1. From Presentation to Type Theory and Back

Shulman's practical type theory, hereafter PTT, is the natural synthesis of the previous two discussions.

1.1. Prop to type theory. Given a presentation

$$\mathfrak{F}\mathcal{R} \rightrightarrows \mathfrak{F}\mathcal{G}$$

of a symmetric monoidal category, we specify a type theory $\mathsf{PTT}_{\langle \mathcal{G} | \mathcal{R} \rangle}$ with:

- contexts Γ, Δ etc. being lists $x_1 : A_1, \ldots, x_n : A_n$, which we may write as $(x_1, \ldots, x_n) : (A_1, \ldots, A_n)$, of names for variables in generating objects A_i drawn from the set G_0; and
- three sorts of judgements:
  - term judgements, e.g

    $$\Gamma \vdash x \ \mathsf{term}$$

    or

    $$\Gamma \vdash f_{(k)}\,(M) \ \ \mathsf{term};$$
  - typing judgements, e.g.

    $$x : A \vdash f\,(x) : B$$

    corresponding to a morphism $f : A \longrightarrow B$ or

    $$x : A \vdash \big(h_{(1)}\,(x)\,, h_{(2)}\,(x)\big) : B$$

    corresponding to a morphism $h : A \longrightarrow (B_1, B_2)$ or

    $$\vdash (\,|\, z^a) : ()$$

    corresponding to a scalar constant, an endomorphism $z : () \longrightarrow ()$ of the unit object; and
  - equality (of typed terms) judgements, e.g.

    $$x : A \vdash f\,(x) = h\,(g\,(x)) : B$$

    corresponding to a commuting triangle

    $$\begin{tikzcd} A \arrow[r, "g"] \arrow[dr, "f"'] & C \arrow[d, "h"] \\ & B \end{tikzcd}$$

    or

    $$x : A \vdash \big(f\,\big(g_{(1)}\,(x)\big)\,, g_{(2)}\,(x)\big) = \big(h_{(1)}\,(x)\,, h_{(2)}\,(x)\big) : (B_1, B_2)$$

    corresponding to a commuting triangle

    $$\begin{tikzcd} A \arrow[r, "g"] \arrow[dr, "h"'] & (C, B_2) \arrow[d, "(f, \mathrm{id}_{B_2})"] \\ & (B_1, B_2) \end{tikzcd}$$

1.1.1. *Rules for the term judgement.* The term judgments of $\mathsf{PTT}_{\langle\mathcal{G}|\mathcal{R}\rangle}$ are, derived by way of rules combining Sweedler's notation and the generating morphisms of the signature $\mathcal{G}$, for example we've the rule

$$\frac{\Gamma \vdash M_1 \quad \ldots \quad \Gamma \vdash M_n \qquad f \in \mathcal{G}\left(A_1, \ldots, A_m; B_1, \ldots, B_n\right) \quad m \geq 1 \quad n \geq 2 \quad 1 \leq k \leq n}{\Gamma \vdash f_{(k)}\left(M_1, \ldots, M_n\right) \ \mathsf{term}}$$

which permits us to form the term $f_{(k)}\left(M_1, \ldots, M_n\right)$ which corresponds to the k^{th} Sweedler component of a morphism

$$f : \left(A_1, \ldots, A_m\right) \longrightarrow \left(B_1, \ldots, B_n\right)$$

in the signature $\mathcal{G}$. When the generator f has nullary domain, we introduce a subtly different term formation rule which add labels drawn from an alphabet $\mathfrak{A}$, syntactic sugar which Shulman uses to great effect later on.

$$\frac{f \in \mathcal{G}\left(; B_1, \ldots, B_n\right) \quad \mathfrak{a} \in \mathfrak{A} \quad n \geq 2 \quad 1 \leq k \leq n}{\Gamma \vdash f^{\mathfrak{a}}_{(k)} \ \mathsf{term}}$$

1.1.2. *Rules for the typing judgement.* The typing judgements of $\mathsf{PTT}_{\langle G|R\rangle}$ are derived from rules specified by the elements of G_1, the generating formal morphisms of the signature $\mathcal{G}$. For example, if in the signature $\mathcal{G}$ we find a formal morphism

$$f : \left(A_1, \ldots, A_m\right) \longrightarrow \left(B_1, \ldots, B_n\right)$$

then, we've a rule for the typing judgement which corresponds to applying that morphism to a list of typed terms $\left(M_1, \ldots, M_n\right) : \left(A_1, \ldots, A_n\right)$. The foreboding vaguery will be addressed shortly.

1.1.3. *Rules for the equality judgement.* Lastly, the rules for the equality judgement assert the generating identities as axioms and build a congruence from them.

1.2. **Type Theory to Prop.** Going the other way, to any practical type theory, $\mathsf{PTT}_{\langle G|R\rangle}$, we may associate its term model, $\mathbf{TM}\left(\mathsf{PTT}_{\langle\mathcal{G}|\mathcal{R}\rangle}\right)$, the category with

- $\mathsf{Ob}\left(\mathbf{TM}\left(\mathsf{PTT}_{\langle\mathcal{G}|\mathcal{R}\rangle}\right)\right)$ being the contexts of $\mathsf{PTT}_{\langle\mathcal{G}|\mathcal{R}\rangle}$; and
- $\mathsf{Mor}\left(\mathbf{TM}\left(\mathsf{PTT}_{\langle\mathcal{G}|\mathcal{R}\rangle}\right)\right)$ being the derivable typing judgments modulo derivable equality judgements.

1.3. **On the meaning of generation and the admissibility of structural rules.** Since, the derivable judgements of a type theory are, in a sense, generated by the rules of that type theory it is easy enough then to believe that, for any presentation

$$\mathfrak{F}\mathcal{R} \rightrightarrows \mathfrak{F}\mathcal{G}$$

of a PROP $\mathcal{C}$, the term model $\mathsf{TM}\left(\mathsf{PTT}_{\langle\mathcal{G}|\mathcal{R}\rangle}\right)$ is equivalent to $\mathcal{C}$.

Indeed, Shulman shows that:

- the term model of the type theory$\mathsf{PTT}_{\langle\mathcal{G}|\emptyset\rangle}$ enjoys the universal property of the free PROP on a signature (**Theorem 5.17**);
- the derivable equality judgements of $\mathsf{PTT}_{\langle\mathcal{G}|\mathcal{R}\rangle}$ comprise a congruence $\sim_R$ on $\mathfrak{F}\mathcal{G}$ (**Proposition 6.1**); and

- the PROP $\mathfrak{F}G_{\sim_R}$ is equivalent to the PROP $\mathscr{C}$ presented

$$\mathfrak{F}\mathcal{R} \rightrightarrows \mathfrak{F}\mathcal{G}$$

(**Theorem 6.2**)

Before one makes the jump from appreciating the naturality of the work to thinking it obvious, we must acknowledge something we've intentionally obscured: exactly how a signature G generates the rules of the typing judgement.

The 'obvious' way define the rules of the typing judgement such that we could expect a result like Theorem 5.17 would be to specify:

- a rule something like

$$\frac{\begin{array}{c}\Gamma \vdash (M_1, \ldots, M_n) : (A_1, \ldots, A_n) \\ f \in \mathcal{G}(A_1, \ldots, A_n; B_1, \ldots, B_m)\end{array}}{\Gamma \vdash \left(f_{(1)}(M_1, \ldots, M_n), \ldots, f_{(m)}(M_1, \ldots, M_n)\right) : (B_1, \ldots, B_m)}$$

  which would allow us to 'apply' a morphism

$$f \in \mathcal{G}(A_1, \ldots, A_n; B_1, \ldots, B_m)$$

  to a term $(M_1, \ldots, M_n)$ of $(A_1, \ldots, A_n)$;
- a rule something like

$$\frac{\Gamma \vdash (M_1, \ldots, M_n) : (A_1, \ldots, A_n) \quad \Delta \vdash (N_1, \ldots N_m) : (B_1, \ldots, B_m)}{\Gamma, \Delta \vdash (M_1, \ldots, M_n, N_1, \ldots, N_m) : (A_1, \ldots, A_n, B_1, \ldots, B_m)}$$

  which would allow us to tensor two morphisms together; and
- a rule something like

$$\frac{\Gamma \vdash (M_1, \ldots, M_n) : (A_1, \ldots, A_n) \quad \sigma \in S_n}{\Gamma \vdash \left(M_{\sigma(1)}, \ldots, M_{\sigma(n)}\right) : \left(A_{\sigma(1)}, \ldots, A_{\sigma(n)}\right)}$$

  corresponding to the exchange isomorphisms permuting the generating objects in a list thereof $(A_1, \ldots, A_n)$.

$$\frac{\Gamma \vdash M : A \quad f \in \mathcal{G}(A; B_1, B_2)}{B}$$

problem with this 'obvious' way however is that derivations for typing judgements would be non-unique. Using Shulman's example, see that the two different associations of three composable morphisms $f : A \longrightarrow B$, $g : B \longrightarrow C$, and $h : C \longrightarrow D$ would beget distinct derivations of the same typing judgement.

$$\frac{\dfrac{x : A \vdash f(x) : B \quad y : B \vdash g(y) : C}{x : A \vdash g(f(x)) : C} \qquad z : C \vdash h(z) : D}{x : A \vdash h(g(f(x)))}$$

$$\frac{x : A \vdash f(x) : A \qquad \dfrac{y : B \vdash g(y) : C \quad z : C \vdash h(z) : D}{y : A \vdash h(g(y)) : D}}{x : A \vdash h(g(f(x)))}$$

As induction over derivations is far more cumbersome than induction over derivable judgements Shulman opts for a more sophisticated tack; Shulman defines rules for the typing judgement such that:

- derivations of typing judgements are unique; and

- the structural rules, i.e. the rules corresponding to composition, tensorings, and exchange are admissible (see Section 5)

As such, not only are Theorem 5.17, Proposition 6.1, and Proposition 6.2 provable by way of a much tamer induction, but any user of the type theory may invoke these more naive 'structural' rules in derivations and then freely ignore the multiplicity of derivations such reasoning may bring about.

2. The Free Dual Pair

Recall that for a vector space V and its dual vector space V^*, we've a bijection

$$\mathsf{Hom}\,(A \otimes V, B) \xrightarrow{\sim} \mathsf{Hom}\,(A, V^* \otimes B)$$

natural in vector spaces A and B. This example is abstracted into the usual defintion of a dual pair in a symmetric strict monoidal category as follows.

Definition 1. A dual pair $(D, D^*, \mathsf{coev}, \mathsf{ev})$ in a symmetric monoidal category $(\mathcal{C}, (\_,\_), ())$ is comprised of:

- a pair of objects D and D^* of $\mathcal{C}$;
- a morphism $\mathsf{coev} : \mathbf{1} \longrightarrow D \otimes D^*$; and
- a morphism $\mathsf{ev} : D^* \otimes D \longrightarrow \mathbf{1}$

satisfying the triangle identities:

$$\begin{tikzcd} D \arrow[dr, "{(\mathsf{coev},D)}"] \arrow[dd, equal] & \\ & (D, D^*, D) \arrow[dl, "{(D,\mathsf{ev})}"] \\ D & \end{tikzcd} \qquad \begin{tikzcd} D^* \arrow[dr, "{(D^*,\mathsf{coev})}"] \arrow[dd, equal] & \\ & (D^*, D, D^*) \arrow[dl, "{(\mathsf{ev},D^*)}"] \\ D^* & \end{tikzcd}$$

What's more, these data clearly suggest a presentation of the PROP generated by a dual pair. We set

$$\mathcal{G} = (\{D, D^*\}, \{\mathsf{coev} : () \longrightarrow (D, D^*), \mathsf{ev} : (D^\star, D) \longrightarrow ()\}),$$

we set

$$\mathcal{R} = (\{D, D^*\}, \{\mathsf{triangle} : D \longrightarrow D, \mathsf{triangle}^* : D^* \longrightarrow D^*\}),$$

and for the maps defining the presentation we pick the ones generated by the assignments

$$\mathsf{triangle} \longmapsto \mathrm{id}_D$$
$$\mathsf{triangle}^* \longmapsto \mathrm{id}_{D^*}$$

and

$$\mathsf{triangle} \longmapsto (D, \mathsf{ev}) \circ (\mathsf{coev}, D)$$
$$\mathsf{triangle}^* \longmapsto (\mathsf{ev}, D^*) \circ (D^*\mathsf{coev})$$

The rules for the term judgment are few:

- $\dfrac{(x : A) \in \Gamma}{\Gamma \vdash x \ \mathsf{term}}$;
- $\dfrac{\mathfrak{a} \in \mathfrak{A} \quad 1 \leq k \leq 2}{\mathsf{coev}^{\mathfrak{a}}_{(k)} \ \mathsf{term}}$; and

- $\dfrac{\Gamma \vdash M \ \mathsf{term} \quad \Gamma \vdash N \ \mathsf{term}}{\Gamma \vdash \mathsf{ev}\,(M, N) \ \mathsf{term}}$.

FILL IN ADMISSIBLE RULES FOR TYPING JUDGEMENTS

note that we may derive the typing judgments corresponding the compositions

$$(D) \xrightarrow{(\mathsf{coev}, D)} (D, D^*, D) \xrightarrow{(D, \mathsf{ev})} (D)$$

and

$$(D^*) \xrightarrow{(D^*, \mathsf{coev})} (D, D^*, D) \xrightarrow{(D, \mathsf{ev})} (D)$$

Using the admissible rulessing the actual rules for the type theory (modulo the consideration of 'activeness') the canonical derivation for that first morphism follows.

$$\dfrac{\dfrac{\mathsf{coev} \in \mathcal{G}\,(; D, D^*) \quad \mathfrak{a} \in \mathfrak{A} \quad (132) : (D, D, D^*) \xrightarrow{\sim} (D, D^*, D)}{x : D \vdash \left(\mathsf{coev}^{\mathfrak{a}}_{(1)}, \mathsf{coev}^{\mathfrak{a}}_{(2)}, x\right) : D} \qquad \mathsf{ev} \in \mathcal{G}\,(D^*, D)}{x : D \vdash \left(\mathsf{coev}^{\mathfrak{a}}_{(1)} \middle| \ \mathsf{ev}\left(\mathsf{coev}^{\mathfrak{a}}_{(2)}, x\right)\right)}$$

Lastly, we've axioms

$$\dfrac{}{M : D \vdash \left(\mathsf{coev}_{(1)} \middle| \ \mathsf{ev}\left(\mathsf{coev}_{(2)}, M\right)\right) = M : D}$$

and

$$\dfrac{}{N : D^* \vdash \left(\mathsf{coev}_{(2)} \middle| \ \mathsf{ev}\left(N, \mathsf{coev}_{(1)}\right)\right) = N : D^*}$$

rules enough to generate a congruence.

We've now developed a type theory for the free dual pair which endows the dual objects D and D^* with a universal notion of element. Since the notion of dual pair abstracted the instance of a pair of dual vector spaces, which in particular have actual elements, it behooves us to ask:

"how much like a vector is an term of type D"

The answer is both practical and electrifying (though perhaps the author of this blog post is too easily electrified).

It's easy enough to believe that the evaluation map

$$\mathsf{ev} : (D, D^\star) \longrightarrow ()$$

endows the terms of type D, or D^* for that matter, with structure of scalar valued functions on the other. At first glance however, the degree to which these terms are determined by these values is unclear. As we'll see, there is an enlightening and practical type theoretic perspective on this question.

Consider that, for a finite dimensional vector space V over a field k, a basis $\{\mathbf{e}_i\}_{i=1}^n$ for V and a dual basis $\{\mathbf{e}_i^*\}_{i=1}^n$ for V^* give us an elegant way to write coev and the the first triangle identity. We write

$$\begin{array}{ccc} k & \xrightarrow{\mathsf{coev}} & V \otimes V^* \\ x & \longmapsto & \sum_{i=1}^n \mathbf{e}_i \otimes \mathbf{e}_i^* \end{array}$$

and see that

$$\begin{tikzcd} V \arrow[dr, "\mathsf{coev} \otimes V"] \arrow[d, equal] & \\ V & V \otimes V^* \otimes V \arrow[l, "V \otimes \mathsf{ev}"] \end{tikzcd} \qquad \begin{tikzcd} \mathbf{v} \arrow[r, mapsto] \arrow[d, mapsto] & \left(\sum_{i=1}^n \mathbf{e}_i \otimes \mathbf{e}_i^*\right) \otimes \mathbf{v} \arrow[dl, mapsto] \\ \mathbf{v} = \sum_{i=1}^n \mathbf{e}_i^*\,(\mathbf{v}) \cdot \mathbf{e}_i & \end{tikzcd}$$

So, the first triangle identity posits that a vector is precisely determined by its values. This property of a vector seems trivial. Thinking more type theoretically however we see something more; the triangle identities impose that the way in which ev endows D and D^* with the structure of collections of scalar valued functions on the other enjoys an extensionality of sorts.

To see this more clearly, let's make a pair of notational changes to bring the parrallel to the fore. In place of writing

$$(x, y) : (D^*, D) \vdash \mathsf{ev}\,(x, y) : ()$$

we'll denote ev infix by $\_ \triangleleft \_$ and write

$$(x, y) : (D^*, D) \vdash x \triangleleft y : ()\,.$$

Similarly, in place of writing

$$\vdash \left(\mathsf{coev}_{(1)}, \mathsf{coev}_{(2)}\right) : (D, D^*)$$

we'll denote coev by the pair $\left(u, \lambda^D u\right)$ and write

$$\vdash \left(u, \lambda^D u\right) : (D, D^*)\,.$$

With this choice of notation then, the axiom which corresponds to the first triangle identity is

$$x : D \vdash \left(u \middle|\, \lambda^D u \triangleleft x\right) = x : D.$$

Then, as Shulman points out, since $=$ is a congruence with respect to substitution, if we've, for some term M, the term $\lambda^D u \triangleleft M$ appearing in the scalars of a list of terms, then we may replace all instances of u in the rest of the term with M. While a mouthful, this is a sort of 'β-reduction for duality' a relationship between function abstraction and function evaluation. Conceptually interesting in its own right, this obeservatin also yields a one line proof for a familiar theorem.

Lemma 2. *(cite original result)(cyclicity of trace)Let* $(\mathscr{C}, (\_\,,\_)\,, ())$ *be a symmetric strict monoidal category, let*

$$\left(A, A^*, \left(u, \lambda^A u\right), \_ \triangleleft \_\right)$$

and

$$\left(B, B^*, \left(v, \lambda^B v\right), \_ \triangleleft \_\right)$$

be dual pairs in $\mathscr{C}$*, and let* $f : A \longrightarrow B$ *and* $g : B \longrightarrow A$ *be morphisms in* $\mathscr{C}$*. Let* $\mathsf{tr}\,(f \circ g)$ *be the composition*

$$() \xrightarrow{\left(v, \lambda^B v\right)} (B, B^*) \xrightarrow{f \circ g} (B, B^*) \xrightarrow{(12)} (B^*, B) \xrightarrow{\_ \triangleleft \_} ()$$

and likewise let $\mathsf{tr}\,(g \circ f)$ *be the composition.*

$$() \xrightarrow{\left(u, \lambda^A u\right)} (A, A^*) \xrightarrow{g \circ f} (A, A^*) \xrightarrow{(12)} (A^*, A) \xrightarrow{\_ \triangleleft \_} ()\,.$$

Then, $\mathsf{tr}\,(f \circ g) = \mathsf{tr}\,(g \circ f)$*.*

Proof.

$$\begin{aligned}
\mathsf{tr}\,(f \circ g) &\stackrel{\mathsf{def}}{=} \left(\middle|\lambda_u^B \lhd f\,(g\,(u))\right) \\
&= \left(\middle|\lambda_u^B \lhd f\,(v)\ , \lambda_v^A \lhd g\,(u)\right) \\
&= \left(\middle|\lambda_v^A \lhd g\,(f\,(v))\right) \\
&\stackrel{\mathsf{def}}{=} \mathsf{tr}\,(g \circ f)
\end{aligned}$$

Where the judged equalities are application of 'β-reduction for a duality'. □