Isbell Duality for Refinement Types

Paul-André Melliès
CNRS, Université Paris Diderot
Sorbonne Paris Cité
mellies@pps.univ-paris-diderot.fr

Noam Zeilberger
MSR-Inria Joint Centre
noam.zeilberger@gmail.com

Abstract—Any refinement system (= functor) has a full and faithful embedding into the refinement system of presheaves over categories, by a simultaneous generalization of the Yoneda embedding and of the slice representation of a category. Motivated by an analogy between side-effects in programming and “context effects” in proof theory, we study logical aspects of this embedding, as well as of an associated “opposite embedding” (= embedding of the opposite functor). We expose a duality between these two embeddings mediated by negation into a “universal refinement” (a presheaf associated with the original refinement system), and use this to derive two interesting formulas for the embedding of a pushforward, which decompose it either as the negation of a pullback of a negation, or as the double-negation of a pushforward.

I. INTRODUCTION

This paper continues the study of type systems from the perspective outlined in [5]. There, we suggested that it is useful to view a type system as a functor from a category of typing derivations to a category of underlying terms, and that this can even serve as a working definition of “type system” (or what we call a refinement system), as being (in the most general case) simply an arbitrary functor.

Definition 1. A (type) refinement system is a functor p : D → T.

Definition 2. We say that an object S ∈ D refines an object A ∈ T (notated S ⊑ A) if p(S) = A.

Definition 3. A typing judgment is a triple (S, f, T) such that S ⊑ dom(f) and T ⊑ cod(f) (notated S ⊑ f T). In the special case where f = id this is also called a subtyping judgment (notated S ⊑ T).

Definition 4. A derivation of a typing judgment (S, f, T) is a morphism α : S → T in D such that p(α) = f (notated S ⊢ f T).

It is useful to think of any category C as a trivial example of a refinement system by the unique functor !C : C → 1 to the terminal category, and in the sequel we will describe several general constructions on refinement systems, which reduce to classical constructions on categories seen as such degenerate refinement systems. Another important (and more guiding) example of a refinement system is Hoare logic: take T to be a category with a single object W representing the state space and morphisms c : W → W corresponding to state transformers, and then define D and p : D → T so that refinements φ, ψ ⊑ W are predicates over the state space, and a derivation of a typing judgment φ ⊢ ψ corresponds exactly to a verification of a Hoare triple |φ|c|ψ| (asserting that c will take any state satisfying φ to a state satisfying ψ). What this latter example highlights is that a typing judgment can describe not just a logical entailment but also a side-effect (namely, the transformation c upon the state).

In fact, one of our original motivations for studying this framework actually came from apparent connections between Hoare logic and proof theory, and especially the proof theory of linear logic [1]. Consider for example the right-rule for multiplicative conjunction (“tensor”) in intuitionistic linear logic:

\[
\Gamma \vdash \phi \quad \Delta \vdash \psi \quad \Gamma, \Delta \vdash \phi \otimes \psi \quad \otimes R
\]

One can try to represent the fact that the rule is parametric in Γ and Δ by first organizing contexts into some category W. Assuming a reasonable definition of morphism (between contexts) in W, any formula χ then induces a presheaf \( \chi^+ : W^{op} \rightarrow Set \) by considering all the proofs of χ in a given context:

\[
\chi^+ = \Gamma \mapsto \{ \pi : \Gamma \vdash \chi \}
\]

Moreover, there is an external tensor product operation on presheaves, which to any pair of presheaves \( F : A^{op} \rightarrow Set \) and \( G : B^{op} \rightarrow Set \) associates a presheaf \( F \bullet G : (A \times B)^{op} \rightarrow Set \) defined by

\[
(F \bullet G)(a, b) = F(a) \times G(b).
\]

Then, one might hope to represent the fact that the \( \otimes R \) rule is parametric in Γ and Δ by interpreting it as a natural transformation from the external tensor product \( \phi^+ \bullet \psi^+ \) to the “internal” tensor product \( (\phi \otimes \psi)^+ \) except...
that this is not well-typed! Since $\phi^+ \otimes \psi^+$ is a presheaf over the product category $W \times W$, whereas $(\phi \otimes \psi)^+$ is a presheaf over $W$, literally interpreted, it does not make sense to speak of natural transformations between them.

What is missing is that implicitly the $\otimes R$ rule also has a side-effect, namely the “comma” operation on pairs of contexts (typically read as concatenation or multiset union). If we postulate that this operation corresponds to a functor

$$m : W \times W \to W$$

then the $\otimes R$ rule can indeed be interpreted as a natural transformation from $\phi^+ \otimes \psi^+$ to $(\phi \otimes \psi)^+$ precomposed with the functor $m$. But this suggests working with the refinement system $u : \text{Psh} \to \text{Cat}$, which organizes presheaves over arbitrary categories all together in one place. Our interpretation of the $\otimes R$ rule as a natural transformation “with side-effects” simply means that $\otimes R$ is interpreted as a derivation of the typing judgment

$$\phi^+ \otimes \psi^+ \Rightarrow_m (\phi \otimes \psi)^+$$

in the refinement system $u$.

As it turns out, the refinement system $u : \text{Psh} \to \text{Cat}$ (we will recall its formal definition in Section [III-A]) has the following quite special property: any refinement system $p : D \to T$ has a full and faithful embedding into $u$. We only noticed this “fundamental” embedding relatively recently (it can be seen as a simultaneous generalization of the Yoneda embedding of a category and of its representation by slice categories), but it seems to give a more principled way of understanding some of the phenomena which we had earlier observed about linear logic.

In particular, in these brief notes we wish to highlight that a duality generalizing Isbell duality naturally arises at the level of refinement types by considering the fundamental embedding of $p$ together with its “opposite embedding” (i.e., the fundamental embedding of $p^{op}$), and moreover that this duality can be expressed logically as negation into a “universal refinement” (a particular presheaf that can be defined for $p$). Combining this duality with the fact that the fundamental embedding preserves pullbacks, we also obtain two interesting formulas for the embedding of a pushforward, which decompose it either as the negation of a pullback of a negation, or as the double-negation of a pushforward. This is of course reminiscent of the classical negative translations from proof theory, but transported to a much more abstract context.

II. Preliminaries

A. Basic conventions and definitions

We recall some conventions from [5] for working with functors as type refinement systems. Given a fixed functor $p : D \to T$, we refer to the objects of $T$ as types, to the morphisms of $T$ as terms, and to the objects of $D$ as refinement types (or refinements for short). Since these notions are relative to a functor $p$, to avoid ambiguity one can speak of $p$-types, $p$-refinements, and so on. We say that a judgment is valid in a given refinement system $p$ if it has a derivation (in the sense of Defn. [3]), and more generally that a typing rule is valid when there is an operation transforming derivations of the premises into a derivation of the conclusion. For example, the rule

$$\frac{S \Rightarrow T \quad T \Rightarrow U}{S \Rightarrow U} \quad ;$$

is valid for any refinement system as an immediate consequence of functoriality: given a morphism $\alpha : S \to T$ such that $p(\alpha) = f$ and a morphism $\beta : T \to U$ such that $p(\beta) = g$, there is a morphism $(\alpha; \beta) : S \to U$ and moreover $p(\alpha; \beta) = (p(\alpha); p(\beta)) = (f; g)$.

We consider $p$-typing judgments modulo equality of terms (i.e., equality of morphisms in $T$), but often we mark applications of an equality by an explicit conversion step (which can be seen as admitting the possibility that $T$ is a higher category, although we will not pursue that idea rigorously here). For example, the “covariant subsumption” rule

$$\frac{S \Rightarrow T \quad T \Rightarrow U}{S \Rightarrow U} \quad ;$$

can be derived from the composition typing rule $(;)$ by

$$\frac{S \Rightarrow T \quad T \Rightarrow U}{\tilde{S} \Rightarrow U} \quad ;$$

where at $\sim$ we have applied the axiom $f = (f; \text{id})$ which is valid in any category (and hence in $T$).

The definition of a fibration of categories (in the sense of Grothendieck) is naturally expressed in the language of refinement systems by first defining a pullback of $T$ along $f$ as a refinement $f^* T$

$$f : A \to B \quad T \sqsubset A$$

equipped with a pair of typing rules

$$\frac{f^* T \Rightarrow T \quad \text{Lf}^* \quad S \Rightarrow T}{S \Rightarrow f^* T \quad \text{Rf}^*}$$

satisfying equations

$$\frac{\beta}{S \Rightarrow f^* T} \quad \text{Lf}^* \quad f^* T \Rightarrow T \quad \text{Rf}^* \quad S \Rightarrow T \quad \beta}$$

;
Dually, a pushforward of $S$ along $f$ is defined as a refinement $fS$

$$\frac{S \sqsupseteq A}{fS \sqsupseteq B} \quad \frac{S \sqsupseteq fA}{fS \sqsupseteq fB}$$

equipped with a pair of typing rules

$$\frac{S \Rightarrow T}{fS \Rightarrow gT} Lf \quad \frac{S \Rightarrow fS}{fS \Rightarrow fS} Rf$$

satisfying a similar pair of equations. Note that pullbacks and pushforwards are always determined up to vertical isomorphism, in the sense that two refinements $S,T \sqsupseteq A$ of a common type are vertically isomorphic (written $S \approx T$) when there exist a pair of subtyping derivations

$$\alpha \quad \beta \quad S \Rightarrow T \quad T \Rightarrow S$$

which compose to the identities on $S$ and $T$. We record the following facts, which are standard:

**Proposition 5.** Whenever the corresponding pullbacks and/or pushforwards exist:

1) the following subtyping rules are valid:

$$\frac{T_1 \leq T_2}{f^*T_1 \leq f^*T_2} \quad \frac{S_1 \leq S_2}{fS_1 \leq fS_2}$$

2) we have vertical isomorphisms

$$(g; f')T \Leftrightarrow g'f'T \quad \text{id'}T \Leftrightarrow T$$

$$(f; g)S \Rightarrow g fS \quad \text{id}S \Rightarrow S$$

A functor $p : \mathcal{D} \rightarrow \mathcal{T}$ is a fibration (resp. opfibration) if and only if a pullback $f^*T$ (resp. pushforward $fS$) exists for all compatible $f$ and $T$ (resp. $f$ and $S$). As the definitions make plain, though, it is possible to speak of specific pullbacks and pushforwards, even if $p$ is not necessarily a fibration and/or opfibration.

When $\mathcal{D}$ and $\mathcal{T}$ are monoidal categories, we call $p : \mathcal{D} \rightarrow \mathcal{T}$ a monoidal refinement system if it strictly preserves the tensor product and unit, and a logical refinement system if it also strictly preserves any residuals which may exist in $\mathcal{D}$. It follows that a logical refinement system admits all of the following refinement rules (we overload the symbols for the connectives of $\mathcal{D}$ and $\mathcal{T}$)

$$\frac{S_1 \sqcap A_1 \quad S_2 \sqcap A_2 \quad T \in I}{S_1 \cdot S_2 \sqcap A_1 \cdot A_2 \quad T \in I} \quad \frac{S \sqcap A \sqsubseteq U \sqsubseteq C \quad T \sqsubseteq U \sqsubseteq C \quad T \sqsubseteq B}{S \sqcap U \sqsubseteq A \cdot C \quad U \sqsubseteq C \cdot B \quad U \sqcap T \sqsubseteq C \cdot B}$$

as well as all of the following typing rules:

$$\frac{S_1 \Rightarrow T_1 \quad S_2 \Rightarrow T_2}{S_1 \cdot S_2 \Rightarrow T_1 \cdot T_2} \quad \frac{T \Rightarrow I}{I \Rightarrow I}$$

Moreover, derivations built using these typing rules satisfy a few equations, which we elide here. Note that the definition of a logical refinement system does not require $\mathcal{D}$ and $\mathcal{T}$ to be closed (i.e., that all residuals exist), although this will be the case in some examples and in particular for the refinement system $\mathcal{u} : \text{Psh} \rightarrow \text{Cat}$. When $\mathcal{D}$ and $\mathcal{T}$ are monoidal and $p : \mathcal{D} \rightarrow \mathcal{T}$ preserves tensor products and residuals, we call $p$ a closed monoidal refinement system.

**B. Morphisms of refinement systems**

Given a pair of refinement systems $p : \mathcal{D} \rightarrow \mathcal{T}$ and $q : \mathcal{E} \rightarrow \mathcal{B}$, by a morphism of refinement systems from $p$ to $q$ we mean a pair $F = (F_D, F_T)$ of functors $F_D : \mathcal{D} \rightarrow \mathcal{E}$ and $F_T : \mathcal{T} \rightarrow \mathcal{B}$ such that the square

$$\begin{array}{ccc}
\mathcal{D} & \xrightarrow{F_D} & \mathcal{E} \\
p \downarrow & & \downarrow q \\
\mathcal{T} & \xrightarrow{F_T} & \mathcal{B}
\end{array}$$

commutes strictly. Omitting subscripts on the functors $F$, a morphism from $p$ to $q$ thus induces a typing rule

$$\frac{S \Rightarrow T}{F[S] \Rightarrow F[T]} f$$

transporting derivations of $p$-judgments to derivations of $q$-judgments.

Given a pair of morphisms of refinement systems

$$(F_D, F_T) : p \rightarrow q \quad \text{and} \quad (G_D, G_T) : q \rightarrow p$$

an adjunction of refinement systems $(F_D, F_T) \dashv (G_D, G_T)$ consists of a pair of adjunctions of categories

$$(F_D \dashv G_D) \quad \text{and} \quad F_T \dashv G_T$$

such that the unit and counit of $F_D \dashv G_D$ are mapped by $p$ and $q$ onto the unit and counit of $F_T \dashv G_T$.

Writing $\iota$ and $\phi$ (without subscripts) for the unit and counit of both adjunctions $F_D \dashv G_D$ and $F_T \dashv G_T$, an adjunction of refinement systems thus induces a pair of typing rules

$$\frac{S \Rightarrow GF[S]}{S \Rightarrow T \Rightarrow \phi} \quad \frac{F\Gamma[T]}{T \Rightarrow \iota}$$
and we remark moreover that typing derivations constructed using these rules are subject to various equations implied by the definition of an adjunction of categories, such as the triangle laws.

C. Right adjoints preserve pullbacks

In this section we prove a basic result about adjunctions of refinement systems, analogous to the well-known fact that in an adjunction of categories the right adjoint functor preserves limits. We do not know whether this property has already been observed in the literature, but our interest lies in Corollary 8 which will be applied later to the refinement system of presheaves.

**Proposition 6.** If \( G : q \to p \) is a right adjoint, then \( G \) sends \( q \)-pullbacks to \( p \)-pullbacks, i.e., we have that
\[
G[f^* T] \leftrightarrow G[f] \ GV[T]
\]
for all \( f : A \to B \) and \( T \subset B \).

**Proof.** We need to show that \( G[f^* T] \) is a pullback of \( G[T] \) along \( G[f] \). By definition, this means constructing a pair of typing rules
\[
\frac{S \Rightarrow G[T]}{G[f^* T] \Rightarrow (G[f]/f)G[T]} \quad \frac{S \Rightarrow G[f]}{S \Rightarrow (G[f]/f)G[T]} \quad \frac{S \Rightarrow G[T]}{RG[f] \Rightarrow S \Rightarrow (G[f]/f)G[T]}
\]
satisfying the \( \beta \) and \( \eta \) equations. The left-rule is derived immediately from the left-rule for \( f^* \) by applying \( G_J \):
\[
f^* T \Rightarrow f \Rightarrow Lf^* \Rightarrow G[f^* T] \Rightarrow G[T]
\]
The right-rule can be derived in a few more steps from the right-rule for \( f^* \), assuming the existence of an \( F \) such that \( F \dashv G \):
\[
\frac{S \Rightarrow G[T]}{F[S] \Rightarrow FG[T]} F \Rightarrow F G[T] \Rightarrow T o \quad \frac{F[S] \Rightarrow T}{F[S] \Rightarrow T o \Rightarrow Rf^*} \beta_1 \quad \frac{F[S] \Rightarrow f T \Rightarrow Rf^*}{S \Rightarrow GF[S] \Rightarrow G[f^* T] \Rightarrow G[T]} \quad \frac{S \Rightarrow G[T]}{S \Rightarrow G[f^* T] \Rightarrow G[T]} \quad \frac{S \Rightarrow GF[S]}{S \Rightarrow G[f^* T] \Rightarrow G[T]}
\]
Here at \( \beta_1 \) and \( \beta_2 \) we invoke, respectively, naturality of the counit and a triangle law for \( F_J \dashv G_J \).

Finally, the fact that these typing rules satisfy the \( \beta \) and \( \eta \) equations can be verified by a long but mechanical calculation (see the Appendix). □

**Proposition 7.** If \( F : p \to q \) is a left adjoint, then \( F \) sends \( p \)-pushforwards to \( q \)-pushforwards, i.e., we have that
\[
F[f] F[S] \leftrightarrow F[S f]
\]
for all \( f : A \to B \) and \( S \subset A \).

**Proof.** By duality.

**Corollary 8.** If \( p : D \to T \) is a closed monoidal refinement system, then whenever the corresponding pullbacks and pushforwards exist we have vertical isomorphisms
\[
\begin{align*}
(f \circ g)(S \cdot T) & \leftrightarrow f S \cdot g T \quad (1) \\
(f S \cdot g^* U) & \leftrightarrow (f / g)^* (S \cdot U) \quad (2) \\
g^* U / f T & \leftrightarrow (g / f)^* (U / T) \quad (3)
\end{align*}
\]

**Proof.** Subtyping judgments in the left-to-right direction can be derived in any logical refinement system, e.g., (1):
\[
\begin{align*}
S \Rightarrow f S R f & \Rightarrow T \Rightarrow g T \\
S \Rightarrow f S \Rightarrow f S \Rightarrow g T \Rightarrow T \Rightarrow g T \\
(f \circ g)(S \cdot T) & \Rightarrow f S \cdot g T \Rightarrow L(f \circ g)
\end{align*}
\]
The additional assumption of monoidal closure implies that these are vertical isomorphisms, using the fact that any closed monoidal category \( C \) induces a pair of adjunctions
\[
\begin{array}{cccc}
C & \perp & C & \perp & C \\
\downarrow & & \downarrow & & \downarrow \\
\perp & & \perp & & \perp
\end{array}
\]
as well as a pair of adjunctions
\[
\begin{array}{cccc}
C & \perp & C & \perp & C \\
\downarrow & & \downarrow & & \downarrow \\
\perp & & \perp & & \perp
\end{array}
\]

Explicitly, we have (1) by
\[
(f \circ g)(S \cdot T) \leftrightarrow (id \circ g)(f \circ id)(S \cdot T) \quad (\text{Prop. 4})
\]
\[
\not\leftrightarrow (id \circ g)(f S \cdot T) \quad (- \cdot T + \cdot f T)
\]
and (2) (and similarly (3)) by
\[
\begin{align*}
f S \cdot g^* U & \leftrightarrow (id \circ g)^* (f S \cdot U) \quad (f S \cdot f S - f S -) \\
& \leftrightarrow (id \circ g)^* (f S \cdot U) \quad (f S \cdot f S - f S -) \\
& \leftrightarrow (f \cdot g)^* (S \cdot U) \quad (\text{Prop. 4})
\end{align*}
\]
where in the second-to-last step we use the fact that a \( p \text{-}p \)-pullback is the same thing as a \( p \text{-}p \)-pushforward. □
III. Embedding Refinement Systems into the Refinement System of Presheaves

A. The Refinement System of Presheaves

The refinement system of presheaves \( \mathbf{u} : \mathbf{Psh} \rightarrow \mathbf{Cat} \) is defined as follows:

- \( \mathbf{Cat} \) is the category whose objects are categories and whose morphisms are functors.
- \( \mathbf{Psh} \) has objects corresponding to pairs \((A, S)\), where \(A\) is a category and \(S : A^{\text{op}} \rightarrow \mathbf{Set}\) is a contravariant presheaf on \(A\), and morphisms \((A, S) \rightarrow (B, T)\) corresponding to pairs of a functor \(f : A \rightarrow B\) together with a natural transformations \(\alpha : S \Rightarrow (f^*; T)\).
- \( \mathbf{u} : \mathbf{Psh} \rightarrow \mathbf{Cat} \) is the evident projection functor.

To put it more succinctly, \( \mathbf{u} : \mathbf{Psh} \rightarrow \mathbf{Cat} \) is defined by the property that a refinement \(S \sqsubseteq A\) corresponds to a presheaf over the category \(A\), while a derivation of a typing judgment

\[
S \implies_T f
\]

corresponds to a natural transformation from \(S\) to \(T\) along the functor \(f\).

**Proposition 9.** \( \mathbf{u} \) is a closed monoidal refinement system with all pullbacks and pushforwards.

**Proof.** Pullbacks are defined by precomposition, while pushforwards are defined as coends:

\[
f^* T : A^{\text{op}} \rightarrow \mathbf{Set} \\
f^* T \overset{\equiv}{=} a \mapsto T(fa) \\
f : B^{\text{op}} \rightarrow \mathbf{Set} \\
f S \overset{\equiv}{=} b \mapsto \int_{a}^{	ext{Set}} B(b, fa) \times_T T(a)
\]

Tensor products and residuals in \( \mathbf{Cat} \) are defined using its usual cartesian closed structure (i.e., by building product categories and functor categories), and lifted to \( \mathbf{Psh} \) as follows (we show only the definitions of presheaves, not the structural maps):

\[
S \cdot T : (A \times B)^{\text{op}} \rightarrow \mathbf{Set} \\
S \cdot T \overset{\equiv}{=} (a, b) \mapsto S(a) \times_T T(b) \\
I \sqsubseteq 1 \rightarrow \mathbf{Set} \\
I = \ast \mapsto \{\ast\} \\
S \setminus U : [A, C]^{\text{op}} \rightarrow \mathbf{Set} \\
S \setminus U \overset{\equiv}{=} f \mapsto \int_{a}^{	ext{Set}} S(a) \rightarrow_T U(fa)
\]

Note that since left and right residuals in \( \mathbf{u} \) only differ in the definition of evaluation and currying, here and often in the sequel we show only the case of left residuals, leaving the other case for the reader. We also remark that these definitions may be naturally generalized to the refinement system of \( \mathcal{V} \)-valued presheaves over \( \mathcal{V} \)-enriched categories (see \cite[Example 5]{5}), but for concreteness we only work with ordinary categories in this paper.

The refinement system of categories with a chosen object \( \mathbf{t} : \mathbf{Cat} \rightarrow \mathbf{Cat} \) is defined so that refinements \( a \sqsubseteq A \) correspond to objects \( a \in A \), and derivations

\[
\alpha : a \Rightarrow b
\]

(where \(a \in A, b \in B,\) and \(f\) is a functor from \(A\) to \(B\)) correspond to morphisms \(\alpha : f(a) \rightarrow b\) in \(B\).

**Remark 10.** The representable presheaves can be seen as the image of \(\mathbf{t}\) along a morphism of refinement systems

\[
\begin{array}{ccc}
\mathbf{Cat} \rightarrow & \mathbf{Psh} \\
\mathbf{Cat} \downarrow & \downarrow \mathbf{u} \\
\end{array}
\]

where \(y\) is the functor sending \(a \in A\) to \(A(-, a) : A^{\text{op}} \rightarrow \mathbf{Set}\). Moreover, viewing each object \(a \in A\) as a functor \(a : 1 \rightarrow A\) in \(\mathbf{Cat}\), the presheaf \(A(-, a)\) can be described as a pushforward along a of the unit presheaf, i.e., \(A(-, a) \Rightarrow a I\).

B. The fundamental embedding

In this section we show that any refinement system \(\mathbf{p} : \mathcal{D} \rightarrow \mathcal{T}\) may be faithfully embedded in the refinement system of presheaves. We begin by describing this embedding directly, and then provide some analysis.

**Definition 11.** For any \(\mathbf{p}\)-type \(B\), the category \(B^\mathbf{p}\) is defined as follows:

- Objects are pairs \((S \sqsubseteq A, f : A \rightarrow B)\)
- Morphisms \((S_1, f_1) \rightarrow (S_2, f_2)\) are derivations

\[
S_1 \overset{\alpha}{\Rightarrow} f_1 \Rightarrow S_2
\]

where \(f_1 \overset{\_}{=} c \cdot f_2\).

**Proposition 12.** The assignment \(B \mapsto B^\mathbf{p}\) extends to a functor \((-)^\mathbf{p} : \mathcal{T} \rightarrow \mathbf{Cat}\).

**Definition 13.** For any \(\mathbf{p}\)-refinement \(T \sqsubseteq B\), there is a presheaf \(T^\mathbf{p}\) over \(B^\mathbf{p}\) defined on objects by

\[
(S, f) \rightarrow \{ a \mid S \overset{\alpha}{\Rightarrow} f \}
\]

and with the contravariant functorial action transforming any morphism \((S_1, f_1) \rightarrow (S_2, f_2)\) given as a derivation

\[
S_1 \overset{\alpha}{\Rightarrow} f_1 \Rightarrow S_2
\]

with \(f_1 \overset{\_}{=} c \cdot f_2\) into a typing rule (parametric in \(T\))

\[
S_2 \overset{f_2}{\Rightarrow} T \\
S_1 \overset{f_1}{\Rightarrow} T
\]
derived as
\[
\alpha \\
S_1 \implies S_2 \quad S_2 \implies T \\
\frac{c_{f_2}}{S_1 \implies T} \quad \frac{f_2}{S_1 \implies T} \sim \\
\frac{c_{f_1}}{S_1 \implies T}
\]

Proposition 14. The assignment \((T \sqsubseteq B) \mapsto (B^p, T^p)\) extends to a functor \((-)^p: \mathcal{D} \to \mathbf{Psh}\).

Proposition 15. The pair of functors \((-)^p: \mathcal{T} \to \mathbf{Cat}\) and \((-)^p: \mathcal{D} \to \mathbf{Psh}\) define a morphism of refinement systems from \(p\) to \(u\), i.e., a commuting square
\[
\begin{array}{ccc}
\mathcal{D} & \xrightarrow{(-)^p} & \mathbf{Psh} \\
\downarrow p & & \downarrow u \\
\mathcal{T} & \xrightarrow{(-)^p} & \mathbf{Cat}
\end{array}
\]

As we discussed in Section II-B, any morphism of refinement systems induces a corresponding typing rule, and in this case in particular we have a rule
\[
S \implies T \quad \frac{\alpha}{S^p \implies T^p} \quad \frac{f^p}{+p}
\]

which we call the fundamental embedding of the refinement system \(p\) into the refinement system of presheaves. In some sense, this embedding is a simultaneous generalization of the slice construction and of the Yoneda embedding of a category.

Remark 16. The category \(B^p\) can be seen as an analogue of the slice category over \(B\), and reduces to the ordinary slice of \(\mathcal{T}\) over \(B\) in the case where \(p\) is the identity functor \(p = \text{id}_T: \mathcal{T} \to \mathcal{T}\). As such, we will sometimes refer to \(B^p\) as the \(p\)-slice of \(B\). Another way to explain this is that the functor \((-)^p: \mathcal{T} \to \mathbf{Cat}\) can be factored as the nerve of \(p\) followed by the category of elements construction:
\[
\mathcal{T} \xrightarrow{B \to \mathcal{T}(p \cdot B)} [\mathcal{D}^p, \mathbf{Set}] \xrightarrow{f} \mathbf{Cat}
\]

Remark 17. The presheaf \(T^p\) can be seen as a generalization of the Yoneda embedding of \(T\), and reduces to the ordinary Yoneda embedding in the special case where \(p\) is the functor \(p = 1_D: \mathcal{D} \to 1\) into the terminal category. (Observe that in that case the \(p\)-slice over the single object \(*\) of \(1\) is \(\mathcal{D}\) itself.)

The ordinary Yoneda embedding of a category is full and faithful, so it is perhaps unsurprising that the same is true for the fundamental embedding of a refinement system. More precisely, by this we mean that the typing rule \(+p\) is invertible, in the sense that to any \(u\)-derivation
\[
\frac{\alpha}{S^p \implies T^p} \quad \frac{f^p}{S \implies T}
\]

there is a unique \(p\)-derivation
\[
\frac{\alpha}{(S^p)^\alpha} \quad \frac{f^p}{S \implies T}
\]

such that
\[
\frac{\alpha}{S^p \implies T^p} \quad \frac{f^p}{(S^p)^\alpha \implies T^p + p}
\]

This property is an easy consequence of the following fact, which is immediate.

Proposition 18. The presheaf \(S^p \subseteq A^p\) is represented by the object \((S, \text{id}_A) \in A^p\), i.e., \(S^p \Leftrightarrow (S, \text{id})I\).

By Remark 17 this last observation can also be rephrased as saying that the fundamental embedding \(p \to u\) factors via the refinement system of categories with a chosen object:
\[
\begin{array}{ccc}
\mathcal{D} & \xrightarrow{(-)^p} & \mathbf{Cat} \\
\downarrow p & & \downarrow u \\
\mathcal{T} & \xrightarrow{(-)^p} & \mathbf{Cat}
\end{array}
\]

C. The opposite embedding

Every functor \(p: \mathcal{D} \to \mathcal{T}\) induces an opposite functor \(p^{op}: \mathcal{D}^{op} \to \mathcal{T}^{op}\), and one can consider the fundamental embedding of \(p^{op}\)
\[
\begin{array}{ccc}
\mathcal{D}^{op} & \xrightarrow{(-)^{op}p} & \mathbf{Psh} \\
\downarrow p^{op} & & \downarrow u \\
\mathcal{T}^{op} & \xrightarrow{(-)^{op}p} & \mathbf{Cat}
\end{array}
\]

as another “opposite embedding” of \(p\). Letting \((-)^{op} \equiv (-)^p\), \(p^{op} \to u\), this means we have an invertible rule
\[
\frac{\alpha}{S \implies T} \quad \frac{f}{T \implies S^p} \quad \frac{-p}{+p}
\]

embedding \(p\) contravariantly into \(u\).

Unravelling the definitions, we can verify that

- \(A^{-p}\) is the opposite of the category whose objects consist of pairs
\[
(g: A \to B, T \sqsubseteq B)
\]

and whose morphisms \((g_1, T_1) \to (g_2, T_2)\) correspond to derivations
\[
T_1 \xrightarrow{\alpha} T_2
\]

with \(g_1 \circ h = g_2\). Dually to \(A^p\), we can read \((A^{-p})^{op}\) as the “\(p\)-coslice” category out of \(A\).
• For any \( p \)-refinement \( S \subseteq A \), \( S^p \) is the presheaf defined on objects by
\[
(g, T) \mapsto \{ \alpha \mid S \xrightarrow{\alpha} T \}
\]
and on morphisms by
\[
\frac{S \xrightarrow{\alpha} T_1 \xrightarrow{h} T_2}{S \xrightarrow{\alpha} T_2} \sim \frac{S \xrightarrow{g} T_1}{S \xrightarrow{g} T_2}.
\]

Note that \( S^p \) is a contravariant presheaf over \( A^p \), and thus a covariant presheaf over the \( p \)-coslice category.

**Proposition 19.** The presheaf \( S^p \subseteq A^p \) is represented by the object \((\text{id}_A, S)\) in \( A^p \), i.e., \( S^p \equiv (\text{id}, S)\).

**D. Preservation of pullbacks**

An important property of the fundamental embedding is that it preserves pullbacks.

**Proposition 20.** Whenever \( f \circ T \) exists in \( p \), we have
\[
(f \circ T)^p \equiv f^p \circ T^p
\]

**Proof.** By expanding definitions, the elements of \((f \circ T)^p\) correspond to \( p \)-derivations
\[
S \xrightarrow{\alpha} f^p T
\]
(for arbitrary \( S \subseteq X \) and \( g : X \to A \)), while the elements of \( f^p \circ T^p \) correspond to \( p \)-derivations
\[
S \xrightarrow{\beta} T
\]
\[
\text{So, the proposition follows from the universal property of the \( p \)-pullback.}
\]

As an immediate corollary, we have that the opposite embedding sends \((p\)-)pushforwards to \((u\)-)pushbacks.

**Proposition 21.** Whenever \( f S \) exists in \( p \), we have
\[
(f S)^p \equiv f^p \circ S^p
\]

On the other hand, the fundamental embedding need not preserve pushforwards: although it’s true that the subtyping judgment
\[
f^p S^p \Rightarrow (f S)^p
\]
is valid whenever the pushforward \( f S \) exists in \( p \) (indeed, this is true whenever one has a morphism of refinement systems and the pushforward exists on both sides), in general the converse subtyping judgment need not be valid. (Thankfully, in Section IV-C we will show that though the fundamental embedding need not preserve pushforwards, it at least preserves them “up to double-negation”.)

**E. Embedding tensors and residuals**

Suppose that \( p : \mathcal{D} \to \mathcal{T} \) is a monoidal refinement system. By definition, this means that \( \mathcal{D} \) and \( \mathcal{T} \) are monoidal and that we have a commuting square
\[
\mathcal{D} \times \mathcal{D} \xrightarrow{p^p} \mathcal{T} \times \mathcal{T}
\]
(as well as a commuting triangle associated to the tensor unit, but we will ignore the unit in this section, since its treatment is completely analogous). Since \( u : \text{Psh} \to \text{Cat} \) is also a monoidal refinement system, the fundamental embedding of \( p \) thus induces a cube
\[
\text{Psh} \times \text{Psh} \longrightarrow \text{Cat} \times \text{Cat}
\]
and moreover the natural transformation on the right is the projection of the one on the left along the cube, this meaning that we have a family of functors
\[
m_{B_1,B_2} : (B_1^p \times B_2^p) \to (B_1 \bullet B_2)^p
\]
and a family of \( u \)-derivations
\[
T_1^p \bullet T_2^p \xrightarrow{m_{T_1,T_2}} (T_1 \bullet T_2)^p
\]
natural in \( T_1 \subseteq B_1 \) and \( T_2 \subseteq B_2 \). Explicitly, the functors \( m_{B_1,B_2} \) are defined by the action sending any pair of objects
\[
(S_1, f_1) \quad (S_2, f_2)
\]
(where \( S_1 \subseteq A_1, f_1 : A_1 \to B_1, S_2 \subseteq A_2, f_2 : A_2 \to B_2 \) to the object
\[
(S_1 \bullet S_2, f_1 \bullet f_2)
\]
while the natural transformations \( m_{T_1,T_2} \) are defined by the action sending any pair of \( p \)-derivations
\[
\alpha_1 \xrightarrow{f_1} T_1 \quad \alpha_2 \xrightarrow{f_2} T_2
\]
to the \( p \)-derivation

\[
\begin{array}{c}
S_1 \rightarrow f \quad T_1 \\
S_2 \rightarrow f_j \quad T_2
\end{array}
\]

\[
\frac{S_1 \cdot S_2 \rightarrow f_j \cdot f_j}{\rightarrow T_1 \cdot T_2}
\]

We can summarize all this by saying that the fundamental embedding is a \textit{lax monoidal} morphism of refinement systems.

From this it follows that we can likewise build functors

\[ a_{A,C} : (A \setminus C)^+ \rightarrow A^+ \setminus C^+ \]

and \( u \)-derivations

\[ a_{S \setminus U} \]

\[ (S \setminus U)^+ \rightarrow (S \setminus U)^+ \]

at least so long as the corresponding residuals exist in \( T \) and \( p \). The derivation proceeds as follows:

\[
\begin{array}{c}
S \cdot S \setminus U \rightarrow U
\end{array}
\]

\[
\frac{S^+ \cdot (S \setminus U)^+ m \rightarrow (S \setminus U)^+}{\rightarrow (S \setminus U)^+ \lambda}
\]

Somewhat more interestingly, \((S \setminus U)^+\) is actually the \textit{pullback} of \( S^+ \cdot U^+ \) along \( a = \lambda[m; \Theta]^+ \).

**Proposition 22.** For all \( S \subseteq A \) and \( U \subseteq C \), if the residual \( S \setminus U \subseteq A \setminus C \) exists then

\[
(S \setminus U)^+ \Leftrightarrow a_{A,C}^-(S^+ \setminus U^+)
\]

**Proof.** We begin with some calculation:

\[
a^+((S^+ \setminus U^+)) \Leftrightarrow a^+ ((S, \text{id}) I \setminus U^+) \quad \text{(Prop. 18)}
\]

\[
eq a^+ ((S, \text{id}) \setminus \text{id})(I \setminus U^+) \quad \text{(Corollary 8)}
\]

\[
eq (a; (S, \text{id}) \setminus \text{id})(U^+) \quad \text{(residuation by I)}
\]

Through a bit more calculation, we get that the elements of the presheaf on the last line correspond to derivations

\[
S \cdot T \rightarrow U
\]

(\text{for arbitrary } T \subseteq B \text{ and } g : B \rightarrow A \setminus C), \text{ and the universal property of the residual in } p \text{ says that these derivations are in one-to-one correspondence with the elements of } (S \setminus U)^+ p.}

\[ \square \]

IV. Duality and Negative Encodings

In any closed monoidal refinement system \( q : E \rightarrow B \) with pullbacks, given an arbitrary binary operation

\[ \odot : X \cdot Y \rightarrow Z \]

together with an arbitrary refinement \( U \subseteq Z \), one can define a pair of operations

\[
\begin{array}{c}
U \rightarrow \lambda \Theta \rightarrow (\phi \setminus U) \\
U \rightarrow \odot \psi \rightarrow (U / \psi)
\end{array}
\]

witnessing a contravariant adjunction

\[
\begin{array}{c}
E_X \leftrightarrow E^\text{op}_Y
\end{array}
\]

between the refinements of \( X \) and the refinements of \( Y \). Equipped with such an adjunction, one moreover might hope to find pairs of refinements \( \phi \subseteq X \) and \( \psi \subseteq Y \) which are \textit{dual} in the sense that both

\[
\phi \Leftrightarrow U \odot \psi \text{ and } \psi \Leftrightarrow U \odot \psi.
\]

Although this might seem like a quite special situation, we will show here that \textit{any} refinement \( S \subseteq A \) in an arbitrary refinement system \( p \) gives rise to a dual pair in the refinement system of presheaves, namely the pair \( S^+ \subseteq A^+ \) and \( S^- \subseteq A^- \).

A. The category of judgments and the universal refinement

Again, we assume given an arbitrary refinement system \( p : D \rightarrow T \).

**Definition 23.** The category of judgments \( I_p \) is defined as follows:

- objects are \( p \)-typing judgments, i.e., triples \((S, f, T)\) where \( S \subseteq A, f : A \rightarrow B, \) and \( T \subseteq B \).
- morphisms \((S_1, f_1, T_1) \rightarrow (S_2, f_2, T_2)\) are pairs of derivations

\[
\begin{array}{c}
S_1 \rightarrow f \quad S_2 \rightarrow f_2 \\
\rightarrow T_1 \rightarrow T_2
\end{array}
\]

such that \( f_1 = e ; f_2 ; h \).

**Definition 24.** The universal refinement of \( p \) is the presheaf \( U_p \subseteq I_p \) defined on objects by

\[
U_p(S, f, T) = \{ a \mid S \rightarrow f \}
\]

and with the functorial action transforming any morphism \((S_1, f_1, T_1) \rightarrow (S_2, f_2, T_2)\) given as a pair of derivations

\[
\begin{array}{c}
S_1 \rightarrow f \quad S_2 \rightarrow f_2 \\
\rightarrow T_1 \rightarrow T_2
\end{array}
\]

such that \( f_1 = e ; f_2 ; h \) into a typing rule

\[
\begin{array}{c}
S_2 \rightarrow f_2 \\
S_1 \rightarrow f_1
\end{array}
\]
derived as
\[
\begin{align*}
\begin{array}{c}
\beta \\
\sigma \\
\beta \\
\end{array}
\begin{array}{c}
S_1 \xrightarrow{c} S_2 \\
S_2 \xrightarrow{f_h} T_2 \\
\gamma \\

\end{array}
\begin{array}{c}
\overset{h}{\xrightarrow{}} T_1 \\
\overset{h}{\xrightarrow{}} T_1 \\
\overset{h}{\xrightarrow{}} T_1 \\
\end{array}
\end{align*}
\]

Remark 25. The category of judgments \( J_p \) can be seen as an analogue of the “twisted arrow category” of \( \mathcal{T} \) (cf. \[2\], p.11, [4], §1.1.18). reducing to \( \text{Twin}(\mathcal{T}) \) in the case \( p = \text{id}_\mathcal{T} \).

Remark 26. In the case where \( p = !_D : \mathcal{D} \to 1 \), the universal refinement of \( p \) reduces to the hom bimodule \( U_p = \mathcal{D}(-,-) \) (noting that in that case \( I_p = \mathcal{D} \times \mathcal{D}^{op} \)).

B. Conjugation and duality

At each type \( A \), we can define a functor
\[ @_A : A^p \times A^{op} \to I_p \]
which pairs an object \( (S, f) \) of the \( p \)-slice over \( A \) together with an object \( (g, T) \) of the \( p \)-coslice over \( A \), to form the \( p \)-typing judgment \( (S, (f \circ g), T) \). Moreover, the family of such functors describes an extranatural transformation, in the sense that

**Proposition 27.** For any \( f : A \to B \) we have
\[(f^p \circ \text{id}_B) ; @_B = (\text{id}_A \bullet f^p) ; @_A \]

Now to explain the reason why the universal refinement of \( p \) is considered “universal”, we observe that the embedding (or opposite embedding) of any \( p \)-refinement is a pullback of \( U_p \), along a map defined using the operations \( @_A \).

**Proposition 28.** For a given \( S \subset A \), define
\[ k_S : A^p \to I_p \overset{\text{def}}{=} \text{id} \circ (\text{id}, S); @_A \]
\[ v_S : A^{op} \to I_p \overset{\text{def}}{=} (\text{id}, \text{id}); @_A \]
Then \( S^p \Leftrightarrow k_S \ast U_p \) and \( S^{op} \Leftrightarrow v_S \ast U_p \).

**Proof.** Immediate from the definitions of \( S^p, S^{op} \), and \( U_p \). The encoding amounts to “instantiating” \( U_p \) on the right or on the left.

Now, using the fact that the embeddings of \( p \)-refinements are pullbacks of the universal refinement, together with other properties we have already established, we can derive a duality between \( S^p \) and \( S^{op} \) via negation into \( U_p \).

**Proposition 29.** For all \( S \subset A \), we have
\[ S^{op} \Leftrightarrow \frac{U_p}{S^p \oplus} \quad S^p \Leftrightarrow \frac{U_p}{@ S^{op}} \]

where
\[ \begin{align*}
U_p & \overset{\text{def}}{=} \lambda[@_A] \ast (\phi \setminus U_p) & (\phi \subset A^p) \\
U_p & \overset{\text{def}}{=} \rho[@_A] \ast (U_p / \psi) & (\psi \subset A^{op})
\end{align*} \]

**Proof.** We show one case (the other is symmetric):
\[ \frac{U_p}{S^p \oplus} \Leftrightarrow (\lambda[\text{id}]; ((S, \text{id}) \setminus \text{id})) \ast U_p \]
\[ \Leftrightarrow ((\text{id}, \text{id}) \circ \text{id}; @_A) \ast U_p \]
\[ \Leftrightarrow S^{op} \]

**Remark 30.** When \( p = !_D : \mathcal{D} \to 1 \), the operation of negation into \( U_p \) reduces to Isbell conjugation (cf. \[3\], §7) between the category \( [\mathcal{D}^{op}, \text{Set}] \) of contravariant presheaves and the category \( [\mathcal{D}, \text{Set}]^{op} \) of covariant presheaves, and Prop. 26 reduces to the fact that Isbell conjugation restricts to a duality on representable presheaves.

C. Negative encodings of the pushforward

The fact that negation into the universal refinement of \( p \) restricts to a duality on “representables” is (we believe) an important observation, and an indication of the link between Isbell duality and computational duality. Just as the classical double-negation translations from proof theory have a counterpart in continuation-passing transformations in programming languages, the operation of negation on presheaves opens up the possibility of defining different refinement type constructors through negative encodings. We would like to study such encodings more deeply in a future paper, but here we content ourselves to describing two such encodings for the fundamental embedding of a pushforward.

**Proposition 31.** Whenever the pushforward \( f S \) exists in \( p \), we have

1) \( (f S)^p \Leftrightarrow \frac{U_p}{@ (f S)^{op}} \)
2) \( (f S)^{op} \Leftrightarrow \frac{U_p}{@ (f S)^p} \)

**Proof.** We can derive equation (1) in two steps:
\[ (f S)^p \Leftrightarrow \frac{U_p}{@ (f S)^{op}} \quad \text{(Prop. 29)} \]
\[ \Leftrightarrow \frac{U_p}{@ S^{op}} \quad \text{(Prop. 21)} \]

For (2), we use the following lemma:
\[ \frac{U_p}{f^p S^p \oplus} \Leftrightarrow (\lambda[\text{id}]; (f^p \setminus \text{id})) \ast (S^p \setminus U_p) \]
\[ \Leftrightarrow ((f^p \bullet \text{id}; @_B) \ast (S^p \setminus U_p) \quad \text{(Prop. 27)} \]
\[ \Leftrightarrow (\text{id} \circ f^p ; @_A) \ast (S^p \setminus U_p) \]
\[ \Leftrightarrow f^p \ast \frac{U_p}{S^p \oplus} \]
\[ \Leftrightarrow f^p S^{op} \]
Equation (2) then follows by applying this lemma to (1).

\[ QED \]

References

[1] Jean-Yves Girard, Yves Lafont, and Paul Taylor. Proofs and Types. Cambridge University Press, 1990.

[2] F. William Lawvere. Equality in hyperdoctrines and comprehension schema as an adjoint functor, In Proceedings of the AMS Symposium on Pure Mathematics XVII, 1970, 1–14.

[3] F. William Lawvere. Taking Categories Seriously. Reprints in Theory and Applications of Categories 8, 2005, 1–24.

[4] Georges Maltsiniotis. La théorie de l’homotopie de Grothendieck. Astérisque, 2005.

[5] Paul-André Melliès and Noam Zeilberger. Functors are Type Refinement Systems. In Proceedings of the 42nd ACM SIGPLAN-SIGACT Symposium on Principles of Programming, Mumbai, 2015.
APPENDIX

We complete the proof of Prop. 6 by showing that the typing rules $LG[f]'$ and $RG[f]'$ defined by

\[
\frac{S \vDash G[T]}{ FG[S] \vDash FG[T] } \quad F \quad \frac{FG[T] \vDash T}{ G } \quad ^{\circ}
\]

\[
\frac{F[S] \vDash FG[T]}{ F } \quad \frac{F[S] \vDash T}{ G } \quad ^{\sim_1}
\]

\[
\frac{F[S] \vDash f^*T}{ G[f^*T] \vDash G[T] } \quad G
\]

satisfy the $\beta$ equation

\[
\frac{S \vDash G[T]}{ G[f^*T] \vDash G[T] } \quad \frac{RG[f]' \quad G[f^*T] \vDash G[T]}{ S \vDash G[T] } \quad ^{\beta} \quad \frac{S \vDash G[T]}{ G[f^*T] \vDash G[T] } \quad \frac{LG[f]' \quad G[f^*T] \vDash G[T]}{ S \vDash G[T] } \quad ^{\sim_2}
\]

as well as the $\eta$ equation

\[
\frac{S \vDash G[f^*T]}{ G[f^*T] \vDash G[T] } \quad \frac{LG[f]' \quad G[f^*T] \vDash G[T]}{ S \vDash G[T] } \quad ^{\eta}
\]

\[
\frac{S \vDash G[f^*T]}{ G[f^*T] \vDash G[T] } \quad \frac{RG[f]' \quad G[f^*T] \vDash G[T]}{ S \vDash G[T] }
\]

In both cases, we establish these equations by showing a series of typing derivations, with each successive derivation related to its predecessor by an elementary step of reasoning.

A. The $\beta$ equation

\[
\frac{S \vDash G[T]}{ FG[S] \vDash FG[T] } \quad F \quad \frac{FG[T] \vDash T}{ G } \quad ^{\circ}
\]

\[
\frac{F[S] \vDash FG[T]}{ F } \quad \frac{F[S] \vDash T}{ G } \quad ^{\sim_1}
\]

\[
\frac{F[S] \vDash f^*T}{ G[f^*T] \vDash G[T] } \quad G
\]

\[
\frac{S \vDash GF[S] }{ G[f^*T] \vDash G[T] } \quad \frac{RG[f]' \quad G[f^*T] \vDash G[T]}{ S \vDash G[T] } \quad ^{\beta}
\]

\[
\frac{S \vDash G[f^*T]}{ G[f^*T] \vDash G[T] } \quad \frac{LG[f]' \quad G[f^*T] \vDash G[T]}{ S \vDash G[T] } \quad ^{\sim_2}
\]
B. The $\eta$ equation

$$S \Rightarrow G[f^* T]$$
