We give an alternate definition of first-order hyperdoctrines that makes the Beck-Chevalley condition follow naturally. The key is to formulate the definitions using the internal language of the category of presheaves. This is based on the fact that, like most algebraic structures a functor from C to the category of Heyting algebras is equivalent to a Heyting algebra internal to the category of functors $C \to \mathrm{Set}$

1 Order-theory preliminaries

Let $f : P \to Q$ be a monotone function of posets. A **right adjoint** to f is a function $g : Q_0 \to P_0$ such that

$$p \leq g(q) \iff f(p) \leq q$$

A **left adjoint** is defined dually as satisfying

$$g(q) \leq p \iff q \leq f(p)$$

Note that in either case g is unique if it exists. The analogous notion for preorders is unique up to order equivalence.

For any poset P and type X we can define the poset P^X of functions $X \to P_0$ with the pointwise ordering. We can define a monotone function $\Delta_X : P \to P^X$

$$\Delta_X(p)(x) = p$$

which we could sensibly call "weakening". We say P has **X-indexed meets** if Δ_X has a right adjoint and **X-indexed joins** if Δ_X has a left adjoint. Examples: binary meets and joins are indexed by a two-element set and top and bottom are indexed by the empty set.

Not all properties are naturally formulated as adjoints. The following is a generalization. Let P be a poset. A *downward-closed subset* or *downset* is a subset $X \subseteq P$ such that if $p \leq q$ and $q \in X$ then $p \in X$. An upper closed set/upset is dual. A *representing element* of a downset is a greatest element of the downset and a representing element of an upset is a least element of the upset. In this case we say the downset/upset is *representable*, the idea is that if $x \in X$ is the greatest element of the downset X .

$$p \in X \iff p \leq x$$

so x "represents" by its inequality predicate the subset.

Let P be a poset with a greatest element $\top$ and X be a set. Then an *equality function for X in P* is a function $=$ in $(P^X)^X$ satisfying

$$(\forall x, y.(x = y) \leq f(x, y)) \iff \forall x.\top \leq f(x, x)$$

the Leibniz/Lawvere formulation of equality. This is a representability condition for the upset defined as

$$f(x, y) \in \mathrm{U} \iff \forall x.\top \leq f(x, x)$$

2 Internal Order Theory

Let C be a category, then the category of presheaves $\mathrm{Set}^{C^{op}}$ inherits almost all nice properties of the category of sets. One way to say this is that it is always a *topos*, a model of intuitionistic type theory. So order theory internal to presheaves means "just" interpret everything in the previous section in this model of intuitionistic type theory.

But here I'll unravel some of the definitions with an eye to how they could be formalized in Agda. The definitions in this chapter should all be interpreted with the modifier "internal to presheaves on C".

A **poset** X consists of

1. A presheaf $X_0 \in \mathrm{Set}^{C^{op}}$. Below I'll write the presheaf operation $X_0 \gamma x$ as $x \circ \gamma$ or simply $x\gamma$. We typically suppress the 0 subscript.
2. A relation $\leq^X : \forall \Gamma \to X_0\Gamma \to X_0\Gamma \to \mathrm{Prop}$ which we write infix as $x \leq^X_\Gamma x'$ (and typicall suppress the X superscript) such that for any $\gamma : \Delta \to \Gamma$, if $x \leq^X_\Gamma y$ then $x\gamma \leq^X_\Delta y\gamma$.
3. Reflexivity and transitivity at each fixed Γ.
4. Antireflexivity/univalence at each fixed Γ.

Note that this is definitionally isomorphic to a functor C^{op} to Poset.

A monotone function from X to Y is

1. A family of functions $f_\Gamma : X_\Gamma \to Y_\Gamma$ where we suppress the subscript if it's clear from context.
2. that is monotone for each Γ: if $x \leq_\Gamma x'$ then $f(x) \leq_\Gamma f(x')$
3. that commutes with reindexing: $f(x)\gamma = f(x\gamma)$

Viewing the posets as functors this is definitionally isomorphic to a natural transformation.

For each poset we can define the opposite poset, the terminal poset and the product poset all by performing the operation on posets point-wise.

A more interesting operation is *powering*[1] a presheaf-internal poset X by a presheaf A. We call this internal poset X^A and it is defined as

1. The elements are given by the internal hom in the presheaf category. This can be defined explicitly as

$$X^A(\Gamma) = \mathrm{Set}^{C^{op}}(Y\Gamma \times A, X)$$

[1] Note that technically terminal and product posets could be defined this way by using the discrete presheaves $0_\Gamma = \emptyset$ and $2_\Gamma = \{0, 1\}$ but the definition simplifies considerably if we define them directly

2. We give this a point-wise ordering:

$$f \leq_\Gamma g \iff \forall \Delta, \gamma : C(\Delta, \Gamma), a \in A_\Delta . f_\Delta(\gamma, a) \leq_\Delta g_\Delta(\gamma, a)$$

   Which satisfies the reindexing condition because if $f \leq_\Gamma g$ then

$$\begin{aligned}
f\gamma \leq_\Delta g\gamma &\iff \forall \Theta, \delta : C(\Theta, \Delta), a \in A_\Theta . (f\gamma)(\delta, a) \leq_\Theta (g\gamma)(\delta, a) \\
&\iff \forall \Theta, \delta : C(\Theta, \Delta), a \in A_\Theta . f(\gamma\delta, a) \leq_\Theta (g)(\gamma\delta, a) \\
&\Leftarrow \forall \Theta, \gamma' : C(\Theta, \Gamma), a \in A_\Theta . f(\gamma', a) \leq_\Theta (g)(\gamma', a) \\
&\iff f \leq_\Gamma g
\end{aligned}$$

A downset[2] S on a poset X consists of

1. $S : \forall \Gamma \to X\Gamma \to \text{Prop}$. We write this suggestively as $x \in_\Gamma S$ or just $x \in S$.
2. that is down closed for each Γ: if $x \in_\Gamma S$ and $y \leq_\Gamma x$ then $y \in_\Gamma S$.
3. and closed under reindexing Γ, i.e. if $\gamma : \Delta \to \Gamma$ and $x \in_\Gamma S$ then $x\gamma \in_\Delta S$

This definition is simple to adapt to Agda. On paper a more abstract definition in terms of functors would be that it's a functor from C^{op} to a category of Downsets such that each downset is over the appropriate poset. Unfortunately I don't see how to do that in such a way that the downset is definitionally "over" the right poset.

Downsets over X define a poset $\mathcal{P}^{\downarrow} X$ with ordering given by implication. We can define the Yoneda embedding $Y : X \to \mathcal{P}^{\downarrow} X$ by

$$y \in_\Gamma Yx \iff y \leq_\Gamma x$$

We say that a monotone function $f : X \to \mathcal{P}^{\downarrow} Y$ is **representable** if there exists $f' : X \to Y$ such that $f = Y \circ f'$. Because the Yoneda embedding is mono, such an f' is unique if it exists. Furthermore, each $f'_\Gamma(x)$ is characterized by the following unique existence property:

$$f'_\Gamma(x) \in_\Gamma f(x) \wedge \forall y \in Y . y \in_\Gamma f(x) \Rightarrow y \leq_\Gamma f'(x)$$

So we can equivalently say that f is representable when for every $\Gamma, x : X_\Gamma$ that there exists a (necessarily unique) r such that

$$r \in_\Gamma f(x) \wedge \forall y \in Y . y \in_\Gamma f(x) \Rightarrow y \leq_\Gamma r$$

The only thing to prove is that such rs are necessarily monotone in x (exercise)[3].

Given a monotone function $f : Y \to X$ we can define a monotone function[4] $\widetilde{f} : X \to \mathcal{P}^{\downarrow} Y$ by $y \in_\Gamma \widetilde{f}(x) \iff f(y) \leq_\Gamma x$. We say f has a **right adjoint**

[2]This could also be defined by defining a poset of propositions and an internal hom of posets

[3]this is the analogue of the representability to functoriality proof Pranav and Steven showed

[4]names here are not bad and should not be used in Agda code

when $\widetilde{f}(x)$ is representable. Given a presheaf A, we say X has **A-indexed meets** when the monotone function

$$\Delta : X \to X^A$$

defined by

$$\Delta_\Gamma(x)(\gamma, a) = x\gamma$$

has a right adjoint. We can similarly define that X has **a top element, or nullary meet** when the unique

$$! : X \to 1$$

has a right adjoint. Note that this gives a morphism $\top : 1 \to X$ which externally is what we might call a *section*

$$\top : \forall \Gamma \to X_\Gamma$$

satisfying

$$\top_\Gamma \gamma = \top_\Delta$$

and representability says that each $\top$ is a top element of X_Γ. So requiring that X has a top element is equivalent to defining X to be a functor form C^{op} to the category of posets with a top elements and morphisms that preserve them.

And define that X has **binary meets** when

$$\Delta : X \to X \times X$$

defined by

$$\Delta_\Gamma(x) = (x, x)$$

has a right adjoint. Externally this is similar to the case for top, we get a meet in each poset that is preserved by reindexing.

Now we can take all the above definitions about downsets and representability and reformulate them dually for upsets. Or we can define

$$\mathcal{P}^\uparrow X = (\mathcal{P}^\downarrow X^{op})^{op}$$

And with some careful op-ing re-use all the theory of representability shown above. For instance we immediately get dual noitons of A-indexed joins, nullary and binary joins.

Finally, let A be a presheaf and X be a poset with a top element. Define a monotone function

$$c : (X^{A \times A}) \to X^A$$

by

$$c_\Gamma(f)(\gamma, a) = f_\Gamma(\gamma, (a, a))$$

This should be easy to prove monotone. Note that the product presheaf $A \times A$ is constructed pointwise.

Then we get $\widetilde{c} : X^A \to \mathcal{P}^\uparrow X^{A \times A}$. Further if X has a top element then X^A does as well which we call $\top : 1 \to X^A$. We say X has an A-equality predicate when

$$\widetilde{c} \circ \top : 1 \to \mathcal{P}^\uparrow X^{A \times A}$$

is representable.

3 First-Order Hyperdoctrines

A first-order hyperdoctrine over a cartesian category C consists of

1. (Propositional Logic) A biHeyting algebra L internal to $\mathrm{Set}^{C^{op}}$.
2. (Universal Quantifiers) Such that for every $A \in C$, L internally has YA-indexed meets
3. (Existential Quantifiers) Such that for every $A \in C$, L internally has YA-indexed joins
4. (Equality) Such that L has an internal equality function for YA

4 Making Hyperdoctrines more Compositional

The definition above is not a modular definition in that certain elements (distributivity of joins, existentials, equality?) follow automatically only because we assume we are working with a Heyting algebra. Instead, let's describe everything in a modular fashion, which also means describing something very close to the syntax itself.

A **simple category with families** consists of

1. A category C with a terminal object
2. A type T of *types*, and a function assigning a presheaf on C to each type $\mathrm{tm} : T \to \mathcal{P}^o C$.
3. For each $\Gamma \in C$ and $A \in T$, an object $\Gamma \times A \in C$ representing

$$\Delta \to (\Gamma \times A) \cong \Delta \to \Gamma \times \mathrm{tm}(A)_\Delta$$

A **hyperdoctrine** extends this with the following

1. A poset P with a top element internal to $\mathcal{P}^o C$
2. A presheaf F of *formulae* with a morphism $\mathrm{pf} : F \to \mathcal{P}^\downarrow P$
3. A morphism $\wedge : P \times F \to P$ satisfying (in the internal language)

$$\Psi \leq \Phi \wedge \phi \iff \Psi \leq \Phi \wedge \mathrm{pf}(\phi)_\Psi$$

This corresponds precisely to the following explicit substitution calculus, presented in traditional type theoretic notation. This felt like a total waste of time to me so please let me know if you found it helpful. First there are 8 judgments, whose formation rules are given as follows

$$\Gamma \text{ ctx} \qquad \frac{\Delta \text{ ctx} \qquad \Gamma \text{ ctx}}{\Delta \vdash \gamma : \Gamma}$$

$$A \text{ type} \qquad \frac{\Gamma \text{ ctx} \qquad A \text{ type}}{\Gamma \vdash M : A}$$

$$\frac{\Gamma \text{ ctx}}{\Gamma \,|\, \Phi \text{ pctx}} \qquad \frac{\Gamma \text{ ctx} \qquad \Gamma \,|\, \Psi \text{ pctx} \qquad \Gamma \,|\, \Phi \text{ pctx}}{\Gamma \,|\, \Psi \vdash \Phi}$$

$$\frac{\Gamma \text{ ctx}}{\Gamma \,|\, \phi \text{ formula}} \qquad \frac{\Gamma \text{ ctx} \qquad \Gamma \,|\, \Phi \text{ pctx} \qquad \Gamma \,|\, \phi \text{ formula}}{\Gamma \,|\, \Phi \vdash \phi}$$

And the following basic constructions on contexts, substitutions and terms

$$\cdot \text{ ctx} \qquad \frac{\Gamma \text{ ctx} \qquad A \text{ type}}{\Gamma, A \text{ ctx}} \qquad \frac{\Delta \vdash \gamma : \Gamma \qquad \Theta \vdash \delta : \Delta}{\Theta \vdash \gamma[\delta] : \Gamma} \qquad \Gamma \vdash \Gamma : \Gamma$$

$$\Delta \vdash \cdot : \cdot \qquad \Gamma, A \vdash \text{wk} : \Gamma \qquad \frac{\Delta \vdash \gamma : \Gamma \qquad \Delta \vdash M : A}{\Delta \vdash (\gamma, M) : \Gamma, A}$$

$$\frac{\Gamma \vdash M : A \qquad \Delta \vdash \gamma : \Gamma}{\Delta \vdash M[\gamma] : A} \qquad \Gamma, A \vdash \text{var} : A$$

And the analogous constructs for the logic:

$$\Gamma \,|\, \cdot \text{ pctx} \qquad \frac{\Gamma \,|\, \Phi \text{ pctx} \qquad \Gamma \,|\, \phi \text{ formula}}{\Gamma \,|\, \Phi, \phi \text{ pctx}} \qquad \Gamma \,|\, \Phi \vdash \Phi$$

$$\frac{\Gamma \,|\, \Psi \vdash \Psi' \qquad \Gamma \,|\, \Psi' \vdash \Phi}{\Gamma \,|\, \Psi \vdash \Phi} \qquad \Gamma \,|\, \Phi \vdash \cdot \qquad \frac{\Gamma \,|\, \Psi \vdash \Phi \qquad \Gamma \,|\, \Psi \vdash \phi}{\Gamma \,|\, \Psi \vdash \Phi, \phi}$$

$$\Gamma \,|\, \Phi, \phi \vdash \Phi \qquad \Gamma \,|\, \Phi, \phi \vdash \phi$$

Subject to the following equations on the substitutions and terms.

$$\Gamma[\gamma] = \gamma \qquad \gamma[\Gamma] = \gamma \qquad \gamma[\delta][\theta] = \gamma[\delta[\theta]] \qquad \frac{\Delta \vdash \gamma : \cdot}{\gamma = \cdot} \qquad \frac{\Delta \vdash \gamma : \Gamma, A}{\gamma = (\text{wk}[\gamma], \text{var}[\gamma])}$$

$$\text{wk}[\gamma, M] = \gamma \qquad M[\Gamma] = M \qquad M[\gamma][\delta] = M[\gamma[\delta]] \qquad \text{var}[\gamma, M] = M$$

4.1 Connectives, Modularly

TODO

5 Syntax

We can give a type theoretic syntax for first-order hyperdoctrines by a combination of a simple type theory with terms $\Gamma \vdash M : A$ as well as intuitionistic propositional logic parameterized by a context Γ i.e. formulae $\Gamma \mid \phi$ and proofs $\Gamma \mid \Phi \vdash \phi$. The universal quantifiers are presented as follows:

$$\frac{\Gamma, x : A \mid \phi}{\Gamma \mid \forall x : A.\phi} \qquad (\forall x : A.\phi)[\gamma] = \forall x : A.\phi[\gamma] \qquad \frac{\Gamma, x : A \mid \Phi \vdash \phi}{\Gamma \mid \Phi \vdash \forall x : A.\phi}$$

$$\frac{\Gamma \vdash M : A}{\Gamma \mid \forall x : A.\phi \vdash \phi[M/x]}$$

Existential quantification is similar but you can eliminate it in any context. Note that this relies on the model being a Heyting algebra, we need some kind of CT structure thing to directly model these rules

$$\frac{\Gamma, x : A \mid \phi}{\Gamma \mid \exists x : A.\phi} \qquad (\exists x : A.\phi)[\gamma] = \exists x : A.\phi[\gamma] \qquad \frac{\Gamma, x : A \mid \Phi, \phi \vdash \psi}{\Gamma \mid \Phi, \exists x : A.\phi \vdash \psi}$$

$$\frac{\Gamma \vdash M : A}{\Gamma \mid \phi[M/x] \vdash \exists x.\phi}$$

Finally we have equality:

$$\frac{\Gamma \vdash M : A \qquad \Gamma \vdash N : A}{\Gamma \mid M =_A N} \qquad (M =_A N)[\gamma] = M[\gamma] =_A N[\gamma]$$

$$\Gamma, x : A \mid \cdot \vdash x =_A x \qquad \frac{\Gamma, x : A \mid \Phi[y/x] \vdash \psi[y/x]}{\Gamma, x : A, y : A \mid \Phi, x =_A y \vdash \psi}$$

6 Grothendieck Construction/Logical Relations

Given a poset X internal to presheaves on C we can define a category $\Sigma_C X$ as follows:

1. The objects are dependent pairs of $\Gamma \in C$ and $x \in X_\Gamma$
2. A morphism from Δ, y to Γ, x is a $\gamma : C(\Delta, \Gamma)$ such that
$$y \leq_\Delta x\gamma$$
3. identity and composition are given by id/comp in C, verifying that the side-condition is preserved.

We also clearly have a “first-projection” functor $\pi : \Sigma_C X \to C$.

Then we have the following theorems:

1. If C has a terminal object and the poset X internally has a terminal object then $\Sigma_C X$ has a terminal objects and π preserves it. Similarly for binary products/binary meets.

2. If C has exponentials and the poset X internally has Heyting implications and universal quantifiers over objects of C then $\Sigma_C X$ has exponentials and π preserves them.

3. If C has an initial object and the poset X internally has a bottom element then $\Sigma_C X$ has an initial object and π preserves is.

4. If C has binary coproducts and the poset X has internal binary joins and X has existential quantifiers and equality for objects of C then $\Sigma_C X$ has binary coproducts and π preserves them.

Additionally in each of these cases the functor π preserves the structure (products/coproducts/exponentials).