THE SYNTHETIC THEORY OF POLYNOMIAL FUNCTORS

REED MULLANIX, VERITY SCHEEL, AND SOLOMON BOTHWELL

1. POLYNOMIALS

We omit the standard rules for Martin Löf Type Theory with Tarski Universes. For simplicity, we shall assume that the universe $\mathcal{U}$ codes itself. This inconsistency can be resolved by instead using a tower of universes. We start by adding a new judgement $\Gamma \vdash A\ Poly$, which denotes that A is a polynomial functor. It has the following formation rule:

POLY-FORMATION
$$\frac{\Gamma \vdash A\ Type \qquad \Gamma, a : A \vdash B\ Type}{\Gamma \vdash (a : A) \times B\ a\ Poly}$$

We also add the following rules for obtaining the base of a polynomial, along with the fibre of a polynomial at a base.

POLY-BASE
$$\frac{\Gamma \vdash P\ Poly}{\Gamma \vdash Base\ P\ Type}$$

POLY-FIBRE
$$\frac{\Gamma \vdash P\ Poly \qquad \Gamma \vdash i : Base\ P}{\Gamma \vdash Fib\ P\ i\ Type}$$

The base and fibre operations do what one would expect when applied to a concrete polynomial.

POLY-BASE-DECODING
$$\frac{}{Base\ (a : A) \times B\ a \equiv A}$$

POLY-FIBRE-DECODING
$$\frac{\Gamma \vdash A\ Type \qquad \Gamma, a : A \vdash B\ Type \qquad \Gamma \vdash i : Base\ (a : A) \times B\ a}{Fib\ ((a : A) \times B\ a)\ i \equiv B\ i}$$

2. MORPHISMS OF POLYNOMIALS

We proceed by defining a type of morphisms between polynomials.

HOM-FORMATION
$$\frac{\Gamma \vdash P\ Poly \qquad \Gamma \vdash Q\ Poly}{\Gamma \vdash P \Rightarrow Q\ Type}$$

2.1. Hom Environments. The introduction and elimination rules for this type are the source of all of the power *and* complexity of the theory, so we will need to build out some machinery before we present them. To start, we define hom environments.

HOM-ENV-EMPTY

$$\frac{}{\Gamma \vdash \cdot \ HomEnv}$$

HOM-ENV-BIND

$$\frac{\Gamma \vdash \rho \ HomEnv \qquad a^- \text{ is a name}}{\Gamma \vdash \rho, a^- \ HomEnv}$$

HOM-ENV-SET

$$\frac{\Gamma \vdash \rho \ HomEnv \qquad \Gamma \vdash v : A \qquad a^- \text{ is a name}}{\Gamma \vdash \rho, a^- := v \ HomEnv}$$

We shall gloss over any difficulties with name freshness, and gesture vaguely in the direction of nameless representations if pressed. We also define a meta-operation $\rho[a^- := v]$, which replaces an a^- in an environment ρ with $a^- := v$. This is defined by induction over the environment. Furthermore, note that the rule for setting a binding in an environment ρ does *not* allow for the the value v to depend on earlier entries in ρ.

We now define all of the normal judgments relative to some environment ρ.

TYPE-IN-ENV

$$\frac{\Gamma \ Ctx \qquad \Gamma \vdash \rho \ HomEnv}{\Gamma \mid \rho \vdash A^+ \ Type}$$

TERM-IN-ENV

$$\frac{\Gamma \ Ctx \qquad \Gamma \vdash \rho \ HomEnv \qquad \Gamma \mid \rho \vdash A^+ \ Type}{\Gamma \mid \rho \vdash a : A^+}$$

TYPE-EQ-IN-ENV

$$\frac{\Gamma \ Ctx \qquad \Gamma \vdash \rho \ HomEnv \qquad \Gamma \mid \rho \vdash A^+ \ Type \qquad \Gamma \mid \rho \vdash B^+ \ Type}{\Gamma \mid \rho \vdash A^+ \equiv B^+ \ Type}$$

TERM-EQ-IN-ENV

$$\frac{\Gamma \ Ctx}{\Gamma \vdash \rho \ HomEnv \qquad \Gamma \mid \rho \vdash A \ Type \qquad \Gamma \mid \rho \vdash x \ Type : A^+ \qquad \Gamma \mid \rho \vdash y \ Type : A^+}{\Gamma \mid \rho \vdash x \equiv y : A^+}$$

We can read variables from an environment.

HOM-ENV-READ

$$\frac{a^- \in \rho}{\Gamma \mid \rho \vdash read \ a^- : A^+}$$

All of the rules for types in the base theory are copied over to the environment relative ones. We shall omit these for brevity, though we shall highlight the fact that we have an environment relative conversion rule.

CONVERSION-IN-ENV

$$\frac{\Gamma \ Ctx \qquad \Gamma \vdash \rho \ HomEnv}{\Gamma \mid \rho \vdash A \ Type \qquad \Gamma \mid \rho \vdash B \ Type \qquad \Gamma \mid \rho \vdash A \equiv B \ Type \qquad \Gamma \mid \rho \vdash x : B}{\Gamma \mid \rho \vdash x : A}$$

2.2. Hom Contexts and Sinks. We now give rules for linear Hom contexts, as well as a judgement for deriving "sink" terms from linear contexts.

HOM-CONTEXT-EMPTY

$$\overline{\Gamma \mid \rho \vdash \cdot \; HomCtx}$$

HOM-CONTEXT-BIND

$$\frac{\Gamma \mid \rho \vdash \Psi \; HomCtx \qquad \Gamma \mid \rho \vdash A \; Type}{\Gamma \mid \rho \vdash \Psi, a^- : A \; HomCtx}$$

SINK-TERM

$$\frac{\Gamma \; Ctx \qquad \qquad \qquad \qquad \qquad}{\Gamma \vdash \rho \; HomEnv \qquad \Gamma \mid \rho \vdash \Psi \; HomCtx \qquad \Gamma \vdash A \; Type \qquad \Gamma \vdash \theta \; HomEnv}{\Gamma \mid \rho \mid \Psi \vdash a^- : A^- \rightsquigarrow \theta}$$

Linearity of the Hom contexts is ensured by the following rule. The intuition here is that in order to discharge an obligation a^-, we are required to provide it with a value of the appropriate type.

SINK-LINEAR-VAR

$$\frac{\Gamma \vdash a^+ : A}{\Gamma \mid \rho \mid a^- : A^- \vdash a^-[a^+] : A^- \rightsquigarrow a^- := a^+}$$

2.3. Hom Expressions. We proceed by defining a new judgement describing hom expressions.

HOM-EXPR

$$\frac{\Gamma \; Ctx \qquad \Gamma \vdash \rho \; HomEnv \qquad \Gamma \mid \rho \vdash \Psi \; HomCtx \qquad \Gamma \vdash Q \; Poly}{\Gamma \mid \rho \mid \Psi \vdash \phi \; Hom(Q)}$$

HOM-SET

$$\frac{\Gamma \vdash a^+ : A \qquad \Gamma, x : A \mid \rho \mid \Psi_1 \vdash b^- : B^- \rightsquigarrow \theta \qquad \Gamma, \rho \cup \theta[a^+/x], \Psi_2 \vdash \phi \; Hom(Q)}{\Gamma \mid \rho \mid \Psi_1, \Psi_2 \vdash a^+ \rightarrow (\lambda x.b^-); \phi \; Hom(Q)}$$

HOM-DONE

$$\frac{\Gamma \vdash q^+ : Base \; Q \qquad \Gamma, x : Fib \; Q \; q^+ \mid \rho \mid \Psi \vdash q^- : A^-}{\Gamma \mid \rho \mid \Psi \vdash (q^+, \lambda x.q^-) \; Hom(Q)}$$

Note that we lambda abstract a variable x when defining the sink expression. This is not essential, but allows us to avoid writing a point-free mess. This also removes the need for a separate sink application rule, as it can be handled by function application.

2.4. Hom Introduction and Elimination.

HOM-INTRO

$$\frac{\Gamma, p^+ : Base \; p \mid \cdot \mid p^- : Fib \; P \; p^+ \vdash \phi \; Hom(Q)}{\Gamma \vdash \lambda p^+ p^- \rightsquigarrow \phi : P \Rightarrow Q}$$

HOM-ELIM

$$\frac{\Gamma \vdash f : P \Rightarrow Q \qquad \Gamma \vdash p^+ : Base \; P}{\Gamma \vdash f p^+ : (q^+ : Base \; Q) \times (Fib \; Q \; q^+ \to Fib \; P \; p^+)}$$

3. Sink Elimination

We need to also have rules for eliminating sink values in hom expressions.

Sigma-sink-elim
$$\frac{\Gamma, x : (a : A) \times B\ a \mid \rho \mid \Psi_1 \vdash c^- : C^- \rightsquigarrow \theta \qquad \Gamma \mid \rho, a^-, b^- \mid \Psi_2, a^- : A, b^- : B(read\ a^-) \vdash \phi\ Hom(Q)}{\Gamma \mid \rho \mid \Psi_1 \vdash \mathsf{unpack}\ ab^- \to (a^-, b^-); \phi\ Hom(Q)}$$

How do we handle the θ here? I want to instantiate it with $(read\ a^-, read\ b^-)$, but that's not allowed currently, as can't have read chains in ρ.