NOTES ON FIBRATIONAL SEMANTICS

DANIEL GRATZER

Abstract. I have lately found myself reading a large number of papers which deal in some way with fibrations in category theory. Particularly to give semantics to a logic or type theory. In order to be sure that I've internalized these concepts in some small way I have written up some notes introducing fibrations.

It would be far too ambitious to claim the completeness of these notes (this would be closer to a dissertation [3] rather than a note). Rather, I shall aim for a level of completeness that let's me state and prove semantics for an interesting type theory. After that, I will gesture towards how the semantics might be extended for more featureful logics or type theories but avoid the same level of proof.

1. Introduction

Let us first start with a brief discussion motivating categorical semantics in general and fibrations in particular. Denotational semantics is a rich and well studied area of type theory dating back to work by Scott and Strachey. The core idea is to reduce the studying of programs to the study of better understood mathematical objects. Originally these tended to be things like *sets* or *continuous lattices*. As time has evolved though we have shifted to objects such as *games* or other exotic objects. Lately however the field of categorical semantics has exploded. Categorical semantics in some sense acts as glue between syntax and traditional denotational semantics.

With categorical semantics one doesn't use concrete. Instead, categorical semantics aim to axiomatize the structure necessary for any concrete model to be to possess. The classic example is that of a cartesian closed category (CCC). This possesses enough structure to model precisely the simply typed lambda calculus. Now the power of categorical semantics becomes apparent, by proving that we can model the simply typed lambda calculus in any CCC, we have suddenly reduced the problem of finding models to the problem of finding CCCs. This is a comparatively easy and well-studied problem. In fact, this immediately gives us dozens of models of the simply typed lambda calculus for free

(1) **Set**

(2) **Dom**

(3) **Eq**

(4) **PEq**

(5) ...

Date: May 8, 2017.

This illustrates the role of categorical semantics. They mediate between the purely syntactic demands of a type theory and the mathematically natural constraints that we need to impose on a category to obtain a model. By finding a convenient set of constraints for a model, models become cheap and easy to find.

This is the role of categorical semantics but what about fibrational semantics? Fibrations are the categorical version of indexing. Indexing is a prevalent phenomenon in type theory; terms depend on types, types may depend on other types, types may even depend on terms! It turns out that be starting with a notion of indexing and imposing constraints on the indexed and the indexing categories we can quickly and flexibly model type theory. Again we encounter the advantages of generality become apparent when working with fibrational models. By specifying the properties we require of a fibration to provide in order to support a model of a type theory, we generalize over even the variety of fibrations that support this. For instance, if a split fibration yields a model of predicate logic then we are not merely limited to one source of fibrations. Throughout these notes the generality that this provides will make it apparent how fibrational semantics generalize the ideas of models into locally cartesian closed categories, toposes, and regular categories.

2. An Introduction to Fibrations

Before diving immediately into the categorical formulation of fibrations I would like to start with the set-theoretic intuition. In set theory, families of sets are a common occurrence. They are usually denoted $(X_i)_{i\in I}$ so that for any element $i \in I$ we have a designated set X_i. This presentation is what we shall refer to as the *index-based* presentation of families of sets. An equivalent and useful way to view this family is as a function

$$\iota : X = \bigcup_{i\in I} X_i \to I$$

we will refer to X as the *total* set and I as the *index* set. The idea is that every element $x \in X_i$ instead becomes $(i, x) \in X$, a pair of the index at which it lives and the actual element. The map ι, the so called *presentation* map, will then send $(i, x) \mapsto i$. We can then easily recover our original X_is from ι.

$$X_i = \iota^{-1}(i)$$

For a user of set theory this presentation may seem slightly stilted but it does have the major advantage that it's purely based on maps between sets. This is something we can conceivably generalize to categories since categories possess this structure. The next question what properties of presentation maps to we wish to preserve when we categorify it.

One feature that will turn out to be of great importance is the *reindexing* of families of sets. For sets, if we have a map $f : I \to J$ and $(X_i)_{i\in I}$ we can conceivably form a family of sets which is indexed by J, $(Y_j)_{j\in J}$, where Y_j is defined as

$$Y_i = X_{f(i)}$$

This reindexing again only depends on maps between sets so it is quite natural to categorify. The first way to categorify this is to maps sets to objects and functions as morphisms. We

then define a family as a morphism $f : B \to A$. We generally will write this as B *over* A or pictorially

$$\begin{array}{c} B \\ \downarrow{\scriptstyle f} \\ A \end{array}$$

We cannot naturally recover what B_a might mean yet because we do not have the same notion of elements that we do in set theory. Instead, we will see later that this is a general result of reindexing.

Reindexing reappears here quite naturally in finite limit categories in the form of pullbacks. Specifically, if we cast a set theoretic function $f : X \to Y$ as a morphism $f : I \to J$, given a family $b : A \to I$ we can define the reindexing $f^*(b)$ as the pullback

$$\begin{array}{ccc} f^*(A) & \longrightarrow & A \\ {\scriptstyle f^*(b)}\downarrow & & \downarrow{\scriptstyle b} \\ I & \xrightarrow[f]{} & J \end{array}$$

Now that we have generalized the notion of indexing from sets, we can start to observe some of the categorical structures which arise from this *indexing-as-presentations* approach. First, families over A for some given A quite naturally form a category: $\mathbb{C}/A$.

Definition 2.1. *The* slice category *over A, written $\mathbb{C}/A$ is defined as the category with*

Objects: *Objects are families $f : B \to A$*

Morphisms: *A morphism between $f : B \to A$ and $g : C \to A$ is a morphism of $h \in \hom_{\mathbb{C}}(B, C)$ so that $f = g \circ h$.*

$$\begin{array}{ccc} B & \xrightarrow{\quad h \quad} & B \\ & {\scriptstyle f}\searrow \quad \swarrow{\scriptstyle g} & \\ & A & \end{array}$$

Now that we have categories for families over A, it is natural to ask how the reindexing maps f^* act on these categories. In particular, do these respect the structure of slice categories and thus form a functor. As it turns out, this is the case.

Theorem 2.2. *For any $f : A \to B$ in a finite limit category, reindexing along f gives a functor $f^* : \mathbb{C}/B \to \mathbb{C}/A$.*

Proof. TODO □

This sort of structure, categories of indexed families with reindexing functors between them will turn out to be a natural setting for models of type theory. To give a flavoring for why this is interesting, we will model each Δ as an object our category, each $\Delta \vdash A\,\mathrm{type}$ as a family over $[\![\Delta]\!]$ and terms as morphisms between these families. The reindexing functor correspond naturally to substitution. With this we reduce the problem of axiomatizing our

semantics to merely demanding the existence of certain categorical constructs in each slice category. In other words, we will use the categorified indexing to represent how types and terms depend on contexts. By abstracting away this structure now then, we can easily wipe away half the work of defining categorical semantics.

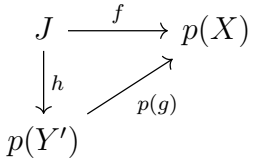
In order to generalize away from one particular form on indexing, we isolate the idea of having a having a family of categories varying over a base or index category. This is a fibration. I will start by giving the formal definition of a fibration which makes references to some as of yet undefined concepts. I will then progressively attempt to fill it in so that the original, formal, definition becomes more intuitive.

Definition 2.3. *A fibration, $p : \mathbb{E} \to \mathbb{B}$, so that for every morphism $u \in \hom_{\mathbb{B}}(I, p(Y))$ there is a cartesian lifting $\bar{u} : u^*(Y) \to Y$.*

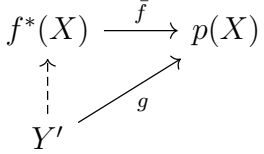
Intuitively a fibration is a way of gluing a collection of categories indexed by another category so we start with a *display functor* $p : \mathbb{E} \to \mathbb{B}$. We will think of $\mathbb{E}$ as the collection of all the objects and morphisms of our indexed categories clumped together and p tells us the index of some particular object. Accordingly, we can then say that a morphism, $f \in \hom_{\mathbb{E}}(A, B)$ is vertical if $p(f) = 1$. It is obvious that $\mathbb{E}_I$ comprised of objects X so that $p(X) = I$ and morphisms vertical over I forms a category. It is important that $\mathbb{E}$ is not necessarily just $\amalg_I \mathbb{E}_I$, the morphisms that are not vertical will important for relating the structure $\mathbb{B}$ to $\mathbb{E}_I$.

What we would like next is a way from moving between $\mathbb{E}_I$ and $\mathbb{E}_J$ using the information we have from $\hom_{\mathbb{B}}(J, I)$. This is just a generalization of what we did with slice categories to fibrations. In order to do this, given a map $f : J \to I$ we wish to find a way to transport from $\mathbb{E}_I$ to $\mathbb{E}_J$. In order to do this, we wish to lift the morphism in some way so that given X so that $p(X) = I$ we have a map $X \to Y$ for some Y so that $p(Y) = J$ that projects down onto f. If we think of this as substitution over contexts (like our above example) this roughly corresponds to saying that if we have a way to chance our context to a different context, we can change types in the first context to types in the second. Something that seems quite plausible, it ought to be just substitution on types.

It would be ideal if this morphism was canonical in some sense. It's quite easy to imagine a scenario where there are infinitely many different morphisms that we could lift to. If we think of this operation as corresponding to substitution, then there must be a canonical choice for us to pick. Now for some notation, we write this hypothetical *best substitution* as a morphism in $\mathbb{E}$, $\bar{f} : f^*(X) \to X$ in order to evoke the similarity to the pullback case we started with. As is usual in category theory, we will characterize this lifting using a universal property. Namely, we want this morphism to be terminal. This means that given



we want a canonical map



in this case, our lifting $\bar{u}$ is said to be *cartesian*.

Definition 2.4. *A map $f : X \to Y$ is said to be cartesian over $u : f(Y) \to f(X)$ if $p(f) = u$ and if for any map $g : Y' \to X$ so that $u \circ v = p(g)$, there is a unique $h : Y' \to Y$ so that $p(h) = v$ and $f \circ h = g$.*

With this we have at least acquired the necessary technical information to understand what a fibration is, however we're still lacking in some intuition. In particular, the connection between cartesian maps and reindexing functors remains unclear. What we really wanted was a family of functors u^* for each $u : J \to I$ mapping $\mathbb{E}_I$ to $\mathbb{E}_J$. Well as it happens, this functor is definable using cartesian maps.

Theorem 2.5. *For a fibration $p : \mathbb{E} \to \mathbb{B}$, any map $u : J \to I$ in $\mathbb{B}$ induces a functor $u^* : \mathbb{E}_I \to \mathbb{E}_J$.*

Proof. todo □

In order to see that this characterization is even equivalent to this sort of indexing structure, we will next investigate the so called *Grothendieck construction*. This allows us to turn a pseudofunctor $\mathbb{B} \to \mathbf{Cat}$ into a particular fibration. This lets us know that we haven't lost or gained anything essential by switching to the fibrational presentation of indexing.

Theorem 2.6. *Every pseudofunctor $\mathbb{B} \to \mathbf{Cat}$ corresponds to a fibration over $\mathbb{B}$.*

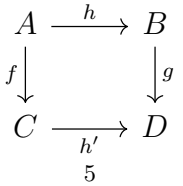
Proof. todo □

Having set out this very abstract definition, I will now relate it to what we were previously studying: indexed families of sets. In particular, let us look at the so called *codomain fibration*: Cod. In order to do this, let us start with a few preliminary definitions.

Definition 2.7. *The arrow category over a category, $\mathbb{C}$, is written $\mathbb{C}^{\to}$ and is defined with*

Objects: *For objects arrows $f : A \to B$*

Morphisms: *A morphism between $f : A \to B$ and $g : C \to D$ is a pair of morphisms (h, h') so that $h : A \to C$ and $h' : B \to D$ and the following commutes*



The reader is referred to Mac Lane [6] for a more detailed investigation of the arrow category. For our purposes we are largely interested in the codomain functor $\mathrm{Cod} : \mathbb{C}^{\to} \to \mathbb{C}$. It is defined

$$\mathrm{Cod}(f : A \to B) = B$$
$$\mathrm{Cod}(h, h') = h'$$

it is trivial to check this indeed defines a functor. More than this though, in a category with pullbacks this functor is in fact a fibration.

Example 2.8. *The* Cod *functor defines a fibration.*

Proof. todo □

Now an important step of understanding a fibration $p : \mathbb{E} \to \mathbb{B}$ is to study the fibers that it defines, $\mathbb{E}_I$. In particular, we shall find this illuminating in the case of the codomain fibration

Lemma 2.9. *The fibers of* $\mathrm{Cod} : \mathbb{C}^{\to} \to \mathbb{C}$ *are precisely the slice categories of* $\mathbb{C}$*. Moreover, reindexing in the fibration corresponds to the pullback functor of slice categories.*

Proof. todo □

Have done this we have come full circle: our very abstract notion of fibration can be specialized to simple set theoretic indexing by instantiating a particular fibration on a particular category.

Having shown that our abstraction is a useful one, we can develop the theory of fibrations abstractly. The reader is referred to Jacobs [4] for a more in-depth coverage of fibered category theory. For our purposes, I only want to flesh out enough theory to explain the semantics of System F in the next section.

The reindexing functors are our first object of consideration. In particular, it is worth mentioning a few details that I would like to promptly handwave away. Firstly, the definition of a fibration stipulates the existence of liftings but does not include the data of which particular liftings we chose. If one wants to model this set theoretically, one needs to demand the axiom of choice or select a particular collection of liftings. The alternative is to not use a broken notion of existential quantification but that seems less likely to see widespread adoption. In any case, a fibration is said to be *cloven* if it comes equipped with a choice of liftings. Next the liftings themselves give rise to reindexing functors as we have seen. However, there is a question as to the coherence of these functors. For instance, we have seen that $u^* \circ v^* \cong (v \circ u)^*$ and $1 \cong 1^*$. In a fibration if our choice of liftings makes these equivalences simply identities, we shall call the fibration *split*. Now it is the case that every cloven fibration is equivalent to a split one so I shall for the rest of this paper assume that we are working with split cloven fibrations unless stated otherwise.

The next area to explore is the structure that may be found in each fiber. In particular, we might want to demand the usual categorical constructs: limits, colimits, exponentials, etc. It turns out to be straightforward to add these to our model.

Definition 2.10. *A fibration $p : \mathbb{E} \to \mathbb{B}$ is said to posses* fiberwise X *if each fiber $\mathbb{E}_I$ possesses Xs and Xs are preserved by reindexing. We will say that this structure is* split *if it is preserved on the nose by reindexing.*

Fiberwise structure, reverting to our intuition that fibrations are indexed categories, is just the assertion that each $\mathbb{C}_i$ possess exponentials, products, or some other structure. It's not hard to see that the examples we've considred thus far has some examples.

Example 2.11. *The* $\mathrm{Cod} : \mathbb{C}^{\to} \to \mathbb{C}$ *fibration has all finite limits if $\mathbb{C}$ is a finite limit category.*

Proof. todo □

Another important piece of categorical logic that we shall need for fibrations is that of quantifiers as adjoints. This is explained in Awodey [1] and originated with Lawvere [5]. Here we shall give a brief summary for how the relate to fibrations. Recall that we like to intuitively view $\mathbb{E}_I$ as the category of formulas in context I, that is if $A \in \mathbb{E}_I$, we'd like to imagine it as the denotation $I \vdash A$. First, let us note that there seems to be a bijection between proofs

$$\Gamma \vdash \forall A.\ B \iff \Gamma, A \vdash B$$

The direction from right to left is simply the introduction introduction rule

$$\frac{\Gamma, x : A \vdash B}{\Gamma \vdash \forall x : A.\ B(x)}$$

Going from left to right on the other hand is a slightly contorted usage of the elimination rule (factoring out an application of cut). This is interesting because it's not at all obvious how to define a quantifier categorically: the heavy reliance on points/variables presents a problem. It's quite clear, however, how to formalize an adjoint. Running with this intuition, we claim

$$\pi_1^* \dashv \forall^*$$

This is because weakening is precisely given by reindexing along π_1 in a fibration. Similarly, existential quantification forms a left adjoint by duality

$$\forall^* \dashv \pi_1^*$$

This is similar to how, for locally cartesian closed categories we have $\Sigma_f \dashv f^* \dashv \Pi_f$ and just like in that situation, these adjoints are crucial for modeling quantifiers in type theory. Confusingly, these adjoints are called (co)products. They should not be confused with *fiberwise* (co)products!

Definition 2.12. *A fibration $p : \mathbb{E} \to \mathbb{B}$ is said to have (co)products when the reindexing of projection functions $\pi^* : \mathbb{E}_I \to \mathbb{E}_{I \times J}$ has a right (respectively left) adjoint.*

We shall usually notate this adjoints as $\amalg_f \dashv f^* \dashv \Pi_f$ rather than $\exists$ and $\forall$ to avoid stick to the notation used in literature.

The final piece of structure that we shall need for our model is something to handle the impredicative polymorphism of System F. For this, we need an object in our index category which classifies types in the total category. This is called a *generic object*.

Definition 2.13. *A generic object in a fibration $p : \mathbb{E} \to \mathbb{B}$ if an object $T \in \mathbb{E}$ so that for any $X \in \mathbb{E}$, there is a unique map $u : p(X) \to p(T)$ so that there exists an $f : X \to T$ so that f is cartesian 2.4 over u.*

The intuition here is that given any type, represented as an object in the category $\mathbb{E}$, we can find a unique element $p(X) \to p(T)$ so that applying this substitution to T gives us back the original type we started with. This tells us that cartesian liftings of $\hom(p(X), p(T))$ are in bijection with the objects of $\mathbb{E}_X$. This will be used to model a "type of small types" for impredicative polymorphism.

3. Semantics for System F

Let us start with a simple definition of System F. To be more explicit than usual, let us start by noting the three core judgments of the system.

$$\Delta \vdash \Gamma \,\mathrm{ctx} \qquad \Delta \vdash \tau \,\mathrm{type} \qquad \Delta; \Gamma \vdash e : \tau$$

The judgment for $\Delta \vdash \Gamma \,\mathrm{ctx}$ is a simple extension of the typehood judgment.

$$\frac{}{\Delta \vdash \cdot \,\mathrm{ctx}} \qquad \frac{\Delta \vdash \Gamma \,\mathrm{ctx} \qquad \Delta \vdash \tau \,\mathrm{type}}{\Delta \vdash \Gamma, x : \tau \,\mathrm{ctx}} \qquad \frac{\Delta, \alpha \vdash \tau \,\mathrm{type}}{\Delta \vdash \forall \alpha.\ \tau \,\mathrm{type}} \qquad \frac{\Delta \vdash \tau_1 \,\mathrm{type} \qquad \Delta \vdash \tau_2 \,\mathrm{type}}{\Delta \vdash \tau_1 \to \tau_2 \,\mathrm{type}}$$

$$\frac{\alpha \in \Delta}{\Delta \vdash \alpha \,\mathrm{type}}$$

Next we have the usual typing rules for terms. One will observe that I deviate from some presentations of System F by using two contexts to represent types *and* terms rather than blending them into one context. This is a nod towards the semantics where such things will be fundamentally separated anyways, with one living in the base category and one living in the fibers above it.

$$\frac{x : \tau \in \Gamma}{\Delta; \Gamma \vdash x : \tau} \qquad \frac{\Delta; \Gamma, x : \tau_1 \vdash e : \tau_2}{\Delta; \Gamma \vdash \lambda x{:}\tau.\ e : \tau_1 \to \tau_2} \qquad \frac{\Delta; \Gamma \vdash e_1 : \tau_1 \to \tau_2 \qquad \Delta; \Gamma \vdash e_2 : \tau_1}{\Delta; \Gamma \vdash e_1\ e_2 : \tau_2}$$

$$\frac{\Delta, \alpha; \Gamma \vdash e : \tau}{\Delta; \Gamma \vdash \Lambda x.\ e : \forall \alpha.\ \tau} \qquad \frac{\Delta; \Gamma \vdash e : \forall \alpha.\ \tau_1 \qquad \Delta \vdash \tau_2 \,\mathrm{type}}{\Delta; \Gamma \vdash e[\tau_2] : [\tau_2/\alpha]\tau_1}$$

Having fleshed out the objects of our language we now turn to the semantics of our language. They will proceed in three parts. First we isolate the specific variety of fibration that we will be studying. Second we will denote the terms and contexts into objects in base category. Finally, we will denote terms into morphisms in the fibers above contexts. Put briefly

Type contexts: objects in the base category.

Types: morphisms in the base category into the generic object from the type context. Equivalently, objects in the fiber above the type context.

Term Contexts: objects in the fiber above a type context.

Terms: morphisms between term contexts. Equivalently, vertical morphisms in the fiber above the type context.

With this plan in mind, we are in a position to define what structure we will need for our fibration.

Definition 3.1. *A polymorphic fibration, $p : \mathbb{E} \to \mathbb{B}$ is a fibration with*

(1) *Fiberwise finite products and exponentials.*

(2) *Finite products in the base category.*

(3) *A generic object T.*

(4) *Right adjoints to the reindexing functors for projections.*

For now we will set aside the question of what fibrations exist which satisfy these constraints and focus instead show that this is sufficient to model System F. First, we present the denotation of type context.

$$\begin{aligned} [\![\cdot]\!] &= 1 \\ [\![\Delta, \alpha]\!] &= [\![\Delta]\!] \times p(T) \end{aligned}$$

This is just the straightforward mapping based on the insight that maps into $p(T)$ will classify types via transposition. That is, we know that there is a bijection $\phi_I : \hom(I, p(T)) \cong \mathrm{obj}(\mathbb{E}_I)$. Next comes the denotation of types.

$$\begin{aligned} [\![\Delta \vdash \alpha \,\mathrm{type}]\!] &= \pi_\alpha \\ [\![\Delta \vdash \tau_1 \to \tau_2 \,\mathrm{type}]\!] &= \phi^{-1}\left(\phi([\![\Delta \vdash \tau_2 \,\mathrm{type}]\!])^{\phi([\![\Delta \vdash \tau_1 \,\mathrm{type}]\!])}\right) \\ [\![\Delta \vdash \forall \alpha.\ \tau \,\mathrm{type}]\!] &= \phi^{-1}\left(\Pi_\pi \phi([\![\Delta, \alpha \vdash \tau_1 \,\mathrm{type}]\!])\right) \end{aligned}$$

Here π_α is the projection extraction $\alpha \in \Delta$ from the product $[\![\Delta]\!]$. Notice the use of ϕ to make use of the structure that we have at the level of fibers in the base category. This ability to tie together the base and total category is a common usage of generic objects. An important lemma that we shall make use of is that implicit weakening of typing derivation corresponds to categorical weakening (reindexing by π^*).

Lemma 3.2. *$[\![\Delta, \alpha \vdash \tau \,\mathrm{type}]\!]$ factors through $[\![\Delta \vdash \tau \,\mathrm{type}]\!]$ by $\pi_1 : [\![\Delta]\!] \times p(T) \to [\![\Delta]\!]$.*

Proof. todo □

Now by the definition 2.13 we know that this means precisely that

$$\pi_1^*(\phi([\![\Delta \vdash \tau \,\mathrm{type}]\!])) = \phi([\![\Delta, \alpha \vdash \tau \,\mathrm{type}]\!])$$

so weakening does precisely correspond to categorical weakening. This is important because we want to use the adjunction $\pi^* \dashv \Pi_\pi$. However, in order to do this we need a morphism with domain $\pi^*(A)$ for some A. The above lemma tells us that when it's $\phi([\![\mathit{typeJ}\Delta, \alpha\tau]\!])$ where we happen to know that $\Delta \vdash \tau$ type holds then this is the case.

Now under ϕ, we know that the denotation $[\![\Delta \vdash A\,\mathrm{type}]\!]$ corresponds to an object in the fiber above $[\![\Delta]\!]$. Using this, we shall denote terms $\Delta; \Gamma \vdash e : \tau$ into morphisms in

$$\hom_{\mathbb{E}_{[\![\Delta]\!]}}([\![\Delta \vdash \Gamma\,\mathrm{ctx}]\!], \phi([\![\Delta \vdash A\,\mathrm{type}]\!]))$$

This is defined, as is standard, by induction on the typing derivation. In order to do this, let us shall write ψ_A for the natural bijection $\hom(\pi^*(A), B) \cong \hom(A, \Pi B)$

$$\begin{aligned}
[\![\Delta; \Gamma \vdash x : \tau]\!] &= \pi_x \\
[\![\Delta; \Gamma \vdash \lambda x{:}\tau_1.\ e : \tau_1 \to \tau_2]\!] &= \Lambda([\![\Delta; \Gamma, x : \tau_1 \vdash e : \tau_2]\!]) \\
[\![\Delta; \Gamma \vdash e_1\ e_2 : \tau_2]\!] &= \mathrm{ev} \circ \langle [\![\Delta; \Gamma \vdash e_1 : \tau_1 \to \tau_2]\!], [\![\Delta; \Gamma \vdash e_2 : \tau_1]\!] \rangle \\
[\![\Delta; \Gamma \vdash \Lambda\alpha.\ e : \forall\alpha.\ \tau]\!] &= \psi([\![\Delta, \alpha; \Gamma \vdash e : \tau]\!]) \\
[\![\Delta; \Gamma \vdash e[\tau_1] : [\tau_1/\alpha]\tau_2]\!] &= \langle 1, [\![\Delta \vdash \tau_1\,\mathrm{type}]\!] \rangle^*([\![\Delta, \alpha; \Gamma \vdash e : \forall\alpha.\ \tau_2]\!])
\end{aligned}$$

Again we are using π_x to refer to the projection removing $[\![\Delta \vdash \tau\,\mathrm{type}]\!]$ from $[\![\Delta \vdash \Gamma\,\mathrm{ctx}]\!]$. Notice that we are making use of lemma 3.2 in order to apply the transposition ψ above. We now want to show soundness, that is, that $[\![-]\!]$ respects $\beta\eta$-equivalence. Let us first explicitly spell out the rules that we expect to hold.

$$\frac{\Delta; \Gamma \vdash e : \tau}{\Delta; \Gamma \vdash e \equiv e : \tau} \qquad \frac{\Delta; \Gamma \vdash e_1 \equiv e_2 : \tau}{\Delta; \Gamma \vdash e_2 \equiv e_1 : \tau} \qquad \frac{\Delta; \Gamma \vdash e_1 \equiv e_2 : \tau \qquad \Delta; \Gamma \vdash e_2 \equiv e_3 : \tau}{\Delta; \Gamma \vdash e_1 \equiv e_3 : \tau}$$

$$\frac{}{\Delta; \Gamma \vdash \lambda x{:}\tau_1.\ e\ x \equiv e : \tau_1 \to \tau_2} \qquad \frac{}{\Delta; \Gamma \vdash (\lambda x{:}\tau_1.\ e_1)\ e_2 \equiv [e_2/x]e_1 : \tau}$$

$$\frac{}{\Delta; \Gamma \vdash \Lambda\alpha.\ e[\alpha] \equiv e : \forall\alpha.\ \tau} \qquad \frac{}{\Delta; \Gamma \vdash (\Lambda\alpha.\ e)[\tau] \equiv [\tau/\alpha]e : \tau}$$

Now the first three equivalences are trivial to validate because the ambient equality in a category is an equivalence relation. Therefore, the other four rules are the interesting ones to prove.

Theorem 3.3. *If $\Delta; \Gamma \vdash e_1 \equiv e_2 : \tau$ then $[\![\Delta; \Gamma \vdash e_1 : \tau]\!] = [\![\Delta; \Gamma \vdash e_2 : \tau]\!]$ as morphisms in the fiber above $[\![\Delta]\!]$.*

Proof. todo □

Now that we know that these fibrations will give us a sound model of System F, we turn to the question of what examples exist in the wild. For the sake of concision, I will only present one example in depth. That the fibration of families of PERs gives a model. Firstly, we start by defining the relevant constructions.

Definition 3.4. *An assembly, X,* [1] *is a pair of a set $|X|$, and a map $E : X \to \mathcal{P}(\mathbb{N})$ so that for every $x \in X$, $E(x) \neq \emptyset$.*

We shall say that $n \in E(x)$ *realizes* x and write this more succinctly as $n \Vdash_X e$. It will be helpful to remember that we can encode Turing Machines[2] as natural numbers. This surjective encoding gives rise to a partial application operation, $\cdot$ where $n \cdot m = k$ if and only

[1] We will specialize this definition to $\mathcal{K}_1$ for simplicity but an arbitrary PCA suffices

[2] Or lambda terms if one wishes to be a bit more civilized.

if the nth Turing machine when fed the input m returns k. We can say that a realizer *tracks* a map $n \Vdash f$ if

$$m \Vdash x \implies n \cdot m \Vdash f(x)$$

This definition lets us form a categorical structure from assemblies.

Definition 3.5. *The category* **Asm** *has assemblies as objects and morphisms are set theoretic maps that are tracked some realizer.*

Theorem 3.6. *The* **Asm** *forms a regular category.*

Proof. For a full proof see van Oosten [9]. I will present the case for exponentials as it's a good illustration of the structure of **Asm**. □

Next we shall introduce a corresponding notion of *families* for assemblies. Such a notion also is intended to capture the computability aspects present in assemblies making them *uniform*. We shall specialize this to partial equivalence families, PERs, on natural numbers. It is not hard to show that these correspond to assemblies for which $x \Vdash a, b$ implies that $a = b$.

Definition 3.7. *A uniform family over an assembly, A, is an indexed family of partial equivalence relations on $\mathbb{N}$ denoted $(R_a)_{a \in A}$. A morphism from $(A, (R_a)_{a \in A})$ to $(B, (S_b)_{b \in B})$ is a pair of morphisms (u, f) so that $u \in \hom_{\mathbf{Asm}}(A, B)$ and a morphism f so $f_a : R_a \to S_{f(a)}$. Moreover, f must be uniformly tracked. This means that there is an e so that $n \Vdash a$ implies $e \cdot n \Vdash f_a$.*

Theorem 3.8. $\pi_1 : \mathbf{UFam} \to \mathbf{Asm}$ *is a polymorphic fibration.*

Proof. todo □

As a final note for this section, consider the subobject fibration of a topos. It is well known that there is no naive set theoretic model of System F [8]. However, it would seem that every topos gives rise to a model of System F. So, where in is the contradiction? The answer is the critical assumption of Reynolds [8] that the model isn't degenerate in that $[\![\lambda x{:}\tau_1.\ \lambda y{:}\tau_1.\ x]\!] \neq [\![\lambda x{:}\tau_1.\ \lambda y{:}\tau_1.\ y]\!]$. However, this will simply not be the case in any subobject fibration. TODO: expand. like a lot.

4. Extending Fibrational Semantics to Other Theories

5. Fibrational Semantics Compared to Other Approaches

6. Conclusion

References

[1] Steve Awodey. *Category Theory*. Oxford Logic Guides. Ebsco Publishing, 2006.

[2] Neil Ghani, Patricia Johann, Fredrik Nordvall Forsberg, Federico Orsanigo, and Tim Revell. Bifibrational functorial semantics of parametric polymorphism. *Electron. Notes Theor. Comput. Sci.*, 2015.

[3] B. Jacobs. *Categorical type theory*. PhD thesis, University of Nijmegen, 1991.

[4] B. Jacobs. *Categorical Logic and Type Theory*. Studies in logic and the foundations of mathematics. Elsevier, 2001.
[5] F. William Lawvere. Adjointness in Foundations. *Dialectica*, 1969.
[6] Saunders Mac Lane. *Categories for the Working Mathematician*. Graduate Texts in Mathematics. Springer New York, 1998.
[7] Andrew Pitts. Polymorphism is set theoretic, constructively. In *Category Theory and Computer Science, Proc. Edinburgh 1987*, 1987.
[8] John C. Reynolds. Polymorphism in not set-theoretic. Lecture notes in computer science, 1984.
[9] Jaap van Oosten. *Realizability: An Introduction to its Categorical Side*. Studies in Logic and the Foundations of Mathematics. 2008.