Triposes

Luna Strah

April 3, 2025

00

1 Notation

Let $\mathcal{C}$ be a cartesian-closed category and $P : \mathcal{C}^{\mathsf{op}} \to \mathrm{HeytAlg}$ a functor between poset-enriched categories. Also, we will call objects of $\mathcal{C}$ *types*. We also fix a *choice* of finite products and exponents in $\mathcal{C}$.

2 Definition of a tripos

Definition 2.1 (Tripos)**.** A functor P is a *tripos* when it satisfies the following:

1. For all types X and Y and maps $f : X \to Y$, the map Pf has monotone left and right adjoints, $\exists_f$ and $\forall_f$ respectively
2. These further satisfy the Beck-Chevalley condition: If

$$\begin{array}{ccc} X & \xrightarrow{f} & Y \\ {\scriptstyle g}\downarrow & & \downarrow{\scriptstyle h} \\ Z & \xrightarrow[k]{} & W \end{array}$$

   is a pullback square then $\forall_f \circ Pg = Ph \circ \forall_k$

3. There is a type Σ and an element $\sigma \in P(\Sigma)$ such that for every object X and every $\varphi \in P(X)$ there is a morphism $[\varphi] : X \to \Sigma$ such that $\varphi = P([\varphi])(\sigma)$

3 Definition of a tripos language

A *context* is a list of pairs of variables and objects of $\mathcal{C}$. We write it like $x_1 : X_1, ..., x_n : X_n$, where x_i are some variables and X_i some types.

Let $\Gamma = x_1 : X_1, ..., x_n : X_n$ be an arbitrary context.

Terms of type X in context Γ are

- variables x_i,
- the unique element of the terminal type $\mathtt{tt}$,
- ordered pairs $\langle a, b\rangle$, where a and b are also terms in context Γ,

- the first and second pojections of terms `fst` and `snd`, or
- terms $f(t)$, where t is a term of type Y in context Γ and $f : Y \to X$ a morphism.

Write $\Gamma \vdash_t t$ to mean "the term t in context Γ".
We can define the evaluation of terms of type X in context Γ to morphisms $X_1 \times \cdots \times X_n \to X$:

- $\Gamma \vdash_t x_i \Downarrow \pi_i$, where π_i is the projection from $X_1 \times \cdots \times X_n$ ommitting the i-th component,
- $\Gamma \vdash_t \texttt{tt} \Downarrow !_{X_1 \times \cdots \times X_n}$,
- $\Gamma \vdash_t \langle a, b \rangle \rightsquigarrow \langle \Gamma \vdash_t a, \Gamma \vdash_t b \rangle$, where $\langle\rangle$ on the right-hand side is the pairing function in $\mathcal{C}$,
- $\Gamma \vdash_t \texttt{fst} \Downarrow \pi$ and $\Gamma \vdash_t \texttt{snd} \Downarrow \pi'$, where π and π' are the first and second projections in $\mathcal{C}$ respectively, and
- $\Gamma \vdash_t f(t) \rightsquigarrow f \circ (\Gamma \vdash_t t)$.

Formulas of type X in context Γ are

- $R(t)$, where t is a term of type Y in context Γ and $R \in P(Y)$,
- one of the usual Heyting algebra formers, or
- the quantifiers $\exists x : X,\ \varphi$ and $\forall x : X,\ \varphi$, where φ is a formula.

Write $\Gamma \vdash_f t$ to mean "the formula t in context Γ".
Again we can define the evaluation of formulas in context Γ to elements of Heyting algebras:

- $\Gamma \vdash_f R(t) \Downarrow P(t)(R)$,
- the Heyting algebra formers get evaluated to themselves in the obvious way,
- if x is a free variable in φ then $\Gamma \vdash_f Qx : X,\ \varphi \Downarrow Q_f(\Gamma, x : X \vdash_f \varphi)$, where Q is a quantifier, and
- if x is not a free variable in φ then $\Gamma \vdash_f Qx : X,\ \varphi \Downarrow Q_\pi(\pi^*(\Gamma, x : X \vdash_f \varphi))$, where Q is a quantifier.

Also, let $P(\Gamma)$ mean the same as $P(X_1, ..., X_n)$.
Note: due to implementation issues, there is an intermediary type of formulas, defined inductively with the above constructions, quotiented with equivalence up to evaluation by the above rules. This gives us a type equivalent to the Heyting algebras $P(X)$, but it behaves much better with rewriting in Lean.

Definition 3.1 (Truth)**.** We call a formula φ in context Γ *true* when it evaluates to the top element of $P(\Gamma)$. We denote this by $\Gamma \vdash \varphi$.

Lemma 3.2 (Characterization of implication)**.** *Showing* $\Gamma \vdash \varphi \Rightarrow \psi$ *is equivalent to showing* $(\Gamma \vdash_f \varphi) \leq (\Gamma \vdash_f \psi)$.

Proof. The expression $\Gamma \vdash \varphi \Rightarrow \psi$ is defined to be $\top = (\Gamma \vdash_f \varphi \Rightarrow \psi) = (\Gamma \vdash_f \varphi) \Rightarrow (\Gamma \vdash_f \psi)$. Equality with $\top$ is the same as being greater than $\top$. Now using that $\sqcap$ is left adjoint to $\Rightarrow$ we get $(\Gamma \vdash_f \varphi) \leq (\Gamma \vdash_f \psi)$. □

4 Definition of a partial equivalence relation

Definition 4.1. A *partial equivalence relation* (PER) on the type X over a tripos P is an element $\rho_X \in P(X \times X)$ that satisfies

1. $x : X, y : X \vdash \rho_X(x, y) \Rightarrow \rho_X(y, x)$
2. $x : X, y : X, z : Z \vdash \rho_X(x, y) \sqcap \rho_X(y, z) \Rightarrow \rho_X(x, z)$

The first condition is called *symmerty* and the second *transitivity*, so a PER indeed a partial equivalence relation in the internal language.

From now on we will write $x =_X y$ to mean $\rho_X(x, y)$, ommitting the subscript where the underlying type is obvious.

Definition 4.2. A morphism between PER's ρ_X and ρ_Y over P is an element $f : P(X \times Y)$ that satisfies

1. $x : X, x' : X, y : Y \vdash x = x' \sqcap f(x', y) \Rightarrow f(x, y)$
2. $x : X, y : Y, y' : Y \vdash f(x, y) \sqcap y = y' \Rightarrow f(x, y')$
3. $x : X, y : Y, y' : Y \vdash f(x, y) \sqcap f(x, y') \Rightarrow y = y'$
4. $x : X \vdash x = x \Rightarrow \exists y : Y,\ f(x, y)$

Remark. *Again, we will write $f(x) = y$ to mean $f(x, y)$. We will also write $f : \rho_X \to \rho_Y$ to mean "a morphism f betweens PER's ρ_X and ρ_Y".*

The first two properties here say the morphism is coherent with the PERs on the domain and codomain, and the other two are *uniqueness* and *totality* of a relation. So morphisms are defined as "functional relations internal to P".

Construction 1. Let $f : \rho_X \to \rho_Y$ and $g : \rho_Y \to \rho_Z$ be PER morphisms. Their composite is defined to be the composte of relations in the internal language, in other words

$$g \circ f := x : X, z : Z \vdash_f \exists y : Y,\ f(x) = y \sqcap g(y) = z.$$

Proof. Had we shown a soundness theorem, we could just say "it is constructively true that the composition of functional relations is a functional relation", and be done with the proof. Nonetheless, the proof itself is relatively straight-forward.

To save space we will ommit the context in the following derivations. Also, to avoid ambiguity, we will denote eqality in a Heyting algebra using $\equiv$.

$$\begin{aligned} x = x' \sqcap g \circ f(x') = z &\equiv x = x' \sqcap \exists y : Y,\ f(x') = y \sqcap g(y) = z \\ &\equiv \exists y : Y,\ x = x' \sqcap f(x') = y \sqcap g(y) = z \\ &\leq \exists y : Y,\ f(x) = y \sqcap g(y) = z \\ &\equiv g \circ f(x) = z \end{aligned}$$

$$\begin{aligned} g \circ f(x) = z \sqcap z = z' &\equiv \exists y : Y,\ f(x) = y \sqcap g(y) = z \sqcap z = z' \\ &\leq \exists y : Y,\ f(x) = y \sqcap g(y) = z' \\ &\equiv g \circ f(x) = z' \end{aligned}$$

$$\begin{aligned}
g \circ f(x) = z \sqcap g \circ f(x) = z' &\equiv \exists y : Y,\ f(x) = y \sqcap g(y) = z \sqcap \exists y' : Y,\ f(x) = y' \sqcap g(y') = z' \\
&\leq \exists y : Y,\ y' : Y,\ f(x) = y \sqcap g(y) = z \sqcap f(x) = y' \sqcap g(y') = z' \\
&\equiv \exists y : Y,\ y' : Y,\ g(y) = z \sqcap f(x) = y \sqcap f(x) = y' \sqcap g(y') = z' \\
&\leq \exists y : Y,\ y' : Y,\ g(y) = z \sqcap y = y' \sqcap g(y') = z' \\
&\leq \exists y : Y,\ y' : Y,\ g(y) = z \sqcap g(y) = z' \\
&\leq \exists y : Y,\ z = z' \\
&\equiv z = z'
\end{aligned}$$

$$\begin{aligned}
x = x &\leq \exists y : Y,\ f(x) = y \\
&\equiv \exists y : Y,\ f(x) = y \sqcap y = y \\
&\leq \exists y : Y,\ f(x) = y \sqcap \exists z : Z,\ g(y) = z \\
&\equiv \exists z : Z,\ \exists y : Y,\ f(x) = y \sqcap g(y) = z \\
&\equiv \exists z : Z,\ g \circ f(x) = z
\end{aligned}$$

□

Construction 2. The identity morphism on ρX is ρX itself.

Proof. In this proof we break convention and will use letters a, b,... to denote elements of X.

$$\begin{aligned}
a = b \sqcap \rho X(b) = c &\equiv a = b \sqcap b = c \\
&\equiv a = c \\
&\equiv \rho X(a) = c
\end{aligned}$$

$$\begin{aligned}
\rho X(a) = b \sqcap b = c &\equiv a = b \sqcap b = c \\
&\leq a = c \\
&\equiv \rho X(a) = c
\end{aligned}$$

$$\begin{aligned}
\rho X(a) = b \sqcap \rho X(a) = c &\equiv a = b \sqcap a = c \\
&\equiv b = a \sqcap a = c \\
&\leq b = c
\end{aligned}$$

The last property to show ρX is a PER morphism, $a = a \leq \exists b : X,\ \rho X(a) = b$, is somewhat tricky to prove using just the internal language, so we need to annotate a few things.

The LHS is actually $\delta^* \rho X$ and the RHS is $\exists_\pi \rho X$, where $\delta : X \to X \times X$ is the diagonal morphism and $\pi : X \times X \to X$ the fist projection.

So we wish to show $\delta^* \rho X \leq \exists_\pi \rho X$. By the $f^* \dashv \forall_f$ adjunction, this is equivalent to $\rho X \leq \forall_\delta \exists_\pi \rho X = \forall_\delta \delta^* \pi^* \exists_\pi \rho X$. But this follows immediately from the units of the $\forall$ and $\exists$ adjunctions.

All that is left is to show composition with the identity is the identity. We only prove one identity law, since the proof of the other one is symmetric.

$$
\begin{aligned}
f(x) = y &\equiv x = x \sqcap f(x) = y \\
&\equiv \exists x' : X,\ x = x' \sqcap f(x') = y \\
&\equiv f \circ \rho X(x) = y
\end{aligned}
$$

□

5 Tripos to topos construction

Definition 5.1. To a tripos P (over a category $\mathcal{C}$) we can associate the category $\mathcal{C}[P]$ of types along with partial equivalence relations over P with morphisms as above.

Theorem 5.2 (Pitts)**.** *The category $\mathcal{C}[P]$ is a topos.*