overline{B}}, (fold_{T_{H,\rho}^{\text{Set}}} \eta)_{\overline{B}'})$$

has type

$$\begin{aligned} & (\mu T_{H,\text{Eq}_\rho})_{\overline{R}} \rightarrow \llbracket \Gamma; \overline{\beta} \vdash F \rrbracket^{\text{Rel}} \text{Eq}_\rho[\overline{\beta} := \overline{R}] \\ &= (\mu T_{H,\text{Eq}_\rho})_{\llbracket \Gamma; \overline{\beta} \vdash \beta \rrbracket^{\text{Rel}} \text{Eq}_\rho[\overline{\beta} := \overline{R}]} \rightarrow \llbracket \Gamma; \overline{\beta} \vdash F \rrbracket^{\text{Rel}} \text{Eq}_\rho[\overline{\beta} := \overline{R}] \\ &= \llbracket \Gamma; \overline{\beta} \vdash (\mu\phi.\lambda\overline{\alpha}.H)\overline{\beta} \rrbracket^{\text{Rel}} \text{Eq}_\rho[\overline{\beta} := \overline{R}] \rightarrow \llbracket \Gamma; \overline{\beta} \vdash F \rrbracket^{\text{Rel}} \text{Eq}_\rho[\overline{\beta} := \overline{R}] \end{aligned}$$

Finally, since $\Phi = \emptyset$, naturality of

$$\llbracket \Gamma; \emptyset \mid \emptyset \vdash \text{fold}_H^F : \text{Nat}^\emptyset (\text{Nat}^{\overline{\beta}} H[\phi :=_{\overline{\beta}} F][\overline{\alpha} := \overline{\beta}] F) (\text{Nat}^{\overline{\beta}} (\mu\phi.\lambda\overline{\alpha}.H)\overline{\beta} F) \rrbracket^{\text{Set}} \rho$$

in ρ is trivial. □

The following theorem is an immediate consequence of Theorem 6.2:

Theorem 6.3. *If $\Gamma; \Phi \mid \Delta \vdash t : F$ and $\rho \in \text{RelEnv}$, and if $(a, b) \in \llbracket \Gamma; \Phi \vdash \Delta \rrbracket^{\text{Rel}} \rho$, then $(\llbracket \Gamma; \Phi \mid \Delta \vdash t : F \rrbracket^{\text{Set}} (\pi_1 \rho) a, \llbracket \Gamma; \Phi \mid \Delta \vdash t : F \rrbracket^{\text{Set}} (\pi_2 \rho) b) \in \llbracket \Gamma; \Phi \vdash F \rrbracket^{\text{Rel}} \rho$.*

Finally, the Abstraction Theorem is the instantiation of Theorem 6.3 to closed terms of closed type:

Theorem 6.4 (Abstraction Theorem). *If $\vdash t : F$, then $(\llbracket \vdash t : F \rrbracket^{\text{Set}}, \llbracket \vdash t : F \rrbracket^{\text{Set}}) \in \llbracket \vdash F \rrbracket^{\text{Rel}}$.*

7. FREE THEOREMS FOR NESTED TYPES

In this section we show how Theorem 6.2 and its consequences can be used to prove free theorems that go beyond mere naturality. We also show that we can extend short cut short cut fusion for lists [GLP93] to nested types, thereby formally proving correctness of the categorically inspired theorem from [JG10].

7.1. Free Theorem for Type of Polymorphic Bottom. Let $\vdash g : \text{Nat}^\alpha \perp \alpha$, let $G^{\text{Set}} = \llbracket \vdash g : \text{Nat}^\alpha \perp \alpha \rrbracket^{\text{Set}}$ and $G^{\text{Rel}} = \llbracket \vdash g : \text{Nat}^\alpha \perp \alpha \rrbracket^{\text{Rel}}$. By Theorem 6.2, $(G^{\text{Set}}(\pi_1\rho), G^{\text{Set}}(\pi_2\rho)) = G^{\text{Rel}}\rho$. Thus, for all $\rho \in \text{RelEnv}$ and any $(a, b) \in \llbracket \vdash \emptyset \rrbracket^{\text{Rel}}\rho = 1$, eliding the only possible instantiations of a and b gives that $(G^{\text{Set}}, G^{\text{Set}}) = (G^{\text{Set}}(\pi_1\rho), G^{\text{Set}}(\pi_2\rho)) \in \llbracket \vdash \text{Nat}^\alpha \perp \alpha \rrbracket^{\text{Rel}}\rho = \{\eta : K_1 \Rightarrow id\} = \{(\eta_1 : K_1 \Rightarrow id, \eta_2 : K_1 \Rightarrow id)\}$. That is, G^{Set} is a natural transformation from the constantly 1-valued functor to the identity functor in Set . In particular, for every $S : \text{Set}$, $G_S^{\text{Set}} : 1 \rightarrow S$. Note, however, that if $S = \emptyset$, then there can be no such morphism, so no such natural transformation, and thus no term $\vdash g : \text{Nat}^\alpha \perp \alpha$, can exist. That is, our calculus admits no (non-terminating) terms with the closed type $\text{Nat}^\alpha \perp \alpha$ of the polymorphic bottom.

7.2. Free Theorem for Type of Polymorphic Identity. Let $\vdash g : \text{Nat}^\alpha \alpha \alpha$, $G^{\text{Set}} = \llbracket \vdash g : \text{Nat}^\alpha \alpha \alpha \rrbracket^{\text{Set}}$, and $G^{\text{Rel}} = \llbracket \vdash g : \text{Nat}^\alpha \alpha \alpha \rrbracket^{\text{Rel}}$. By Theorem 6.2, $(G^{\text{Set}}(\pi_1\rho), G^{\text{Set}}(\pi_2\rho)) = G^{\text{Rel}}\rho$. Thus, for all $\rho \in \text{RelEnv}$ and any $(a, b) \in \llbracket \vdash \emptyset \rrbracket^{\text{Rel}}\rho = 1$, eliding the only possible instantiations of a and b gives that $(G^{\text{Set}}, G^{\text{Set}}) = (G^{\text{Set}}(\pi_1\rho), G^{\text{Set}}(\pi_2\rho)) \in \llbracket \vdash \text{Nat}^\alpha \alpha \alpha \rrbracket^{\text{Rel}}\rho = \{\eta : id \Rightarrow id\} = \{(\eta_1 : id \Rightarrow id, \eta_2 : id \Rightarrow id)\}$. That is, G^{Set} is a natural transformation from the identity functor on Set to itself. Now let S be any set. For any element a of S , if $K_a : S \rightarrow S$ is the constantly a -valued morphism on S then instantiating the naturality square implied by the above equality gives that $G_S^{\text{Set}} \circ K_a = K_a \circ G_S^{\text{Set}}$, i.e., $G_S^{\text{Set}} a = a$, i.e., $G_S^{\text{Set}} = id_S$. Putting these two cases together we have that for every $S : \text{Set}$, $G_S^{\text{Set}} = id_S$, i.e., G^{Set} is the identity natural transformation for the identity functor on Set . So every closed term g of closed type $\text{Nat}^\alpha \alpha \alpha$ always denotes the identity natural transformation for the identity functor on Set , i.e., every closed term g of type $\text{Nat}^\alpha \alpha \alpha$ denotes the polymorphic identity function.

7.3. Standard Free Theorems for ADTs and Nested Types. We can derive in our calculus even those free theorems for polymorphic functions over ADTs that are not consequences of naturality. We can, e.g., prove the free theorem for *filter*'s type as follows:

Theorem 7.1. *If $g : A \rightarrow B$, $\rho \in \text{RelEnv}$, $\rho\alpha = (A, B, \langle g \rangle)$, $(a, b) \in \llbracket \alpha; \emptyset \vdash \Delta \rrbracket^{\text{Rel}}\rho$, $(s \circ g, s) \in \llbracket \alpha; \emptyset \vdash \text{Nat}^\emptyset \alpha \text{Bool} \rrbracket^{\text{Rel}}\rho$, and*

$$\text{filter} = \llbracket \alpha; \emptyset \mid \Delta \vdash t : \text{Nat}^\emptyset(\text{Nat}^\emptyset \alpha \text{Bool})(\text{Nat}^\emptyset(\text{List } \alpha)(\text{List } \alpha)) \rrbracket^{\text{Set}}$$

for some t , then

$$\text{map}_{\text{List}} g \circ \text{filter}(\pi_1\rho) a (s \circ g) = \text{filter}(\pi_2\rho) b s \circ \text{map}_{\text{List}} g$$

Proof. By Theorem 6.4,

$$(\text{filter}(\pi_1\rho) a, \text{filter}(\pi_2\rho) b) \in \llbracket \alpha; \emptyset \vdash \text{Nat}^\emptyset(\text{Nat}^\emptyset \alpha \text{Bool})(\text{Nat}^\emptyset(\text{List } \alpha)(\text{List } \alpha)) \rrbracket^{\text{Rel}}\rho$$

so if $(s', s) \in \llbracket \alpha; \emptyset \vdash \text{Nat}^\emptyset \alpha \text{Bool} \rrbracket^{\text{Rel}}\rho = \rho\alpha \rightarrow \text{Eq}_{\text{Bool}}$ and $(xs', xs) \in \llbracket \alpha; \emptyset \vdash \text{List } \alpha \rrbracket^{\text{Rel}}\rho$ then

$$(\text{filter}(\pi_1\rho) a s' xs', \text{filter}(\pi_2\rho) b s xs) \in \llbracket \alpha; \emptyset \vdash \text{List } \alpha \rrbracket^{\text{Rel}}\rho \quad (7.1)$$

If $\rho\alpha = (A, B, \langle g \rangle)$, then $\llbracket \alpha; \emptyset \vdash \text{List } \alpha \rrbracket^{\text{Rel}}\rho = \langle \text{map}_{\text{List}} g \rangle$ by Lemma 4.5 and demotion. Moreover, $xs = \text{map}_{\text{List}} g xs'$ and $(s', s) \in \langle g \rangle \rightarrow \text{Eq}_{\text{Bool}}$, so $s' = s \circ g$. The result follows from Equation 7.1. \square

A similar proof establishes the analogous result for, say, generalized rose trees.

Theorem 7.2. *If $g : A \rightarrow B$, $F, G : \text{Set} \rightarrow \text{Set}$, $\eta : F \rightarrow G$ in Set , $\rho \in \text{RelEnv}$, $\rho\alpha = (A, B, \langle g \rangle)$, $\rho\psi = (F, G, \langle \eta \rangle)$, $(a, b) \in \llbracket \alpha; \psi; \emptyset \vdash \Delta \rrbracket^{\text{Rel}}\rho$, $(s \circ g, s) \in \llbracket \alpha; \emptyset \vdash \text{Nat}^\emptyset \alpha \text{Bool} \rrbracket^{\text{Rel}}\rho$, and*

$$\text{filter} = \llbracket \alpha; \psi; \emptyset \mid \Delta \vdash t : \text{Nat}^\emptyset(\text{Nat}^\emptyset \alpha \text{Bool})(\text{Nat}^\emptyset(G\text{Rose } \psi \alpha)(G\text{Rose } \psi(\alpha + \perp))) \rrbracket^{\text{Set}}$$

for some t , then

$$\text{map}_{G\text{Rose } \eta} (g + 1) \circ \text{filter}(\pi_1\rho) a (s \circ g) = \text{filter}(\pi_2\rho) b s \circ \text{map}_{G\text{Rose } \eta} g$$

This is not surprising since rose trees are essentially ADT-like. However, as noted in Section 2.2, our calculus cannot express the type of a polymorphic filter function for a proper nested type.

7.4. Short Cut Fusion for Lists. We can recover standard short cut fusion for lists [GLP93] in our calculus:

Theorem 7.3. *If $\vdash F, \vdash H$, and $G = \llbracket \beta; \emptyset \mid \emptyset \vdash g : \text{Nat}^\emptyset(\text{Nat}^\emptyset(\mathbb{1} + F \times \beta) \beta) \beta \rrbracket^{\text{Set}}$ for some g , and if $c \in \llbracket \vdash F \rrbracket^{\text{Set}} \times \llbracket \vdash H \rrbracket^{\text{Set}} \rightarrow \llbracket \vdash H \rrbracket^{\text{Set}}$ and $n \in \llbracket \vdash H \rrbracket^{\text{Set}}$, then*

$$\text{fold}_{1+\llbracket \vdash F \rrbracket^{\text{Set}} \times _} n c (G (\text{List } \llbracket \vdash F \rrbracket^{\text{Set}}) \text{nil cons}) = G \llbracket \vdash H \rrbracket^{\text{Set}} n c$$

Proof. Theorem 6.4 gives that, for any $\rho \in \text{RelEnv}$,

$$\begin{aligned} (G (\pi_1 \rho), G (\pi_2 \rho)) &\in \llbracket \beta; \emptyset \mid \text{Nat}^\emptyset(\text{Nat}^\emptyset(\mathbb{1} + F \times \beta) \beta) \beta \rrbracket^{\text{Rel}} \rho \\ &\cong (((\llbracket \vdash F \rrbracket^{\text{Rel}} \rho \times \rho \beta) \rightarrow \rho \beta) \times \rho \beta) \rightarrow \rho \beta \end{aligned}$$

so if $(c', c) \in \llbracket \vdash F \rrbracket^{\text{Rel}} \rho \times \rho \beta \rightarrow \rho \beta$ and $(n', n) \in \rho \beta$ then $(G (\pi_1 \rho) n' c', G (\pi_2 \rho) n c) \in \rho \beta$. In addition,

$$\begin{aligned} &\llbracket \vdash \text{fold}_{1+F \times \beta}^H : \text{Nat}^\emptyset(\text{Nat}^\emptyset(\mathbb{1} + F \times H) H) (\text{Nat}^\emptyset(\mu \alpha. \mathbb{1} + F \times \alpha) H) \rrbracket^{\text{Set}} \\ &= \text{fold}_{1+\llbracket \vdash F \rrbracket^{\text{Set}} \times _} \end{aligned}$$

so that if $c \in \llbracket \vdash F \rrbracket^{\text{Set}} \times \llbracket \vdash H \rrbracket^{\text{Set}} \rightarrow \llbracket \vdash H \rrbracket^{\text{Set}}$ and $n \in \llbracket \vdash H \rrbracket^{\text{Set}}$, then

$$(n, c) \in \llbracket \vdash \text{Nat}^\emptyset(\mathbb{1} + F \times H) H \rrbracket^{\text{Set}}$$

The instantiation

$$\begin{aligned} \pi_1 \rho \beta &= \llbracket \vdash \mu \alpha. \mathbb{1} + F \times \alpha \rrbracket^{\text{Set}} = \text{List } \llbracket \vdash F \rrbracket^{\text{Set}} \\ \pi_2 \rho \beta &= \llbracket \vdash H \rrbracket^{\text{Set}} \\ \rho \beta &= \langle \text{fold}_{1+\llbracket \vdash F \rrbracket^{\text{Set}} \times _} n c \rangle : \text{Rel}(\pi_1 \rho \beta, \pi_2 \rho \beta) \\ c' &= \text{cons} \\ n' &= \text{nil} \end{aligned}$$

thus gives that $(G (\text{List } \llbracket \vdash F \rrbracket^{\text{Set}}) \text{nil cons}, G \llbracket \vdash H \rrbracket^{\text{Set}} n c) \in \langle \text{fold}_{1+\llbracket \vdash F \rrbracket^{\text{Set}} \times _} n c \rangle$, i.e., that $\text{fold}_{1+\llbracket \vdash F \rrbracket^{\text{Set}} \times _} n c (G (\text{List } \llbracket \vdash F \rrbracket^{\text{Set}}) \text{nil cons}, G \llbracket \vdash H \rrbracket^{\text{Set}} n c)$. \square

We can extend short cut fusion results to arbitrary ADTs, as in [Joh02, Pit98].

7.5. Short Cut Fusion for Arbitrary Nested Types. We can extend short cut fusion for lists [GLP93] to nested types, thereby formally prove correctness of the categorically inspired theorem from [JG10]. We have:

Theorem 7.4. *If $\emptyset; \phi, \alpha \vdash F$, $\emptyset; \alpha \vdash K$, $H : [\text{Set}, \text{Set}] \rightarrow [\text{Set}, \text{Set}]$ is defined by*

$$H f x = \llbracket \emptyset; \phi, \alpha \vdash F \rrbracket^{\text{Set}} [\phi := f][\alpha := x]$$

and

$$G = \llbracket \phi; \emptyset \mid \emptyset \vdash g : \text{Nat}^\emptyset(\text{Nat}^\alpha F(\phi \alpha)) (\text{Nat}^\alpha \mathbb{1}(\phi \alpha)) \rrbracket^{\text{Set}}$$

for some g , then, for every $B \in H \llbracket \emptyset; \alpha \vdash K \rrbracket^{\text{Set}} \rightarrow \llbracket \emptyset; \alpha \vdash K \rrbracket^{\text{Set}}$,

$$\text{fold}_H B (G \mu H \text{in}_H) = G \llbracket \emptyset; \alpha \vdash K \rrbracket^{\text{Set}} B$$

Proof. Theorem 6.4 gives that, for any $\rho \in \text{RelEnv}$,

$$\begin{aligned} (G (\pi_1 \rho), G (\pi_2 \rho)) &\in \llbracket \phi; \emptyset \mid \text{Nat}^\emptyset(\text{Nat}^\alpha F(\phi \alpha)) (\text{Nat}^\alpha \mathbb{1}(\phi \alpha)) \rrbracket^{\text{Rel}} \rho \\ &= \llbracket \phi; \emptyset \mid \text{Nat}^\alpha F(\phi \alpha) \rrbracket^{\text{Rel}} \rho \rightarrow \llbracket \phi; \emptyset \mid \text{Nat}^\alpha \mathbb{1}(\phi \alpha) \rrbracket^{\text{Rel}} \rho \\ &= \llbracket \phi; \emptyset \mid \text{Nat}^\alpha F(\phi \alpha) \rrbracket^{\text{Rel}} \rho \rightarrow \rho \phi \end{aligned}$$

so if $(A, B) \in \llbracket \phi; \emptyset \mid \text{Nat}^\alpha F(\phi \alpha) \rrbracket^{\text{Rel}} \rho$ then $(G (\pi_1 \rho) A, G (\pi_2 \rho) B) \in \rho \phi$. Also,

$$\llbracket \vdash \text{fold}_F^K : \text{Nat}^\emptyset(\text{Nat}^\alpha F[\phi := K] K) (\text{Nat}^\alpha((\mu \phi. \lambda \alpha. F) \alpha) K) \rrbracket^{\text{Set}} = \text{fold}_H$$

Now let $A = in_H : H(\mu H) \Rightarrow \mu H$, $B : H[\emptyset; \alpha \vdash K]^{\text{Set}} \Rightarrow [\emptyset; \alpha \vdash K]^{\text{Set}}$, $\rho\phi = \langle fold_H B \rangle$, $\pi_1\rho\phi = \mu H$, $\pi_2\rho\phi = [\emptyset; \alpha \vdash K]^{\text{Set}}$, $\rho\phi : \text{Rel}(\pi_1\rho\phi, \pi_2\rho\phi)$, $A : [\emptyset; \emptyset \vdash \text{Nat}^\alpha F(\phi\alpha)]^{\text{Set}}(\pi_1\rho)$, and $B : [\emptyset; \emptyset \vdash \text{Nat}^\alpha F(\phi\alpha)]^{\text{Set}}(\pi_2\rho)$. Demotion ensures that $A = in_H : H(\mu H) \Rightarrow \mu H = [\emptyset; \emptyset \vdash \text{Nat}^\alpha F(\phi\alpha)]^{\text{Set}}(\pi_1\rho)$, and demotion and Lemma 4.5 together give that

$$\begin{aligned} (A, B) = (in_H, B) &\in [\emptyset; \emptyset \vdash \text{Nat}^\alpha F(\phi\alpha)]^{\text{Rel}} \rho \\ &= \lambda A. [\emptyset; \alpha \vdash F]^{\text{Rel}}[\phi := \langle fold_H B \rangle][\alpha := A] \Rightarrow \langle fold_H B \rangle \\ &= [\emptyset; \phi, \alpha \vdash F]^{\text{Rel}} \langle fold_H B \rangle \Rightarrow \langle fold_H B \rangle \\ &= \langle [\emptyset; \phi, \alpha \vdash F]^{\text{Set}} (fold_H B) \rangle \Rightarrow \langle fold_H B \rangle \\ &= \langle map_H (fold_H B) \rangle \Rightarrow \langle fold_H B \rangle \end{aligned}$$

since if $(x, y) \in \langle map_H (fold_H B) \rangle$, then

$$fold_H B (in_H x) = B y = B (map_H (fold_H B) x)$$

by the definition of $fold_H$ as a (indeed, the unique) morphism from in_H to B . Thus, $(G(\pi_1\rho) A, G(\pi_2\rho) B) \in \langle fold_H B \rangle$, i.e., $fold_H B (G(\pi_1\rho) in_H) = G(\pi_2\rho) B$. But since ϕ is the only free variable in G , this simplifies to $fold_H B (G \mu H in_H) = G [\emptyset; \alpha \vdash K]^{\text{Set}} B$. \square

As in [JG10], replacing $\mathbb{1}$ with any type $\emptyset; \alpha \vdash C$ generalizes Theorem 7.4 to deliver a more general free theorem whose conclusion is $fold_H B \circ G \mu H in_H = G [\emptyset; \alpha \vdash K]^{\text{Set}} B$.

Although it is standard to prove that the parametric model constructed verifies the existence of initial algebras, this is unnecessary here since initial algebras are built directly into our model.

8. PARAMETRICITY FOR GADTs

As discussed in Section 1, type indices for nested types can be any types, including, in the case of truly nested types like that of bushes, types involving the very same nested type that is being defined. But every data constructor for a nested type must still have as its return type exactly the instance being defined. For example, the data constructors for the instance **PTree** **A** of the nested type **PTree** are **pleaf** :: **PTree** **A** and **pnode** :: **PTree** (**A** \times **A**) \rightarrow **PTree** **A**, and the data constructors for the instance **Bush** **A** of the truly nested type **Bush** are **bnil** :: **Bush** **A** and **bcons** :: **A** \rightarrow **Bush** (**Bush** **A**) \rightarrow **Bush** **A**.

Generalized algebraic datatypes (GADTs) — also known as guarded recursive data types or first-class phantom types — generalize nested types by relaxing the above restriction to allow the return types of data constructors to be different instances of the datatype than the one being defined. For example, the GADT

```
data Seq (A : Set) : Set where
  sconst : A  $\rightarrow$  Seq A
  spair  : Seq A  $\rightarrow$  Seq B  $\rightarrow$  Seq (A  $\times$  B)
  sseq   : (Nat  $\rightarrow$  Seq A)  $\rightarrow$  Seq (Nat  $\rightarrow$  A)
```

has data constructors **spair** and **sseq** with return types **Seq** (**A** \times **B**) and **Seq** (**Nat** \rightarrow **A**). These types are not only at different instances of **Seq** from the instance **Seq** **A** being defined, but also at different instances from one another. The resulting interdependence of different instances of GADTs means that they can express more constraints than ADTs and nested types. For example, the ADT **List** expresses the invariant that all of the data in the lists it defines is of the same type, while the nested type **PTree** expresses this invariant *as well as* the invariant that all of the lists it defines have lengths that are powers of 2. The GADT **Seq** enforces *even more general* well-formedness conditions for sequences of values that simply cannot be expressed with ADTs and nested types alone.

GADTs are widely used in modern functional languages, such as Haskell, as well as in proof assistants, such as Agda, that are based on dependent type theories. A natural next step in the line of work reported in this paper is therefore to extend our parametricity results to GADTs. A promising starting point for this endeavor is the observation from [JG08] that the data objects of GADTs can be represented using object-level left Kan extensions over discrete categories. The more recent results

of [JP19] further show that adding a carefully designed object-level left Kan extension construct to a calculus supporting primitive nested types preserves the cocontinuity needed for primitive GADTs to have well-defined, properly functorial interpretations. Together this suggests extending the type system in Definition 2.1 with such a left Kan extension construct, and extending the calculus in Figure 1 with corresponding categorically inspired constructs to introduce and eliminate terms of these richer types. This approach exactly mirrors the (entirely standard) approach taken above for product, coproduct, and fixpoint types. In this section we outline this approach and show how our model from Section 5 can be extended to a parametric model when some classes of primitive GADTs are incorporated in this manner. However, as we argue at the end of this section, such extension is likely not possible.

8.1. Left Kan Extensions. We begin by recalling the definition of a left Kan extension and establishing some useful notation and results for them.

Definition 8.1. If $F : \mathbf{Set}^k \rightarrow \mathbf{Set}$ and $\bar{K} : \mathbf{Set}^k \rightarrow \mathbf{Set}^h$ are functors over \mathbf{Set} , then the *left Kan extension of F along \bar{K}* is a functor $\text{Lan}_{\bar{K}} F : \mathbf{Set}^h \rightarrow \mathbf{Set}$ together with a natural transformation $\eta : F \rightarrow (\text{Lan}_{\bar{K}} F) \circ \bar{K}$ such that, for every functor $G : \mathbf{Set}^h \rightarrow \mathbf{Set}$ and natural transformation $\gamma : F \rightarrow G \circ \bar{K}$, there exists a unique natural transformation $\mu : \text{Lan}_{\bar{K}} F \rightarrow G$ such that $(\mu \bar{K}) \circ \eta = \gamma$. This is depicted in the following diagram:

Replacing \mathbf{Set} by \mathbf{Rel} everywhere in Definition 8.1 we can similarly define the left Kan extension of F along \bar{K} for functors F and \bar{K} over \mathbf{Rel} .

An alternative presentation characterizes the left Kan extension $(\text{Lan}_{\bar{K}} F, \eta)$ in terms of the bijection between natural transformations from F to $G \circ \bar{K}$ and natural transformations from $\text{Lan}_{\bar{K}} F$ to G , for which η is the unit. If Agda were to support a primitive `Lan` for left Kan extensions, we could use this bijection to rewrite the type of each data constructor for the GADT `Seq` to arrive at the following equivalent representation:

```
data Seq (A : Set) : Set where
  sconst : A → Seq A
  spair  : (LanλC.D. C × D (λC.D. Seq C × Seq D)) A → Seq A
  sseq   : (LanλC. Nat → C (λC. Nat → Seq C)) A → Seq A
```

Our calculus will represent `Seq` and other GADTs in precisely this way.

A third representation of left Kan extensions in locally presentable categories is given in terms of colimits. Writing \mathcal{C}_0 for the full subcategory of finitely presentable objects in the locally presentable category \mathcal{C} , the left Kan extension can be expressed as

$$(\text{Lan}_{\bar{K}} F) \bar{A} = \varinjlim_{X : \mathcal{C}_0, f : K \bar{X} \rightarrow A} F \bar{X}$$

Thus, if $F : \mathbf{Set}^k \rightarrow \mathbf{Set}$, $\bar{K} : \mathbf{Set}^k \rightarrow \mathbf{Set}^h$, $\bar{A} : \mathbf{Set}^h$, and \mathbf{Set}_0 is the full subcategory of finitely presentable objects in \mathbf{Set} , — i.e., is the category of finite sets — we have that

$$(\text{Lan}_{\bar{K}} F) \bar{A} = \varinjlim_{\bar{S} : \mathbf{Set}_0^k, f : K \bar{S} \rightarrow A} F \bar{S} \quad (8.1)$$

If $\bar{S} : \mathbf{Set}_0^k$ and $f : K \bar{S} \rightarrow A$, let $j_{\bar{S}, \bar{f}} : F \bar{S} \rightarrow (\text{Lan}_{\bar{K}} F) \bar{A}$ be the morphism indexed by \bar{S} and \bar{f} mapping the cocone into the colimit in Equation 8.1. For any h -tuple of functions $\bar{g} : A \rightarrow \bar{B}$, the

functorial action $(\text{Lan}_{\overline{K}}F)\overline{g}$ is the unique function from $(\text{Lan}_{\overline{K}}F)\overline{A}$ to $(\text{Lan}_{\overline{K}}F)\overline{B}$ such that, for all $\overline{S} : \text{Set}_0^k$ and $f : K\overline{S} \rightarrow A$,

$$(\text{Lan}_{\overline{K}}F)\overline{g} \circ j_{\overline{S}, \overline{f}} = j'_{\overline{S}, \overline{g \circ f}} : F\overline{S} \rightarrow (\text{Lan}_{\overline{K}}F)\overline{B} \quad (8.2)$$

holds, where j is the morphism mapping the cocone into $(\text{Lan}_{\overline{K}}F)\overline{A} = \varinjlim_{\overline{S} : \text{Set}_0^k, f : K\overline{S} \rightarrow A} F\overline{S}$ and j' is the morphism mapping the cocone into $(\text{Lan}_{\overline{K}}F)\overline{B} = \varinjlim_{\overline{S} : \text{Set}_0^k, f : K\overline{S} \rightarrow B} F\overline{S}$. Moreover, if $\alpha : F \rightarrow F'$ is a natural transformation, then $\text{Lan}_{\overline{K}}\alpha : (\text{Lan}_{\overline{K}}F) \rightarrow (\text{Lan}_{\overline{K}}F')$ is defined to be the induced natural transformation whose component $(\text{Lan}_{\overline{K}}\alpha)\overline{A} : (\text{Lan}_{\overline{K}}F)\overline{A} \rightarrow (\text{Lan}_{\overline{K}}F')\overline{A}$ is the unique function such that $(\text{Lan}_{\overline{K}}\alpha)\overline{A} \circ j_{\overline{S}, \overline{f}} = j'_{\overline{S}, \overline{f}} \circ \alpha_{\overline{S}}$, where j is the cocone into $(\text{Lan}_{\overline{K}}F)\overline{A}$ and j' is the cocone into $(\text{Lan}_{\overline{K}}F')\overline{A}$. We can similarly represent left Kan extensions of functors over Rel in terms of colimits. In that setting we will denote the morphism mapping the cocone into the colimit ι . We will make good use of both of these representations in Section 8.4 below.

More about each of the above three representations of left Kan extensions and the connections between them can be found in, e.g., [Rie16].

8.2. Extending the Calculus. We incorporate GADTs into our calculus by first adding to Definition 2.1 the following type formation rule for **Lan**-types:

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

Here, the type constructor **Lan** binds the variables in $\overline{\alpha}$, and these variables must always have arity 0. In addition, the vectors \overline{K} and \overline{A} must have the same length.

Intuitively, $\text{Lan}_{\overline{K}}^{\overline{\alpha}^0} F$ is a syntactic representation of the left Kan extension of the functor in the variables in $\overline{\alpha}$ denoted by F along the functor in the variables in $\overline{\alpha}$ denoted by \overline{K} . Using **Lan**-types, we can therefore represent the GADT **Seq** in our calculus as

$$\text{Seq } \alpha := (\mu\phi.\lambda\beta.\beta + (\text{Lan}_{\gamma_1 \times \gamma_2}^{\gamma_1, \gamma_2}(\phi\gamma_1 \times \phi\gamma_2))\beta + (\text{Lan}_{\text{Nat} \rightarrow \gamma}^{\gamma}(\text{Nat} \rightarrow \phi\gamma))\beta)\alpha$$

where $\text{Nat} = \mu\alpha.1 + \alpha$ and $\text{Nat} \rightarrow C$ abbreviates $(\text{Lan}_{\text{Nat}}^{\emptyset} 1) C$, which equals $\varinjlim_{f : \text{Nat} \rightarrow C} 1$ by Equation 8.1 (here, $k = 0$). This representation of *Seq* corresponds exactly to the rewriting in Section 8.1. As explained in Section IV.D of [JP19], the more general construct $\text{Lan}_{\overline{K}}^{\overline{\alpha}^0} F$ allowing extensions along vectors of functors as depicted above makes it possible to represent GADTs with two or more type arguments that depend on one another. Such GADTs cannot be represented using just unary **Lan**-types, i.e. **Lan**-types of the form $(\text{Lan}_{\overline{K}}^{\overline{\alpha}^0} F)A$.

As we will see below, in order to define the term-formation rules for terms of **Lan**-types we also need to generalize **Nat**-types to bind type constructor variables not just of arity 0, but of arbitrary arity. Accordingly, we replace the rule for **Nat**-types in Definition 2.1 with

$$\frac{\Gamma; \Phi \vdash F \quad \Gamma; \Phi \vdash G}{\Gamma; \emptyset \vdash \text{Nat}^\Phi F G}$$

Of course, we could have performed this replacement from the outset of the present paper. But since using the above rule instead of the one currently in Definition 2.1 only makes the notation of our calculus heavier without increasing its expressivity, we have chosen not to.

Finally, adding **Lan**-types to the types of our calculus also requires the extension of Definition 2.2 with the following new clause:

$$((\text{Lan}_{\overline{K}}^{\overline{\alpha}} F)\overline{A})[\phi :=_{\overline{\beta}} G] = (\text{Lan}_{\overline{K}}^{\overline{\alpha}} F[\phi :=_{\overline{\beta}} G])\overline{A}[\phi :=_{\overline{\beta}} G]$$

We must also extend our term calculus to accommodate GADTs. To this end, we first give introduction and elimination rules appropriate to our generalized **Nat**-types, replacing the tenth and eleventh rules in Figure 1 with

$$\frac{\Gamma; \Phi \vdash F \quad \Gamma; \Phi \vdash G \quad \Gamma; \Phi \mid \Delta, x : F \vdash t : G}{\Gamma; \emptyset \mid \Delta \vdash L_{\Phi} x.t : \mathbf{Nat}^{\Phi} F G}$$

and

$$\frac{\overline{\Gamma; \Phi, \bar{\beta} \vdash K} \quad \Gamma; \emptyset \mid \Delta \vdash t : \mathbf{Nat}^{\bar{\psi}} F G \quad \Gamma; \Phi \mid \Delta \vdash s : F[\bar{\psi} := \bar{\beta} \bar{K}]}{\Gamma; \Phi \mid \Delta \vdash t_{\bar{K}} s : G[\bar{\psi} := \bar{\beta} \bar{K}]}$$

respectively. Next, we add to the term calculus in Figure 1 introduction and elimination rules for terms of **Lan**-types according to Definition 8.1. This gives the rules

$$\frac{\Gamma; \Phi, \bar{\alpha} \vdash F \quad \overline{\Gamma; \bar{\alpha} \vdash \bar{K}}}{\Gamma; \emptyset \mid \emptyset \vdash \int_{\bar{K}, F} : \mathbf{Nat}^{\Phi, \bar{\alpha}} F (\mathbf{Lan}_{\bar{K}}^{\bar{\alpha}} F) \bar{K}}$$

and

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

respectively. Note that both of these rules make essential use of our generalized **Nat**-types. If Φ were required to be empty then we would not be able to express, e.g., the types

$$\mathbf{Nat}^{\phi, \gamma, \delta} (\phi \gamma \times \phi \delta) (\phi (\gamma \times \phi))$$

and

$$\mathbf{Nat}^{\phi, \beta} (\mathbf{Lan}_{\gamma \times \delta}^{\gamma, \delta} (\phi \gamma \times \phi \delta)) \beta (\phi \beta)$$

associated with **Seq**'s data constructor **spair**.

8.3. Extending the Type Semantics. To construct a parametric model of the kind we seek, we must give suitable interpretations of our generalized **Nat**-types and **Lan**-types, as well as of terms of both such types, in both **Set** and **Rel**. Recalling that the type constructor variables bound by **Nat** in the formation rule for generalized **Nat**-types are interpreted in **Set** as functors and in **Rel** as relation transformers, we let k_i be the arity of the i^{th} element of the sequence Φ and extend the set and relational interpretations for **Nat** types from Section 3 as follows:

$$\begin{aligned} \llbracket \Gamma; \emptyset \vdash \mathbf{Nat}^{\Phi} F G \rrbracket^{\mathbf{Set}} \rho &= \{ \eta : \lambda \bar{K}. \llbracket \Gamma; \Phi \vdash F \rrbracket^{\mathbf{Set}} \rho[\Phi := \bar{K}] \Rightarrow \lambda \bar{K}. \llbracket \Gamma; \Phi \vdash G \rrbracket^{\mathbf{Set}} \rho[\Phi := \bar{K}] \mid \\ &\quad \forall \bar{K} = (K^1, K^2, K^*) : RT_k. \\ &\quad (\eta_{\bar{K}^1}, \eta_{\bar{K}^2}) : \llbracket \Gamma; \Phi \vdash F \rrbracket^{\mathbf{Rel}} \text{Eq}_{\rho}[\Phi := \bar{K}] \rightarrow \llbracket \Gamma; \Phi \vdash G \rrbracket^{\mathbf{Rel}} \text{Eq}_{\rho}[\Phi := \bar{K}] \} \\ \llbracket \Gamma; \emptyset \vdash \mathbf{Nat}^{\Phi} F G \rrbracket^{\mathbf{Rel}} \rho &= \{ \eta : \lambda \bar{K}. \llbracket \Gamma; \Phi \vdash F \rrbracket^{\mathbf{Rel}} \rho[\Phi := \bar{K}] \Rightarrow \lambda \bar{K}. \llbracket \Gamma; \Phi \vdash G \rrbracket^{\mathbf{Rel}} \rho[\Phi := \bar{K}] \} \\ &= \{ (\eta_1, \eta_2) \in \llbracket \Gamma; \emptyset \vdash \mathbf{Nat}^{\Phi} F G \rrbracket^{\mathbf{Set}} (\pi_1 \rho) \times \llbracket \Gamma; \emptyset \vdash \mathbf{Nat}^{\Phi} F G \rrbracket^{\mathbf{Set}} (\pi_2 \rho) \mid \\ &\quad \forall \bar{K} = (K^1, K^2, K^*) : RT_k. \\ &\quad ((\eta_1)_{\bar{K}^1}, (\eta_2)_{\bar{K}^2}) \in (\llbracket \Gamma; \Phi \vdash G \rrbracket^{\mathbf{Rel}} \rho[\Phi := \bar{K}]) \llbracket \Gamma; \Phi \vdash F \rrbracket^{\mathbf{Rel}} \rho[\Phi := \bar{K}] \} \end{aligned}$$

Of course, we intend to interpret **Lan**-types as actual left Kan extensions. To get started, we define their set and relational interpretations to be

$$\llbracket \Gamma; \Phi \vdash (\text{Lan}_{\overline{K}}^{\overline{\alpha}} F) \overline{A} \rrbracket^{\text{Set}} \rho = (\text{Lan}_{\lambda \overline{S}. \llbracket \Gamma; \overline{\alpha} \vdash K \rrbracket^{\text{Set}} \rho[\overline{\alpha} := \overline{S}]} \lambda \overline{S}. \llbracket \Gamma; \Phi, \overline{\alpha} \vdash F \rrbracket^{\text{Set}} \rho[\overline{\alpha} := \overline{S}]) \llbracket \Gamma; \Phi \vdash A \rrbracket^{\text{Set}} \rho$$

and

$$\llbracket \Gamma; \Phi \vdash (\text{Lan}_{\overline{K}}^{\overline{\alpha}} F) \overline{A} \rrbracket^{\text{Rel}} \rho = (\text{Lan}_{\lambda \overline{R}. \llbracket \Gamma; \overline{\alpha} \vdash K \rrbracket^{\text{Rel}} \rho[\overline{\alpha} := \overline{R}]} \lambda \overline{R}. \llbracket \Gamma; \Phi, \overline{\alpha} \vdash F \rrbracket^{\text{Rel}} \rho[\overline{\alpha} := \overline{R}]) \llbracket \Gamma; \Phi \vdash A \rrbracket^{\text{Rel}} \rho$$

respectively. But in order to guarantee that the IEL, and thus parametricity, continues to hold for the model we are constructing for our extended calculus, we might think we have to cut down the set interpretations of **Lan**-types by restricting to the subcollection of those interpretations that are well-behaved with respect to the IEL. This would mirror the cutting down we have already seen in the set interpretations of both our original **Nat**-types and our of generalized **Nat**-types. However, as we will see in Proposition 8.2 below, this will not be possible if we want the set and relational interpretations of our **Lan**-types to include the set and relational interpretations of the terms given by the **Lan**-introduction rule. Since we do indeed want this, the upshot is that the above interpretations of **Lan**-types are, in fact, the only possibility.

We must also define the functorial actions of the above set and relational interpretations of **Lan**-types on morphisms. If $f : \rho \rightarrow \rho'$ is a morphism of set environments, then

$$\llbracket \Gamma; \Phi \vdash (\text{Lan}_{\overline{K}}^{\overline{\alpha}} F) \overline{A} \rrbracket^{\text{Set}} f : \llbracket \Gamma; \Phi \vdash (\text{Lan}_{\overline{K}}^{\overline{\alpha}} F) \overline{A} \rrbracket^{\text{Set}} \rho \rightarrow \llbracket \Gamma; \Phi \vdash (\text{Lan}_{\overline{K}}^{\overline{\alpha}} F) \overline{A} \rrbracket^{\text{Set}} \rho'$$

is defined to be

$$\begin{aligned} & (\text{Lan}_{\lambda \overline{S}. \llbracket \Gamma; \overline{\alpha} \vdash K \rrbracket^{\text{Set}} \rho[\overline{\alpha} := \overline{S}]} \lambda \overline{S}. \llbracket \Gamma; \Phi, \overline{\alpha} \vdash F \rrbracket^{\text{Set}} f[\overline{\alpha} := id_S]) \llbracket \Gamma; \Phi \vdash A \rrbracket^{\text{Set}} \rho' \\ & \circ (\text{Lan}_{\lambda \overline{S}. \llbracket \Gamma; \overline{\alpha} \vdash K \rrbracket^{\text{Set}} \rho[\overline{\alpha} := \overline{S}]} \lambda \overline{S}. \llbracket \Gamma; \Phi, \overline{\alpha} \vdash F \rrbracket^{\text{Set}} \rho[\overline{\alpha} := \overline{S}]) \llbracket \Gamma; \Phi \vdash A \rrbracket^{\text{Set}} f \end{aligned}$$

or, equivalently by naturality,

$$\begin{aligned} & (\text{Lan}_{\lambda \overline{S}. \llbracket \Gamma; \overline{\alpha} \vdash K \rrbracket^{\text{Set}} \rho[\overline{\alpha} := \overline{S}]} \lambda \overline{S}. \llbracket \Gamma; \Phi, \overline{\alpha} \vdash F \rrbracket^{\text{Set}} \rho'[\overline{\alpha} := \overline{S}]) \llbracket \Gamma; \Phi \vdash A \rrbracket^{\text{Set}} f \\ & \circ (\text{Lan}_{\lambda \overline{S}. \llbracket \Gamma; \overline{\alpha} \vdash K \rrbracket^{\text{Set}} \rho[\overline{\alpha} := \overline{S}]} \lambda \overline{S}. \llbracket \Gamma; \Phi, \overline{\alpha} \vdash F \rrbracket^{\text{Set}} f[\overline{\alpha} := id_S]) \llbracket \Gamma; \Phi \vdash A \rrbracket^{\text{Set}} \rho \end{aligned}$$

The functorial action for the relational interpretations of **Lan**-types is defined analogously.

8.4. Extending the Term Semantics. Recalling the notation for the colimit representation of left Kan extensions from Section 8.1, we define the set and relational interpretations of the terms from the introduction and elimination rules for **Lan** types from Section 8.2 by

$$\llbracket \Gamma; \emptyset \mid \emptyset \vdash \int_{\overline{K}, F} : \text{Nat}^{\Phi, \overline{\alpha}} F (\text{Lan}_{\overline{K}}^{\overline{\alpha}} F) \overline{K} \rrbracket^{\text{Set}} \rho d = \eta$$

and

$$\llbracket \Gamma; \emptyset \mid \Delta \vdash \partial_F^{G, \overline{K}} t : \text{Nat}^{\Phi, \overline{\beta}} (\text{Lan}_{\overline{K}}^{\overline{\alpha}} F) \overline{\beta} G \rrbracket^{\text{Set}} \rho = \mu$$

respectively, where, for each $\overline{N} : RT_k$, $\eta_{\overline{N}}$ is the natural transformation associated with the left Kan extension

$$\begin{array}{ccc} \text{Set}^{\llbracket \overline{\alpha} \rrbracket} & \xrightarrow{\lambda \overline{A}. \llbracket \Gamma; \Phi, \overline{\alpha} \vdash F \rrbracket^{\text{Set}} \rho[\Phi, \overline{\alpha} := \overline{N}, \overline{A}]} & \text{Set} \\ & \searrow \eta_{\overline{N}} \quad \swarrow \text{Lan}_{\lambda \overline{A}. \llbracket \Gamma; \overline{\alpha} \vdash K \rrbracket^{\text{Set}} \rho[\overline{\alpha} := \overline{A}]} \lambda \overline{A}. \llbracket \Gamma; \Phi, \overline{\alpha} \vdash F \rrbracket^{\text{Set}} \rho[\Phi, \overline{\alpha} := \overline{N}, \overline{A}] & \\ \overline{\lambda \overline{A}. \llbracket \Gamma; \overline{\alpha} \vdash K \rrbracket^{\text{Set}} \rho[\overline{\alpha} := \overline{A}]} & & \text{Set}^{\llbracket \overline{K} \rrbracket} \end{array}$$

and, for all $\overline{N} : RT_k$ and $d : \llbracket \Gamma; \emptyset \vdash \Delta \rrbracket^{\text{Set}}$,

$$\begin{aligned} \mu_{\overline{N}d} : \text{Lan}_{\lambda \overline{A}. \llbracket \Gamma; \overline{\alpha} \vdash K \rrbracket^{\text{Set}} \rho[\overline{\alpha} := \overline{A}]} \lambda \overline{A}. \llbracket \Gamma; \Phi, \overline{\alpha} \vdash F \rrbracket^{\text{Set}} \rho[\Phi, \overline{\alpha} := \overline{N}, \overline{A}] \\ \rightarrow \lambda \overline{B}. \llbracket \Gamma; \Phi, \overline{\beta} \vdash G \rrbracket^{\text{Set}} \rho[\Phi, \overline{\beta} := \overline{N}, \overline{B}] \end{aligned}$$

is the unique natural transformation such that

$$(\mu_{\overline{N}d} \lambda \overline{A}. \llbracket \Gamma; \overline{\alpha} \vdash K \rrbracket^{\text{Set}} \rho[\overline{\alpha} := \overline{A}]) \circ \eta_{\overline{N}} = (\llbracket \Gamma; \emptyset \mid \Delta \vdash t : \text{Nat}^{\Phi, \overline{\alpha}} F \ G[\overline{\beta} := \overline{K}] \rrbracket^{\text{Set}} \rho d)_{\overline{N}} \quad (8.3)$$

as given by the universal property of the left Kan extension. The relational interpretations $\llbracket \Gamma; \emptyset \mid \emptyset \vdash \int_{\overline{K}, F} : \text{Nat}^{\Phi, \overline{\alpha}} F (\text{Lan}_{\overline{K}}^{\overline{\alpha}} F) \overline{K} \rrbracket^{\text{Rel}}$ and $\llbracket \Gamma; \emptyset \mid \Delta \vdash \partial_F^{G, \overline{K}} t : \text{Nat}^{\Phi, \overline{\beta}} (\text{Lan}_{\overline{K}}^{\overline{\alpha}} F) \overline{\beta} \ G \rrbracket^{\text{Set}}$ are defined entirely analogously.

We conclude this subsection by showing, as promised above, that no cutting down of set and relational interpretations of **Lan**-types is possible. We have:

Proposition 8.2. *Let $F : \text{Set}^k \rightarrow \text{Set}$ and $\overline{K} : \text{Set}^k \rightarrow \text{Set}^h$ be functors. If $L : \text{Set}^h \rightarrow \text{Set}$ is a functor such that*

- $L\overline{A} \subseteq (\text{Lan}_{\overline{K}} F)\overline{A}$ for all $\overline{A} : \text{Set}$,
- $(\text{Lan}_{\overline{K}} F)\overline{f}|_{L\overline{A}} x \in L\overline{B}$ for all $\overline{f} : \overline{A} \rightarrow \overline{B}$ and $x \in L\overline{A}$, and
- $\eta_{\overline{A}} y \in L(K\overline{A})$ for all $\overline{A} : \text{Set}$ and $y \in F\overline{A}$,

then $L = \text{Lan}_{\overline{K}} F$.

Proof. For all $\overline{A} : \text{Set}$ and $z \in (\text{Lan}_{\overline{K}} F)\overline{A}$ there exist $\overline{S} : \text{Set}_0$, $f : K\overline{S} \rightarrow \overline{A}$, and $w \in F\overline{S}$ such that $z = \iota_{\overline{S}, \overline{f}} w$. Thus, $z = \iota_{\overline{S}, \overline{f}} w = (\text{Lan}_{\overline{K}} F)\overline{f}(\eta_{\overline{S}} w)$ by Equation 8.2. Then by the third assumption above we have that $\eta_{\overline{S}} w \in L(K\overline{S})$, and by the second assumption above we have that $z \in L\overline{A}$. This gives $(\text{Lan}_{\overline{K}} F)\overline{A} \subseteq L\overline{A}$. Finally, by the first assumption above, we therefore have that $L\overline{A} = (\text{Lan}_{\overline{K}} F)\overline{A}$. \square

Thus, if L were a restriction of $\lambda \overline{B}. \llbracket \Gamma; \Phi, \overline{\beta} \vdash (\text{Lan}_{\overline{K}}^{\overline{\alpha}} F) \overline{\beta} \rrbracket^{\text{Set}} \rho[\Phi := \overline{N}] [\overline{\beta} := \overline{B}]$, if the functorial action of L were a restriction of that of $\lambda \overline{B}. \llbracket \Gamma; \Phi, \overline{\beta} \vdash (\text{Lan}_{\overline{K}}^{\overline{\alpha}} F) \overline{\beta} \rrbracket^{\text{Set}} \rho[\Phi := \overline{N}] [\overline{\beta} := \overline{B}]$, and if $L(\llbracket \Gamma; \overline{\alpha} \vdash K \rrbracket^{\text{Set}} \rho[\overline{\alpha} := \overline{A}])$ contained $\llbracket \Gamma; \emptyset \mid \emptyset \vdash \int_{\overline{K}, F} : \text{Nat}^{\Phi, \overline{\alpha}} F (\text{Lan}_{\overline{K}}^{\overline{\alpha}} F) \overline{K} \rrbracket^{\text{Set}} \rho d \overline{N} \ \overline{A} \ y$ for all \overline{N} , \overline{A} , and y , then L would have to be the entirety of $\lambda \overline{B}. \llbracket \Gamma; \Phi, \overline{\beta} \vdash (\text{Lan}_{\overline{K}}^{\overline{\alpha}} F) \overline{\beta} \rrbracket^{\text{Set}} \rho[\Phi := \overline{N}] [\overline{\beta} := \overline{B}]$. An analogous result holds for relational interpretations of **Lan**-types.

8.5. Parametricity and GADTs. Having extended our calculus with both **Lan**-types and terms of such types, and having given the only sensible set and relational interpretations for these types and terms, we now need to verify that these interpretations give rise to a parametric model. The first step in this process is to extend Lemma 3.13 to **Lan**-types by adding a clause for **Lan**-types to the proof. Unfortunately, however, Lemma 3.13 does not extend to arbitrary **Lan**-types, as the following example shows.

Example 8.3. Consider the type $\emptyset; \alpha \vdash (\text{Lan}_1^\emptyset \mathbb{1})\alpha$. The analogue for Rel of Equation 8.1 gives

$$\begin{aligned} \llbracket \emptyset; \alpha \vdash (\text{Lan}_1^\emptyset \mathbb{1})\alpha \rrbracket^{\text{Rel}} \rho[\alpha := (1, 0, 0)] &= (\text{Lan}_{\text{Eq}_1} \text{Eq}_1)(1, 0, 0) \\ &= \lim_{\xrightarrow{f: \text{Eq}_1 \rightarrow (1, 0, 0)}} \text{Eq}_1 \\ &= (0, 0, 0) \end{aligned}$$

Here, the last equality holds because there are no morphisms in Rel from Eq_1 to $(1, 0, 0)$, and because $(0, 0, 0)$ is the initial object in that category. On the other hand, for the set interpretation with

respect to the first projection of the relation environment $\rho[\alpha := (1, 0, 0)]$, Equation 8.1 gives

$$\begin{aligned} \llbracket \emptyset; \alpha \vdash (\text{Lan}_{\mathbf{1}}^{\emptyset} \mathbf{1}) \alpha \rrbracket^{\text{Set}}(\pi_1(\rho[\alpha := (1, 0, 0)])) &= \llbracket \emptyset; \alpha \vdash (\text{Lan}_{\mathbf{1}}^{\emptyset} \mathbf{1}) \alpha \rrbracket^{\text{Set}}(\pi_1 \rho)[\alpha := 1] \\ &= (\text{Lan}_{\mathbf{1}} \mathbf{1}) \mathbf{1} \\ &= \lim_{\xrightarrow{1 \rightarrow 1}} \mathbf{1} \\ &= 1 \end{aligned}$$

because $\text{id}_{\mathbf{1}}$ is the unique arrow from 1 to 1 in Set . Lemma 3.13 therefore cannot hold.

The problem in Example 8.3 lies in the fact that the type along whose interpretation we extend contains constants, i.e., subtypes constructed from $\mathbf{1}$. Indeed, if $\overline{\emptyset; \bar{\alpha} \vdash K}$ consists only of polynomial (i.e., sum-of-products) types not containing constants (i.e., formed only from $+$, \times , and the variables in α), then Lemma 3.13 actually does hold for $\Gamma; \Phi \vdash (\text{Lan}_{\bar{K}}^{\bar{\alpha}} F) \bar{A}$. This is proved in the following proposition, which covers the case of Lemma 3.13 for such Lan -types.

Proposition 8.4. *If $\overline{\emptyset; \bar{\alpha} \vdash K}$ consists only of polynomial types not containing constants, then*

$$(\llbracket \Gamma; \Phi \vdash (\text{Lan}_{\bar{K}}^{\bar{\alpha}} F) \bar{A} \rrbracket^{\text{Set}}, \llbracket \Gamma; \Phi \vdash (\text{Lan}_{\bar{K}}^{\bar{\alpha}} F) \bar{A} \rrbracket^{\text{Set}}, \llbracket \Gamma; \Phi \vdash (\text{Lan}_{\bar{K}}^{\bar{\alpha}} F) \bar{A} \rrbracket^{\text{Rel}}) \in [\text{RelEnv}, R\text{To}]$$

Proof. We need only show that

$$\pi_i(\llbracket \Gamma; \Phi \vdash (\text{Lan}_{\bar{K}}^{\bar{\alpha}} F) \bar{A} \rrbracket^{\text{Rel}} \rho) = \llbracket \Gamma; \Phi \vdash (\text{Lan}_{\bar{K}}^{\bar{\alpha}} F) \bar{A} \rrbracket^{\text{Set}}(\pi_i \rho) \quad (8.4)$$

and

$$\pi_i(\llbracket \Gamma; \Phi \vdash (\text{Lan}_{\bar{K}}^{\bar{\alpha}} F) \bar{A} \rrbracket^{\text{Rel}} f) = \llbracket \Gamma; \Phi \vdash (\text{Lan}_{\bar{K}}^{\bar{\alpha}} F) \bar{A} \rrbracket^{\text{Set}}(\pi_i f) \quad (8.5)$$

for all $\rho, \rho', f : \rho \rightarrow \rho'$, and $i \in \{1, 2\}$.

To prove Equation 8.4 we first observe that, by (the analogue for Rel of) Equation 8.1, we have

$$\llbracket \Gamma; \Phi \vdash (\text{Lan}_{\bar{K}}^{\bar{\alpha}} F) \bar{A} \rrbracket^{\text{Rel}} \rho = \lim_{\xrightarrow{R: \text{Rel}_0, m: \llbracket \emptyset; \bar{\alpha} \vdash K \rrbracket^{\text{Rel}} \rho[\bar{\alpha} := \bar{R}] \rightarrow \llbracket \Gamma; \Phi \vdash A \rrbracket^{\text{Rel}} \rho}} \llbracket \Gamma; \Phi, \bar{\alpha} \vdash F \rrbracket^{\text{Rel}} \rho[\bar{\alpha} := \bar{R}]$$

with ι mapping the cocone into this colimit. Since each projection π_i is cocontinuous, we therefore have that

$$\pi_i(\llbracket \Gamma; \Phi \vdash (\text{Lan}_{\bar{K}}^{\bar{\alpha}} F) \bar{A} \rrbracket^{\text{Rel}} \rho) = \lim_{\xrightarrow{R: \text{Rel}_0, m: \llbracket \emptyset; \bar{\alpha} \vdash K \rrbracket^{\text{Rel}} \rho[\bar{\alpha} := \bar{R}] \rightarrow \llbracket \Gamma; \Phi \vdash A \rrbracket^{\text{Rel}} \rho}} \llbracket \Gamma; \Phi, \bar{\alpha} \vdash F \rrbracket^{\text{Set}}(\pi_i \rho)[\bar{\alpha} := \pi_i \bar{R}]$$

with $\pi_i \iota$ mapping the cocone into this colimit. On the other hand, Equation 8.1 also gives

$$\llbracket \Gamma; \Phi \vdash (\text{Lan}_{\bar{K}}^{\bar{\alpha}} F) \bar{A} \rrbracket^{\text{Set}}(\pi_i \rho) = \lim_{\xrightarrow{S: \text{Set}_0, f: \llbracket \emptyset; \bar{\alpha} \vdash K \rrbracket^{\text{Set}}(\pi_i \rho)[\bar{\alpha} := \bar{S}] \rightarrow \llbracket \Gamma; \Phi \vdash A \rrbracket^{\text{Set}}(\pi_i \rho)}} \llbracket \Gamma; \Phi, \bar{\alpha} \vdash F \rrbracket^{\text{Set}}(\pi_i \rho)[\bar{\alpha} := \bar{S}]$$

with j mapping the cocone into this colimit.

When $i = 1$ we prove that

$$\pi_1(\llbracket \Gamma; \Phi \vdash (\text{Lan}_{\bar{K}}^{\bar{\alpha}} F) \bar{A} \rrbracket^{\text{Rel}} \rho) = \llbracket \Gamma; \Phi \vdash (\text{Lan}_{\bar{K}}^{\bar{\alpha}} F) \bar{A} \rrbracket^{\text{Set}}(\pi_1 \rho)$$

by providing a pair of inverse functions; the proof when $i = 2$ is entirely analogous. To this end, we define

$$h : \pi_1(\llbracket \Gamma; \Phi \vdash (\text{Lan}_{\bar{K}}^{\bar{\alpha}} F) \bar{A} \rrbracket^{\text{Rel}} \rho) \rightarrow \llbracket \Gamma; \Phi \vdash (\text{Lan}_{\bar{K}}^{\bar{\alpha}} F) \bar{A} \rrbracket^{\text{Set}}(\pi_1 \rho)$$

to be the unique function given by the universal property of the colimit representation of h 's domain. That is, we take h to be the unique morphism such that, for any $\bar{R} : \text{Rel}_0$ and

$$m : \llbracket \emptyset; \bar{\alpha} \vdash K \rrbracket^{\text{Rel}} \rho[\bar{\alpha} := \bar{R}] \rightarrow \llbracket \Gamma; \Phi \vdash A \rrbracket^{\text{Rel}} \rho$$

we have

$$h \circ (\pi_1 \iota_{\bar{R}, \bar{m}}) = j_{\pi_1 \bar{R}, \pi_1 \bar{m}} \quad (8.6)$$

Now, since K does not contain constants, $\llbracket \emptyset; \bar{\alpha} \vdash K \rrbracket^{\text{Set}}(\pi_2 \rho)[\bar{\alpha} := \bar{0}] = 0$. Thus, for any $\bar{S} : \text{Set}_0$ and $f : \llbracket \emptyset; \bar{\alpha} \vdash K \rrbracket^{\text{Set}}(\pi_1 \rho)[\bar{\alpha} := \bar{S}] \rightarrow \llbracket \Gamma; \Phi \vdash A \rrbracket^{\text{Set}}(\pi_1 \rho)$ we have relations $(\bar{S}, 0, 0)$ and morphisms $(f, !): \llbracket \emptyset; \bar{\alpha} \vdash K \rrbracket^{\text{Rel}} \rho[\bar{\alpha} := (\bar{S}, 0, 0)] \rightarrow \llbracket \Gamma; \Phi \vdash A \rrbracket^{\text{Rel}} \rho$, where, for any set X , we suppress the sub- and superscripts and write $!$ for the unique morphism $!_X^0$ from 0 to X . We then define

$$k : \llbracket \Gamma; \Phi \vdash (\text{Lan}_{\bar{K}}^{\bar{\alpha}} F) \bar{A} \rrbracket^{\text{Set}}(\pi_1 \rho) \rightarrow \pi_1(\llbracket \Gamma; \Phi \vdash (\text{Lan}_{\bar{K}}^{\bar{\alpha}} F) \bar{A} \rrbracket^{\text{Rel}} \rho)$$

to be the unique function given by the universal property of the colimit representation of k 's domain.

That is, we take k to be the unique morphism such that, for any $\overline{S} : \mathbf{Set}_0$ and $f : \llbracket \emptyset; \bar{\alpha} \vdash K \rrbracket^{\mathbf{Set}}(\pi_1 \rho)[\bar{\alpha} := \overline{S}] \rightarrow \llbracket \Gamma; \Phi \vdash A \rrbracket^{\mathbf{Set}}(\pi_1 \rho)$ we have $k \circ j_{\overline{S}, \bar{f}} = \pi_1 \iota_{(\overline{S}, 0, 0), (\bar{f}, !)}$.

To see that h and k are mutually inverse we first observe that

$$h \circ k \circ j_{\overline{S}, \bar{f}} = h \circ (\pi_1 \iota_{(\overline{S}, 0, 0), (\bar{f}, !)}) = j_{\pi_1(\overline{S}, 0, 0), \pi_1(\bar{f}, !)} = j_{\overline{S}, \bar{f}}$$

for all $\overline{S} : \mathbf{Set}_0$ and $f : \llbracket \emptyset; \bar{\alpha} \vdash K \rrbracket^{\mathbf{Set}}(\pi_1 \rho)[\bar{\alpha} := \overline{S}] \rightarrow \llbracket \Gamma; \Phi \vdash A \rrbracket^{\mathbf{Set}}(\pi_1 \rho)$, so that $h \circ k = id$. To see that $k \circ h = id$ we first observe that

$$k \circ h \circ (\pi_1 \iota_{\overline{R}, \overline{m}}) = k \circ j_{\pi_1 \overline{R}, \pi_1 \overline{m}} = \pi_1 \iota_{(\pi_1 \overline{R}, 0, 0), (\pi_1 \overline{m}, !)} \quad (8.7)$$

for all $\overline{R} : \mathbf{Rel}_0$ and $m : \llbracket \emptyset; \bar{\alpha} \vdash K \rrbracket^{\mathbf{Rel}} \rho[\bar{\alpha} := \overline{R}] \rightarrow \llbracket \Gamma; \Phi \vdash A \rrbracket^{\mathbf{Rel}} \rho$. Then note that, for each sequence of morphisms $(id_{\pi_1 \overline{R}}, !) : (\pi_1 \overline{R}, 0, 0) \rightarrow \overline{R}$, we have that

$$m \circ \llbracket \emptyset; \bar{\alpha} \vdash K \rrbracket^{\mathbf{Rel}} id_{\rho}[\bar{\alpha} := (id_{\pi_1 \overline{R}}, !)] = (\pi_1 m, !)$$

in the indexing category for the colimit representation of the codomain of k . This implies that

$$\iota_{\overline{R}, \overline{m}} \circ \llbracket \Gamma; \Phi, \bar{\alpha} \vdash F \rrbracket^{\mathbf{Rel}} id_{\rho}[\bar{\alpha} := (id_{\pi_1 \overline{R}}, !)] = \iota_{(\pi_1 \overline{R}, 0, 0), (\pi_1 \overline{m}, !)}$$

Projecting the first component thus gives

$$(\pi_1 \iota_{\overline{R}, \overline{m}}) \circ \pi_1(\llbracket \Gamma; \Phi, \bar{\alpha} \vdash F \rrbracket^{\mathbf{Rel}} id_{\rho}[\bar{\alpha} := (id_{\pi_1 \overline{R}}, !)]) = \pi_1 \iota_{(\pi_1 \overline{R}, 0, 0), (\pi_1 \overline{m}, !)}$$

Now, the induction hypothesis on $\Gamma; \Phi, \bar{\alpha} \vdash F$ gives that

$$\begin{aligned} & \pi_1(\llbracket \Gamma; \Phi, \bar{\alpha} \vdash F \rrbracket^{\mathbf{Rel}} id_{\rho}[\bar{\alpha} := (id_{\pi_1 \overline{R}}, !)]) \\ &= \llbracket \Gamma; \Phi, \bar{\alpha} \vdash F \rrbracket^{\mathbf{Set}} id_{\pi_1 \rho}[\bar{\alpha} := id_{\pi_1 \overline{R}}] \\ &= id_{\llbracket \Gamma; \Phi, \bar{\alpha} \vdash F \rrbracket^{\mathbf{Set}}(\pi_1 \rho)[\bar{\alpha} := \pi_1 \overline{R}]} \end{aligned}$$

so that, in fact, $\pi_1 \iota_{\overline{R}, \overline{m}} = \pi_1 \iota_{(\pi_1 \overline{R}, 0, 0), (\pi_1 \overline{m}, !)}$. Finally, by Equation 8.7 we have $k \circ h \circ (\pi_1 \iota_{\overline{R}, \overline{m}}) = \pi_1 \iota_{(\pi_1 \overline{R}, 0, 0), (\pi_1 \overline{m}, !)} = \pi_1 \iota_{\overline{R}, \overline{m}}$ for all \overline{R} and \overline{m} , and thus $k \circ h = id$.

To prove Equation 8.5, first recall that $\llbracket \Gamma; \Phi \vdash (\mathbf{Lan}_{\overline{K}}^{\overline{\alpha}} F) \overline{A} \rrbracket^{\mathbf{Rel}} f$ is defined to be

$$\begin{aligned} & (\mathbf{Lan}_{\lambda \overline{R}. \llbracket \Gamma; \bar{\alpha} \vdash K \rrbracket^{\mathbf{Rel}} \rho[\bar{\alpha} := \overline{R}]} \lambda \overline{R}. \llbracket \Gamma; \Phi, \bar{\alpha} \vdash F \rrbracket^{\mathbf{Rel}} f[\bar{\alpha} := id_{\overline{R}}]) \llbracket \Gamma; \Phi \vdash A \rrbracket^{\mathbf{Rel}} \rho' \\ & \circ (\mathbf{Lan}_{\lambda \overline{R}. \llbracket \Gamma; \bar{\alpha} \vdash K \rrbracket^{\mathbf{Rel}} \rho[\bar{\alpha} := \overline{R}]} \lambda \overline{R}. \llbracket \Gamma; \Phi, \bar{\alpha} \vdash F \rrbracket^{\mathbf{Rel}} \rho[\bar{\alpha} := \overline{R}]) \llbracket \Gamma; \Phi \vdash A \rrbracket^{\mathbf{Rel}} f \end{aligned}$$

and $\llbracket \Gamma; \Phi \vdash (\mathbf{Lan}_{\overline{K}}^{\overline{\alpha}} F) \overline{A} \rrbracket^{\mathbf{Set}}(\pi_i f)$ is defined to be

$$\begin{aligned} & (\mathbf{Lan}_{\lambda \overline{S}. \llbracket \Gamma; \bar{\alpha} \vdash K \rrbracket^{\mathbf{Set}}(\pi_i \rho)[\bar{\alpha} := \overline{S}]} \lambda \overline{S}. \llbracket \Gamma; \Phi, \bar{\alpha} \vdash F \rrbracket^{\mathbf{Set}}(\pi_i f)[\bar{\alpha} := id_{\overline{S}}]) \llbracket \Gamma; \Phi \vdash A \rrbracket^{\mathbf{Set}}(\pi_i \rho') \\ & \circ (\mathbf{Lan}_{\lambda \overline{S}. \llbracket \Gamma; \bar{\alpha} \vdash K \rrbracket^{\mathbf{Set}}(\pi_i \rho)[\bar{\alpha} := \overline{S}]} \lambda \overline{S}. \llbracket \Gamma; \Phi, \bar{\alpha} \vdash F \rrbracket^{\mathbf{Set}}(\pi_i \rho)[\bar{\alpha} := \overline{S}]) \llbracket \Gamma; \Phi \vdash A \rrbracket^{\mathbf{Rel}}(\pi_i f) \end{aligned}$$

It therefore suffices to show that

$$\begin{aligned} & \pi_i \left((\mathbf{Lan}_{\lambda \overline{R}. \llbracket \Gamma; \bar{\alpha} \vdash K \rrbracket^{\mathbf{Rel}} \rho[\bar{\alpha} := \overline{R}]} \lambda \overline{R}. \llbracket \Gamma; \Phi, \bar{\alpha} \vdash F \rrbracket^{\mathbf{Rel}} f[\bar{\alpha} := id_{\overline{R}}]) \llbracket \Gamma; \Phi \vdash A \rrbracket^{\mathbf{Rel}} \rho' \right) \\ &= (\mathbf{Lan}_{\lambda \overline{S}. \llbracket \Gamma; \bar{\alpha} \vdash K \rrbracket^{\mathbf{Set}}(\pi_i \rho)[\bar{\alpha} := \overline{S}]} \lambda \overline{S}. \llbracket \Gamma; \Phi, \bar{\alpha} \vdash F \rrbracket^{\mathbf{Set}}(\pi_i f)[\bar{\alpha} := id_{\overline{S}}]) \llbracket \Gamma; \Phi \vdash A \rrbracket^{\mathbf{Set}}(\pi_i \rho') \quad (8.8) \end{aligned}$$

and that

$$\begin{aligned} & \pi_i \left((\mathbf{Lan}_{\lambda \overline{R}. \llbracket \Gamma; \bar{\alpha} \vdash K \rrbracket^{\mathbf{Rel}} \rho[\bar{\alpha} := \overline{R}]} \lambda \overline{R}. \llbracket \Gamma; \Phi, \bar{\alpha} \vdash F \rrbracket^{\mathbf{Rel}} \rho[\bar{\alpha} := \overline{R}]) \llbracket \Gamma; \Phi \vdash A \rrbracket^{\mathbf{Rel}} f \right) \\ &= (\mathbf{Lan}_{\lambda \overline{S}. \llbracket \Gamma; \bar{\alpha} \vdash K \rrbracket^{\mathbf{Set}}(\pi_i \rho)[\bar{\alpha} := \overline{S}]} \lambda \overline{S}. \llbracket \Gamma; \Phi, \bar{\alpha} \vdash F \rrbracket^{\mathbf{Set}}(\pi_i \rho)[\bar{\alpha} := \overline{S}]) \llbracket \Gamma; \Phi \vdash A \rrbracket^{\mathbf{Set}}(\pi_i f) \quad (8.9) \end{aligned}$$

To prove Equation 8.8 we first observe that the morphisms $\iota'_{\overline{R}, \overline{m}} \circ \llbracket \Gamma; \Phi, \overline{\alpha} \vdash F \rrbracket^{\text{Rel}} f[\overline{\alpha} := id_R]$ form a cocone for

$$(Lan_{\overline{\lambda R}. \llbracket \Gamma; \overline{\alpha} \vdash K \rrbracket^{\text{Rel}} \rho[\overline{\alpha} := \overline{R}]} \overline{\lambda R}. \llbracket \Gamma; \Phi, \overline{\alpha} \vdash F \rrbracket^{\text{Rel}} \rho[\overline{\alpha} := \overline{R}]) \overline{\llbracket \Gamma; \Phi \vdash A \rrbracket^{\text{Rel}} \rho'}$$

with vertex

$$(Lan_{\overline{\lambda R}. \llbracket \Gamma; \overline{\alpha} \vdash K \rrbracket^{\text{Rel}} \rho[\overline{\alpha} := \overline{R}]} \overline{\lambda R}. \llbracket \Gamma; \Phi, \overline{\alpha} \vdash F \rrbracket^{\text{Rel}} \rho'[\overline{\alpha} := \overline{R}]) \overline{\llbracket \Gamma; \Phi \vdash A \rrbracket^{\text{Rel}} \rho'}$$

The universal property (Equation 8.2) of

$$(Lan_{\overline{\lambda R}. \llbracket \Gamma; \overline{\alpha} \vdash K \rrbracket^{\text{Rel}} \rho[\overline{\alpha} := \overline{R}]} \overline{\lambda R}. \llbracket \Gamma; \Phi, \overline{\alpha} \vdash F \rrbracket^{\text{Rel}} f[\overline{\alpha} := id_R]) \overline{\llbracket \Gamma; \Phi \vdash A \rrbracket^{\text{Rel}} \rho'}$$

therefore gives that, for all $\overline{R} : \overline{\text{Rel}}_0$ and $\overline{m} : \llbracket \Gamma; \overline{\alpha} \vdash K \rrbracket^{\text{Rel}} \rho[\overline{\alpha} := \overline{R}] \rightarrow \llbracket \Gamma; \Phi \vdash A \rrbracket^{\text{Rel}} \rho'$,

$$\begin{aligned} & (Lan_{\overline{\lambda R}. \llbracket \Gamma; \overline{\alpha} \vdash K \rrbracket^{\text{Rel}} \rho[\overline{\alpha} := \overline{R}]} \overline{\lambda R}. \llbracket \Gamma; \Phi, \overline{\alpha} \vdash F \rrbracket^{\text{Rel}} f[\overline{\alpha} := id_R]) \overline{\llbracket \Gamma; \Phi \vdash A \rrbracket^{\text{Rel}} \rho'} \circ \iota_{\overline{R}, \overline{m}} \\ &= \iota'_{\overline{R}, \overline{m}} \circ \llbracket \Gamma; \Phi, \overline{\alpha} \vdash F \rrbracket^{\text{Rel}} f[\overline{\alpha} := id_R] \end{aligned}$$

Here, ι is the morphism mapping the cocone into the colimit

$$(Lan_{\overline{\lambda R}. \llbracket \Gamma; \overline{\alpha} \vdash K \rrbracket^{\text{Rel}} \rho[\overline{\alpha} := \overline{R}]} \overline{\lambda R}. \llbracket \Gamma; \Phi, \overline{\alpha} \vdash F \rrbracket^{\text{Rel}} \rho[\overline{\alpha} := \overline{R}]) \overline{\llbracket \Gamma; \Phi \vdash A \rrbracket^{\text{Rel}} \rho'}$$

and ι' is the morphism mapping the cocone into the colimit

$$(Lan_{\overline{\lambda R}. \llbracket \Gamma; \overline{\alpha} \vdash K \rrbracket^{\text{Rel}} \rho[\overline{\alpha} := \overline{R}]} \overline{\lambda R}. \llbracket \Gamma; \Phi, \overline{\alpha} \vdash F \rrbracket^{\text{Rel}} \rho'[\overline{\alpha} := \overline{R}]) \overline{\llbracket \Gamma; \Phi \vdash A \rrbracket^{\text{Rel}} \rho'}$$

Projecting, together with the induction hypothesis for $\Gamma; \Phi, \overline{\alpha} \vdash F$, thus gives

$$\begin{aligned} & \pi_i \left((Lan_{\overline{\lambda R}. \llbracket \Gamma; \overline{\alpha} \vdash K \rrbracket^{\text{Rel}} \rho[\overline{\alpha} := \overline{R}]} \overline{\lambda R}. \llbracket \Gamma; \Phi, \overline{\alpha} \vdash F \rrbracket^{\text{Rel}} f[\overline{\alpha} := id_R]) \overline{\llbracket \Gamma; \Phi \vdash A \rrbracket^{\text{Rel}} \rho'} \right) \circ \pi_i \iota_{\overline{R}, \overline{m}} \\ &= \pi_i \iota'_{\overline{R}, \overline{m}} \circ \llbracket \Gamma; \Phi, \overline{\alpha} \vdash F \rrbracket^{\text{Set}} (\pi_i f)[\overline{\alpha} := id_{\pi_i R}] \end{aligned}$$

Equation 8.6 and its analogue for ι' and j' then give

$$\begin{aligned} & \pi_i \left((Lan_{\overline{\lambda R}. \llbracket \Gamma; \overline{\alpha} \vdash K \rrbracket^{\text{Rel}} \rho[\overline{\alpha} := \overline{R}]} \overline{\lambda R}. \llbracket \Gamma; \Phi, \overline{\alpha} \vdash F \rrbracket^{\text{Rel}} f[\overline{\alpha} := id_R]) \overline{\llbracket \Gamma; \Phi \vdash A \rrbracket^{\text{Rel}} \rho'} \right) \circ k \circ j_{\pi_i \overline{R}, \pi_i \overline{m}} \\ &= \pi_i \left((Lan_{\overline{\lambda R}. \llbracket \Gamma; \overline{\alpha} \vdash K \rrbracket^{\text{Rel}} \rho[\overline{\alpha} := \overline{R}]} \overline{\lambda R}. \llbracket \Gamma; \Phi, \overline{\alpha} \vdash F \rrbracket^{\text{Rel}} f[\overline{\alpha} := id_R]) \overline{\llbracket \Gamma; \Phi \vdash A \rrbracket^{\text{Rel}} \rho'} \right) \circ k \circ h \circ \pi_i \iota_{\overline{R}, \overline{m}} \\ &= \pi_i \left((Lan_{\overline{\lambda R}. \llbracket \Gamma; \overline{\alpha} \vdash K \rrbracket^{\text{Rel}} \rho[\overline{\alpha} := \overline{R}]} \overline{\lambda R}. \llbracket \Gamma; \Phi, \overline{\alpha} \vdash F \rrbracket^{\text{Rel}} f[\overline{\alpha} := id_R]) \overline{\llbracket \Gamma; \Phi \vdash A \rrbracket^{\text{Rel}} \rho'} \right) \circ \pi_i \iota_{\overline{R}, \overline{m}} \\ &= \pi_i \iota'_{\overline{R}, \overline{m}} \circ \llbracket \Gamma; \Phi, \overline{\alpha} \vdash F \rrbracket^{\text{Set}} (\pi_i f)[\overline{\alpha} := id_{\pi_i R}] \\ &= k' \circ h' \circ \pi_i \iota'_{\overline{R}, \overline{m}} \circ \llbracket \Gamma; \Phi, \overline{\alpha} \vdash F \rrbracket^{\text{Set}} (\pi_i f)[\overline{\alpha} := id_{\pi_i R}] \\ &= k' \circ j'_{\pi_i \overline{R}, \pi_i \overline{m}} \circ \llbracket \Gamma; \Phi, \overline{\alpha} \vdash F \rrbracket^{\text{Set}} (\pi_i f)[\overline{\alpha} := id_{\pi_i R}] \end{aligned}$$

where j is the morphism mapping the cocone into the colimit

$$(Lan_{\overline{\lambda S}. \llbracket \Gamma; \overline{\alpha} \vdash K \rrbracket^{\text{Set}} (\pi_i \rho)[\overline{\alpha} := \overline{S}]} \overline{\lambda S}. \llbracket \Gamma; \Phi, \overline{\alpha} \vdash F \rrbracket^{\text{Set}} (\pi_i \rho)[\overline{\alpha} := \overline{S}]) \overline{\llbracket \Gamma; \Phi \vdash A \rrbracket^{\text{Set}} (\pi_i \rho')}$$

and j' is the morphism mapping the cocone into the colimit

$$(Lan_{\overline{\lambda S}. \llbracket \Gamma; \overline{\alpha} \vdash K \rrbracket^{\text{Set}} \rho[\overline{\alpha} := \overline{S}]} \overline{\lambda S}. \llbracket \Gamma; \Phi, \overline{\alpha} \vdash F \rrbracket^{\text{Set}} (\pi_i \rho')[\overline{\alpha} := \overline{S}]) \overline{\llbracket \Gamma; \Phi \vdash A \rrbracket^{\text{Set}} (\pi_i \rho')}$$

Since both k and k' are isomorphisms, we have that, up to isomorphism,

$$\begin{aligned} & \pi_i \left((Lan_{\overline{\lambda R}. \llbracket \Gamma; \overline{\alpha} \vdash K \rrbracket^{\text{Rel}} \rho[\overline{\alpha} := \overline{R}]} \overline{\lambda R}. \llbracket \Gamma; \Phi, \overline{\alpha} \vdash F \rrbracket^{\text{Rel}} f[\overline{\alpha} := id_R]) \overline{\llbracket \Gamma; \Phi \vdash A \rrbracket^{\text{Rel}} \rho'} \right) \circ j_{\pi_i \overline{R}, \pi_i \overline{m}} \\ &= j'_{\pi_i \overline{R}, \pi_i \overline{m}} \circ \llbracket \Gamma; \Phi, \overline{\alpha} \vdash F \rrbracket^{\text{Set}} (\pi_i f)[\overline{\alpha} := id_{\pi_i R}] \end{aligned}$$

By the surjectivity of each $\pi_i : \mathbf{Rel} \rightarrow \mathbf{Set}$ we have that

$$\begin{aligned} & \pi_i \left(\left(\text{Lan}_{\lambda \bar{R}. [\Gamma; \bar{\alpha} \vdash K]^\mathbf{Rel} \rho [\alpha := \bar{R}]} \lambda \bar{R}. [\Gamma; \Phi, \bar{\alpha} \vdash F]^\mathbf{Rel} f [\alpha := id_R] \right) \overline{[\Gamma; \Phi \vdash A]^\mathbf{Rel} \rho'} \right) \circ j_{\bar{S}, \bar{g}} \\ &= j'_{\bar{S}, \bar{g}} \circ \overline{[\Gamma; \Phi, \bar{\alpha} \vdash F]^\mathbf{Set} (\pi_i f) [\alpha := id_S]} \end{aligned}$$

for all $\bar{S} : \mathbf{Set}_0$ and $\bar{g} : \overline{[\Gamma; \bar{\alpha} \vdash K]^\mathbf{Set} (\pi_i \rho) [\alpha := \bar{S}] \rightarrow [\Gamma; \Phi \vdash A]^\mathbf{Set} (\pi_i \rho')}$. That is,

$$\pi_i \left(\left(\text{Lan}_{\lambda \bar{R}. [\Gamma; \bar{\alpha} \vdash K]^\mathbf{Rel} \rho [\alpha := \bar{R}]} \lambda \bar{R}. [\Gamma; \Phi, \bar{\alpha} \vdash F]^\mathbf{Rel} f [\alpha := id_R] \right) \overline{[\Gamma; \Phi \vdash A]^\mathbf{Rel} \rho'} \right)$$

satisfies the universal property (Equation 8.2) of

$$\left(\text{Lan}_{\lambda \bar{S}. [\Gamma; \bar{\alpha} \vdash K]^\mathbf{Set} (\pi_i \rho) [\alpha := \bar{S}]} \lambda \bar{S}. [\Gamma; \Phi, \bar{\alpha} \vdash F]^\mathbf{Set} (\pi_i f) [\alpha := id_S] \right) \overline{[\Gamma; \Phi \vdash A]^\mathbf{Set} (\pi_i \rho')}$$

The two expressions must therefore be equal, and thus Equation 8.8 holds.

To prove Equation 8.9 we first observe that the universal property (Equation 8.2) of

$$\left(\text{Lan}_{\lambda \bar{R}. [\Gamma; \bar{\alpha} \vdash K]^\mathbf{Rel} \rho [\alpha := \bar{R}]} \lambda \bar{R}. [\Gamma; \Phi, \bar{\alpha} \vdash F]^\mathbf{Rel} \rho [\alpha := \bar{R}] \right) \overline{[\Gamma; \Phi \vdash A]^\mathbf{Rel} f}$$

gives that, for every $\bar{R} : \mathbf{Rel}_0$ and $m : \overline{[\Gamma; \bar{\alpha} \vdash K]^\mathbf{Rel} \rho [\alpha := \bar{R}] \rightarrow [\Gamma; \Phi \vdash A]^\mathbf{Rel} \rho}$,

$$\begin{aligned} & \left(\text{Lan}_{\lambda \bar{R}. [\Gamma; \bar{\alpha} \vdash K]^\mathbf{Rel} \rho [\alpha := \bar{R}]} \lambda \bar{R}. [\Gamma; \Phi, \bar{\alpha} \vdash F]^\mathbf{Rel} \rho [\alpha := \bar{R}] \right) \overline{[\Gamma; \Phi \vdash A]^\mathbf{Rel} f} \circ \iota_{\bar{R}, \bar{m}} \\ &= \iota'_{\bar{R}, \overline{[\Gamma; \Phi \vdash A]^\mathbf{Rel} f \circ m}} \end{aligned}$$

where ι is the cocone into the colimit

$$\left(\text{Lan}_{\lambda \bar{R}. [\Gamma; \bar{\alpha} \vdash K]^\mathbf{Rel} \rho [\alpha := \bar{R}]} \lambda \bar{R}. [\Gamma; \Phi, \bar{\alpha} \vdash F]^\mathbf{Rel} \rho [\alpha := \bar{R}] \right) \overline{[\Gamma; \Phi \vdash A]^\mathbf{Rel} \rho}$$

and ι' is the cocone into the colimit

$$\left(\text{Lan}_{\lambda \bar{R}. [\Gamma; \bar{\alpha} \vdash K]^\mathbf{Rel} \rho [\alpha := \bar{R}]} \lambda \bar{R}. [\Gamma; \Phi, \bar{\alpha} \vdash F]^\mathbf{Rel} \rho [\alpha := \bar{R}] \right) \overline{[\Gamma; \Phi \vdash A]^\mathbf{Rel} \rho'}$$

Projecting, together with the induction hypothesis for $\Gamma; \Phi, \bar{\alpha} \vdash F$, thus gives

$$\begin{aligned} & \pi_i \left(\left(\text{Lan}_{\lambda \bar{R}. [\Gamma; \bar{\alpha} \vdash K]^\mathbf{Rel} \rho [\alpha := \bar{R}]} \lambda \bar{R}. [\Gamma; \Phi, \bar{\alpha} \vdash F]^\mathbf{Rel} \rho [\alpha := \bar{R}] \right) \overline{[\Gamma; \Phi \vdash A]^\mathbf{Rel} f} \right) \circ \pi_i \iota_{\bar{R}, \bar{m}} \\ &= \pi_i \iota'_{\bar{R}, \overline{[\Gamma; \Phi \vdash A]^\mathbf{Rel} f \circ m}} \end{aligned}$$

Equation 8.6 and its analogue for ι' and j' then give that, up to isomorphism,

$$\begin{aligned} & \pi_i \left(\left(\text{Lan}_{\lambda \bar{R}. [\Gamma; \bar{\alpha} \vdash K]^\mathbf{Rel} \rho [\alpha := \bar{R}]} \lambda \bar{R}. [\Gamma; \Phi, \bar{\alpha} \vdash F]^\mathbf{Rel} \rho [\alpha := \bar{R}] \right) \overline{[\Gamma; \Phi \vdash A]^\mathbf{Rel} f} \right) \circ j_{\pi_i \bar{R}, \pi_i \bar{m}} \\ &= j'_{\pi_i \bar{R}, \overline{[\Gamma; \Phi \vdash A]^\mathbf{Rel} (\pi_i f) \circ \pi_i m}} \end{aligned}$$

where j is the cocone into the colimit

$$\left(\text{Lan}_{\lambda \bar{S}. [\Gamma; \bar{\alpha} \vdash K]^\mathbf{Set} (\pi_i \rho) [\alpha := \bar{S}]} \lambda \bar{S}. [\Gamma; \Phi, \bar{\alpha} \vdash F]^\mathbf{Set} (\pi_i \rho) [\alpha := \bar{S}] \right) \overline{[\Gamma; \Phi \vdash A]^\mathbf{Set} (\pi_i \rho')}$$

and j' is the cocone into the colimit

$$\left(\text{Lan}_{\lambda \bar{S}. [\Gamma; \bar{\alpha} \vdash K]^\mathbf{Set} (\pi_i \rho) [\alpha := \bar{S}]} \lambda \bar{S}. [\Gamma; \Phi, \bar{\alpha} \vdash F]^\mathbf{Set} (\pi_i \rho) [\alpha := \bar{S}] \right) \overline{[\Gamma; \Phi \vdash A]^\mathbf{Rel} (\pi_i \rho')}$$

By the surjectivity of each $\pi_i : \mathbf{Rel} \rightarrow \mathbf{Set}$ we have that

$$\begin{aligned} & \pi_i \left(\left(\text{Lan}_{\lambda \bar{R}. [\Gamma; \bar{\alpha} \vdash K]^\mathbf{Rel} \rho [\alpha := \bar{R}]} \lambda \bar{R}. [\Gamma; \Phi, \bar{\alpha} \vdash F]^\mathbf{Rel} \rho [\alpha := \bar{R}] \right) \overline{[\Gamma; \Phi \vdash A]^\mathbf{Rel} f} \right) \circ j_{\bar{S}, \bar{g}} \\ &= j'_{\bar{S}, \overline{[\Gamma; \Phi \vdash A]^\mathbf{Rel} (\pi_i f) \circ g}} \end{aligned}$$

for every $\bar{S} : \mathbf{Set}_0$ and $\bar{g} : \overline{[\Gamma; \bar{\alpha} \vdash K]^\mathbf{Set} (\pi_i \rho) [\alpha := \bar{S}] \rightarrow [\Gamma; \Phi \vdash A]^\mathbf{Set} (\pi_i \rho')}$. That is,

$$\pi_i \left(\left(\text{Lan}_{\lambda \bar{R}. [\Gamma; \bar{\alpha} \vdash K]^\mathbf{Rel} \rho [\alpha := \bar{R}]} \lambda \bar{R}. [\Gamma; \Phi, \bar{\alpha} \vdash F]^\mathbf{Rel} \rho [\alpha := \bar{R}] \right) \overline{[\Gamma; \Phi \vdash A]^\mathbf{Rel} f} \right)$$

satisfies the universal property (Equation 8.2) of

$$(Lan_{\lambda \overline{S}. [\Gamma; \overline{\alpha} \vdash K]^{\text{Set}}(\pi_i \rho) [\overline{\alpha} := \overline{S}]} \lambda \overline{S}. [\Gamma; \Phi; \overline{\alpha} \vdash F]^{\text{Set}}(\pi_i \rho) [\overline{\alpha} := \overline{S}]) \overline{[\Gamma; \Phi \vdash A]^{\text{Set}}(\pi_i f)}}$$

The two expressions must therefore be equal, and thus Equation 8.9 holds. \square

The requirement that the types in \overline{K} be constant-free polynomials is quite restrictive. In fact, it precludes the expression of many GADTs commonly used in practice, such as the following GADT `Expr` of typed expressions:

```
data Expr (A : Set) : Set where
  const : Int → Expr Int
  is_zero : Expr Int → Expr Bool
  if : Expr Bool → Expr A → Expr A → Expr A
```

Even when extending along constant-free polynomials, it is unclear how to prove the Identity Extension Lemma for `Lan`-types or, indeed, whether it holds at all in their presence. To prove the IEL for `Nat`-types, for example, it is necessary to cut down the semantic interpretation by requiring that the natural transformations in the set interpretation preserve equalities. The impossibility of restricting the semantic interpretations of `Lan`-types, as shown in Proposition 8.2, suggests that the IEL may fail for them. We have been able neither to prove nor disprove it thus far, but our substantial and lengthy investigations into the issue lead us to conjecture that it is not possible to define a functorial semantics for `Lan`-types that gives rise to parametric models for languages supporting even constant-free polynomial primitive GADTs.

9. CONCLUSION AND DIRECTIONS FOR FUTURE WORK

We have constructed a parametric model for a calculus providing primitives for constructing nested types directly as fixpoints, rather than representing them via their Church encodings. We have also used the Abstraction Theorem for this model to derive free theorems for nested types. This was not possible before [JP19] because such types were not previously known to have well-defined interpretations in locally finitely presentable categories (here, `Set` and `Rel`), and, to our knowledge, no term calculus for them existed either. The key to obtaining our parametric model is the delicate threading of functoriality and its accompanying naturality conditions throughout our model construction.

We were surprised to find that, although GADTs were shown in [JP19] to have appropriately cocontinuous functorial semantics in terms of left Kan extensions, our model construction does not extend to give a parametric model for them. It may be possible to modify the categories of relations and relation transformers so that if (A, B, R) is a relation then $\pi_1 : R \rightarrow A$ and $\pi_2 : R \rightarrow B$ are always surjective; this would prohibit situations like that in Example 8.3, and might therefore allow us to recover our IEL and, ultimately, an Abstraction Theorem appropriate to GADTs. If this turns out to be possible, then generalizing the resulting model construction to locally λ -presentable categories for $\lambda > \omega$ would make it possible to handle broader classes of GADTs. (As shown in [JP19], $\lambda > \omega_1$ is required to interpret even common GADTs.) We could even attempt to carry out our construction in locally λ -presentable cartesian closed categories (lpcacs) \mathcal{C} whose categories of (abstract) relations, obtained by pullback as in [Jac99], are also lpcacs and are appropriately fibred over \mathcal{C} . This would give a framework for constructing parametric models for calculi with primitive GADTs that is based on locally λ -presentable fibrations, for some appropriate definition thereof.

The expressivity of folds for nested types has long been a vexing issue (see, e.g., [BM98]), and this is naturally inherited by the calculus presented here. Since it codes all recursion using standard folds, and since folds for nested types must return natural transformations, many standard functions over nested types cannot be represented in this calculus. Another important direction for future work is therefore to improve the expressivity of our calculus by adding, say, generalized folds [BP99],

or Mendler iterators [AMU05], or otherwise extending standard folds to express computations whose results are not natural transformations. In particular, we may wish to add term-level fixpoints as, e.g., in [Pit00]. This would require the categories interpreting types to be not just locally λ -presentable, but also to support some kind of domain structure. At the moment it seems that such an endeavor will have to wait for a generalization of the results presented here to at least locally ω_1 -presentable categories, however: ω -CPO, the most natural category of domains to replace **Set**, is not locally finitely presentable.

REFERENCES

- [AMU05] A. Abel, R. Matthes, and T. Uustalu. Iteration and coiteration schemes for higher-order and nested datatypes. *Theoretical Computer Science*, 333:3–66, 2005.
- [AR94] J. Adámek and J. Rosický. *Locally Presentable and Accessible Categories*. Cambridge University Press, 1994.
- [Atk12] R. Atkey. Relational parametricity for higher kinds. In *Computer Science Logic*, pages 46–61, 2012.
- [BFSS90] E. S. Bainbridge, P. J. Freyd, A. Scedrov, and P. J. Scott. Functorial polymorphism. *Theoretical Computer Science*, 70:35–64, 1990.
- [BM98] R. Bird and L. Meertens. Nested datatypes. In *Mathematics of Program Construction*, pages 52–67, 1998.
- [BM05] L. Birkedal and R. E. Møgelberg. Categorical models for Abadi and Plotkin’s logic for parametricity. *Mathematical Structures in Computer Science*, 15:709–772, 2005.
- [BP99] R. Bird and R. Paterson. Generalised folds for nested datatypes. *Formal Aspects of Computing*, 11:200–222, 1999.
- [Car84] L. Cardelli. Type systems. In *CRC Handbook of Computer Science and Engineering*, pages 2208–2236. CRC Press, 1984.
- [DR04] B. Dunphy and U. Reddy. Parametric limits. In *Logic in Computer Science*, pages 242–251, 2004.
- [Gir72] J.-Y. Girard. *Interprétation fonctionnelle et élimination des coupures de l’arithmétique d’ordre supérieur*. PhD thesis, University of Paris, 1972.
- [GJF⁺15] N. Ghani, P. Johann, F. Nordvall Forsberg, F. Orsanigo, and T. Revell. Bifibrational functorial semantics for parametric polymorphism. In *Mathematical Foundations of Program Semantics*, pages 165–181, 2015.
- [GLP93] A. Gill, J. Launchbury, and S.L. Peyton Jones. A short cut to deforestation. In *Functional Programming Languages and Computer Architecture*, pages 223–232, 1993.
- [Has94] R. Hasegawa. Categorical data types in parametric polymorphism. *Mathematical Structures in Computer Science*, 4:71–109, 1994.
- [Jac99] B. Jacobs. *Categorical Logic and Type Theory*. Elsevier, 1999.
- [JG08] P. Johann and N. Ghani. Foundations for structured programming with GADTs. In *Principles of Programming Languages*, pages 297–308, 2008.
- [JG10] P. Johann and N. Ghani. Haskell programming with nested types: A principled approach. *Higher-Order and Symbolic Computation*, 22(2):155–189, 2010.
- [JGJ21] P. Johann, E. Ghiorzi, and D. Jeffries. Parametricity for primitive nested types. In *Foundations of Software Science and Computation Structures*, 2021.
- [Joh02] P. Johann. A generalization of short-cut fusion and its correctness proof. *Higher-Order and Symbolic Computation*, 15:273–300, 2002.
- [JP19] P. Johann and A. Polonsky. Higher-kinded data types: Syntax and semantics. In *Logic in Computer Science*, pages 1–13, 2019.
- [JP20] P. Johann and A. Polonsky. Deep induction: Induction rules for (truly) nested types. In *Foundations of Software Science and Computation Structures*, 2020.
- [Mat11] R. Matthes. Map fusion for nested datatypes in intensional type theory. *Science of Computer Programming*, 76(3):204–224, 2011.
- [MG01] C. Martin and J. Gibbons. On the semantics of nested datatypes. *Information Processing Letters*, 80(5):233–238, 2001.
- [MR92] Q. Ma and J. C. Reynolds. Types, abstractions, and parametric polymorphism, part 2. In *Mathematical Foundations of Program Semantics*, pages 1–40, 1992.

- [Pit98] A. Pitts. Parametric polymorphism, recursive types, and operational equivalence. 1998.
- [Pit00] A.M. Pitts. Parametric polymorphism and operational equivalence. *Mathematical Structures in Computer Science*, 10:321–359, 2000.
- [Rey83] J. C. Reynolds. Types, abstraction, and parametric polymorphism. *Information Processing*, 83(1):513–523, 1983.
- [Rey84] J. C. Reynolds. Polymorphism is not set-theoretic. *Semantics of Data Types*, pages 145–156, 1984.
- [Rie16] E. Riehl. *Category Theory in Context*. Aurora, 2016.
- [RR94] E. Robinson and G. Rosolini. Reflexive graphs and parametric polymorphism. In *Logic in Computer Science*, pages 364–371, 1994.
- [Wad89] P. Wadler. Theorems for free! In *Functional Programming Languages and Computer Architecture*, pages 347–359, 1989.