Tiny Types

Mitchell Riley

CQTS

April 21st 2024

Definition

A tiny object $\mathbb{T}$ in a category $\mathcal{C}$ is one for which $(\mathbb{T} \to -) : \mathcal{C} \to \mathcal{C}$ has a *right* adjoint $\surd : \mathcal{C} \to \mathcal{C}$.

- 1 in Set.
- The interval $\mathbb{I}$ in many models of cubical type theory.
- The infinitesimal disk $D :\equiv \{x : \mathbb{R} \mid x^2 = 0\}$ in models of synthetic differential geometry.
- The universal object in the topos classifying objects, $[\mathsf{FinSet}, \mathsf{Set}]$.
- Any representable presheaf for a site with finite products.

Definition

A tiny object $\mathbb{T}$ in a category $\mathcal{C}$ is one for which $(\mathbb{T} \to -) : \mathcal{C} \to \mathcal{C}$ has a *right* adjoint $\sqrt{\ } : \mathcal{C} \to \mathcal{C}$.

- 1 in Set.
- The interval $\mathbb{I}$ in many models of cubical type theory.
- The infinitesimal disk $D :\equiv \{x : \mathbb{R} \mid x^2 = 0\}$ in models of synthetic differential geometry.
- The universal object in the topos classifying objects, $[\mathsf{FinSet}, \mathsf{Set}]$.
- Any representable presheaf for a site with finite products.

In a model of SDG, let $D :\equiv \{x : \mathbb{R} \mid x^2 = 0\}$.

The *tangent space* of X is the type $TX :\equiv D \to X$. A (not-necessarily linear) *1-form* on X is a map $TX \to \mathbb{R}$.

These correspond to maps $X \to \sqrt{\mathbb{R}}$.

- [LOPS18] axiomatises:

$$\begin{aligned} \surd &: \flat\mathcal{U} \to \mathcal{U} \\ \mathsf{R} &: \flat((\mathbb{T} \to A) \to B) \simeq \flat(A \to \surd B) \\ \mathsf{R}\text{-nat} &: \{R \text{ is natural in } A\} \end{aligned}$$

- [Mye22] improves to:

$$\begin{aligned} \surd &: \flat\mathcal{U} \to \mathcal{U} \\ \varepsilon &: (\mathbb{T} \to \surd B) \to B \\ \mathsf{e} &: \mathsf{isEquiv}(\flat(A \to \surd B) \to \flat((\mathbb{T} \to A) \to B)) \end{aligned}$$

Previous Approaches

- [LOPS18] axiomatises:

$$\begin{aligned} \surd &: \flat\mathcal{U} \to \mathcal{U} \\ \mathsf{R} &: \flat((\mathbb{T} \to A) \to B) \simeq \flat(A \to \surd B) \\ \mathsf{R}\text{-nat} &: \{R \text{ is natural in } A\} \end{aligned}$$

- [Mye22] improves to:

$$\begin{aligned} \surd &: \flat\mathcal{U} \to \mathcal{U} \\ \varepsilon &: (\mathbb{T} \to \surd B) \to B \\ \mathsf{e} &: \mathsf{isEquiv}(\flat(A \to \surd B) \to \flat((\mathbb{T} \to A) \to B)) \end{aligned}$$

- [ND21] targets a right adjoint to "telescope quantification":

$$\frac{\Gamma, (\forall i : \mathbb{T}.\Delta) \vdash A \text{ type}}{\Gamma, i : \mathbb{T}, \Delta \vdash \between A \text{ type}}$$

- The system of MTT modalities we saw in Daniel's talk with an axiom $\Gamma, \{p\} \equiv \Gamma, i : \mathbb{I}$.

- [ND21] targets a right adjoint to "telescope quantification":

$$\frac{\Gamma, (\forall i : \mathbb{T}.\Delta) \vdash A \ \mathsf{type}}{\Gamma, i : \mathbb{T}, \Delta \vdash \between A \ \mathsf{type}}$$

- The system of MTT modalities we saw in Daniel's talk with an axiom $\Gamma, \{p\} \equiv \Gamma, i : \mathbb{I}$.

Desiderata

- No axioms
- Comprehensible rules (relatively speaking)
- Usable by hand
- Normalisable

$$(-, x : A) \text{ `}\dashv\text{' } (x : A) \to -$$

$$\frac{\Gamma, x : A \vdash b : B}{\Gamma \vdash \mathsf{lam}(b) : (x : A) \to B} \qquad \frac{\Gamma \vdash f : (x : A) \to B}{\Gamma, x : A \vdash \mathsf{unlam}(f) : B}$$

$$\mathsf{unlam}(\mathsf{lam}(b)) \equiv b \qquad f \equiv \mathsf{lam}(\mathsf{unlam}(f))$$

The Less Amazing Right Adjoint

$$(-, x : A) \text{ `}\dashv\text{' } (x : A) \to -$$

$$\frac{\Gamma, x : A \vdash b : B}{\Gamma \vdash \mathsf{lam}(b) : (x : A) \to B} \qquad \frac{\Gamma \vdash f : (x : A) \to B \qquad \Gamma \vdash a : A}{\Gamma \vdash \mathsf{app}(f, a) : B[a/x]}$$

$$\mathsf{app}(\mathsf{lam}(b), a) \equiv b[a/x] \qquad f \equiv \mathsf{lam}(\mathsf{app}(f, x))$$

The Fitch-Style Right Adjoint

$$\mathcal{L} \text{ `}\dashv\text{' } \mathcal{R}$$

$$\frac{\Gamma, \mathcal{L} \vdash a : A}{\Gamma \vdash \mathsf{lam}(a) : RA} \qquad \frac{\Gamma \vdash f : \mathcal{R}A}{\Gamma, \mathcal{L} \vdash \mathsf{unlam}(f) : A}$$

$$\mathsf{unlam}(\mathsf{lam}(b)) \equiv b \qquad f \equiv \mathsf{lam}(\mathsf{unlam}(f))$$

Following [BCMEPS20]. By $\Gamma, \mathcal{L}$ I mean $\mathcal{L}(\Gamma)$.

The Fitch-Style Right Adjoint

$$\mathcal{L} \text{ `}\dashv\text{' } \mathcal{R}$$

$$\frac{\Gamma, \mathcal{L} \vdash a : A}{\Gamma \vdash \mathsf{lam}(a) : RA} \qquad \frac{\Gamma \vdash f : \mathcal{R}A \qquad \mathcal{L} \notin \Gamma'}{\Gamma, \mathcal{L}, \Gamma' \vdash \mathsf{unlam}(f) : A}$$

$$\mathsf{unlam}(\mathsf{lam}(b)) \equiv b \qquad f \equiv \mathsf{lam}(\mathsf{unlam}(f))$$

Following [BCMEPS20]. By $\Gamma, \mathcal{L}$ I mean $\mathcal{L}(\Gamma)$.

The FitchTT-Style Right Adjoint

$$\mathcal{E} \dashv \mathcal{L} \text{ `}\dashv\text{' } \mathcal{R}$$

$$\frac{\Gamma, \mathcal{L} \vdash b : B}{\Gamma \vdash \mathsf{lam}(b) : \mathcal{R}B} \qquad \frac{\Gamma, \mathcal{E} \vdash f : \mathcal{R}B}{\Gamma, \mathcal{E}, \mathcal{L} \vdash \mathsf{unlam}(f) : B}$$

$$\mathsf{unlam}(\mathsf{lam}(b)) \equiv b \qquad f \equiv \mathsf{lam}(\mathsf{unlam}(f))$$

Following [GCKGB22].

The FitchTT-Style Right Adjoint

$$\mathcal{E} \dashv \mathcal{L} \text{`}\dashv\text{'} \mathcal{R}$$

$$\frac{\Gamma, \mathcal{L} \vdash b : B}{\Gamma \vdash \mathsf{lam}(b) : \mathcal{R}B} \qquad \frac{\Gamma, \mathcal{E} \vdash f : \mathcal{R}B}{\Gamma \vdash \mathsf{app}(f) : B[\eta]}$$

$$\mathsf{app}(\mathsf{lam}(b)) \equiv b[\eta] \qquad f \equiv \mathsf{lam}(\mathsf{app}(f[\varepsilon]))$$

where

$$\frac{}{\Gamma \vdash \eta : \Gamma, \mathcal{E}, \mathcal{L}} \qquad \frac{}{\Gamma, \mathcal{L}, \mathcal{E} \vdash \varepsilon : \Gamma}$$

Following [GCKGB22].

The Amazing Right Adjoint

$$(-, i : \mathbb{T}) \dashv (-, \text{🔒}) \text{ ‘}\dashv\text{’ } \surd$$

$$\frac{\Gamma, \text{🔒} \vdash b : B}{\Gamma \vdash \text{🔒}.b : \surd B} \qquad \frac{\Gamma, i : \mathbb{T} \vdash r : \surd B}{\Gamma \vdash r(\text{⅄}i.) : B[\text{⅄}i./\text{🔒}]}$$

$$(\text{🔒}.b)(\text{⅄}i.) \equiv b[\text{⅄}i./\text{🔒}] \qquad r \equiv \text{🔒}.(r[i/\text{🔑}](\text{⅄}i.))$$

where

$$\frac{}{\Gamma \vdash [\text{⅄}i./\text{🔒}] : \Gamma, i : \mathbb{T}, \text{🔒}} \qquad \frac{}{\Gamma, \text{🔒}, i : \mathbb{T} \vdash [i/\text{🔑}] : \Gamma}$$

Specialising to a tiny type, [Ril24].

$$(-, i : \mathbb{T}) \dashv (-, \text{🔒}_{\mathcal{N}}) \text{ `}\dashv\text{' } \surd$$

$$\frac{\Gamma, \text{🔒}_{\mathcal{N}} \vdash b : B}{\Gamma \vdash \text{🔒}_{\mathcal{N}}.b : \surd_{\mathcal{N}} B} \qquad \frac{\Gamma, i : \mathbb{T} \vdash r : \surd_{\mathcal{N}} B}{\Gamma \vdash r(\curlyvee i.) : B[\curlyvee i./\text{🔒}_{\mathcal{N}}]}$$

$$(\text{🔒}_{\mathcal{N}}.b)(\curlyvee i.) \equiv b[\curlyvee i./\text{🔒}_{\mathcal{N}}] \qquad r \equiv \text{🔒}_{\mathcal{N}}.(r[i/\text{🔑}_{\mathcal{N}}](\curlyvee i.))$$

where

$$\frac{}{\Gamma \vdash [\curlyvee i./\text{🔒}_{\mathcal{N}}] : \Gamma, i : \mathbb{T}, \text{🔒}_{\mathcal{N}}} \qquad \frac{}{\Gamma, \text{🔒}_{\mathcal{N}}, i : \mathbb{T} \vdash [i/\text{🔑}_{\mathcal{N}}] : \Gamma}$$

Specialising to a tiny type, [Ril24].

$$\textsc{counit}\ \frac{\Gamma \vdash a : A \qquad \Gamma, 🔒, \Gamma' \vdash t : \mathbb{T} \qquad 🔒 \notin \Gamma'}{\Gamma, 🔒, \Gamma' \vdash a[t/🔑] : A[t/🔑]}$$

Roughly:

$$[(\mathbb{T} \to \Gamma) \times \Gamma'] \longrightarrow [(\mathbb{T} \to \Gamma) \times \Gamma' \times \mathbb{T}] \longrightarrow \Gamma \longrightarrow A$$

If there are many locks to get past:

$$\textsc{counit}\ \frac{\Gamma \vdash a : A \qquad \Gamma, \Gamma' \vdash t_i : \mathbb{T} \text{ for } \mathcal{L}_i \in \mathsf{locks}(\Gamma')}{\Gamma, \Gamma' \vdash a[t_1/🔑_{\mathcal{L}_1}, \ldots, t_n/🔑_{\mathcal{L}_n}] : A[t_1/🔑_{\mathcal{L}_1}, \ldots, t_n/🔑_{\mathcal{L}_n}]}$$

The Counit

The counit travels down to free variables and gets stuck:

$$
\begin{aligned}
(x, y)[i/\text{🔑}] &\equiv (x[\![i/\text{🔑}]\!], y[\![i/\text{🔑}]\!]) \\
(\lambda y.x + y)[i/\text{🔑}] &\equiv (\lambda y.x[\![i/\text{🔑}]\!] + y)
\end{aligned}
$$

$$
\text{VAR}\ \frac{\Gamma, x : A, \Gamma' \vdash t_i : \mathbb{T} \text{ for } \mathcal{L}_i \in \mathsf{locks}(\Gamma')}{\Gamma, x : A, \Gamma' \vdash x[\![t_1/\text{🔑}_{\mathcal{L}_1}, \ldots, t_n/\text{🔑}_{\mathcal{L}_n}]\!] : A[t_1/\text{🔑}_{\mathcal{L}_1}, \ldots, t_n/\text{🔑}_{\mathcal{L}_n}]}
$$

$$\textsc{unit}\ \frac{\Gamma, i : \mathbb{T}, \text{🔒} \vdash a : A}{\Gamma \vdash a[\text{⅄}i./\text{🔒}] : A[\text{⅄}i./\text{🔒}]}$$

Roughly:

$$\Gamma \longrightarrow [\mathbb{T} \to (\Gamma \times \mathbb{T})] \longrightarrow A$$

Also travels down to free variables:

$$(x, y)[\text{⅄}i./\text{🔒}] \equiv (x[\text{⅄}i./\text{🔒}], y[\text{⅄}i./\text{🔒}])$$
$$(\lambda y.x + y)[\text{⅄}i./\text{🔒}] \equiv (\lambda y.x[\text{⅄}i./\text{🔒}] + y)$$

The Twain Shall Meet

But! To have been used at all, these variables *must* have an attached key.

$$
\begin{aligned}
(x\llbracket t/\text{🔑}\rrbracket, y\llbracket t/\text{🔑}\rrbracket)[\text{⅄}i./\text{🔒}] &\equiv (x\llbracket t/\text{🔑}\rrbracket[\text{⅄}i./\text{🔒}], y\llbracket t/\text{🔑}\rrbracket[\text{⅄}i./\text{🔒}]) \\
(\lambda y.x\llbracket t/\text{🔑}\rrbracket + y)[\text{⅄}i./\text{🔒}] &\equiv (\lambda y.x\llbracket t/\text{🔑}\rrbracket[\text{⅄}i./\text{🔒}] + y)
\end{aligned}
$$

When a unit meets a stuck counit, it turns back into a regular substitution:

$$a\llbracket t/\text{🔑}\rrbracket[\text{⅄}i./\text{🔒}] :\equiv a[t/i]$$

That is:

$$
\begin{aligned}
[\Gamma \times \Gamma'] &\longrightarrow [(\mathbb{T} \to \Gamma \times \mathbb{T}) \times \Gamma'] \\
&\longrightarrow [(\mathbb{T} \to \Gamma \times \mathbb{T}) \times \Gamma' \times \mathbb{T}] \longrightarrow \Gamma \longrightarrow A
\end{aligned}
$$

Delayed Substitutions?

These are not just substitutions waiting to be "activated".

$$x : \mathbb{T}, \text{🔒}_{\mathcal{L}}, \text{🔒}_{\mathcal{K}} \vdash x[\![1/\text{🔑}_{\mathcal{L}}, 2/\text{🔑}_{\mathcal{K}}]\!] : \mathbb{T}$$

$$\begin{aligned}
&x[\![1/\text{🔑}_{\mathcal{L}}, 2/\text{🔑}_{\mathcal{K}}]\!][i/x][\reflectbox{λ} i./\text{🔒}_{\mathcal{L}}][\reflectbox{λ} j./\text{🔒}_{\mathcal{K}}] \\
\equiv\ & i[\![1/\text{🔑}_{\mathcal{L}}, 2/\text{🔑}_{\mathcal{K}}]\!][\reflectbox{λ} i./\text{🔒}_{\mathcal{L}}][\reflectbox{λ} j./\text{🔒}_{\mathcal{K}}] \\
\equiv\ & i[1/i][2/\text{🔑}_{\mathcal{K}}][\reflectbox{λ} j./\text{🔒}_{\mathcal{K}}] \\
\equiv\ & 1[2/\text{🔑}_{\mathcal{K}}][\reflectbox{λ} j./\text{🔒}_{\mathcal{K}}] \\
\equiv\ & 1[2/j] \equiv 1
\end{aligned}$$

$$\begin{aligned}
&x[\![1/\text{🔑}_{\mathcal{L}}, 2/\text{🔑}_{\mathcal{K}}]\!][j/x][\reflectbox{λ} i./\text{🔒}_{\mathcal{L}}][\reflectbox{λ} j./\text{🔒}_{\mathcal{K}}] \\
\equiv\ & j[\![1/\text{🔑}_{\mathcal{L}}, 2/\text{🔑}_{\mathcal{K}}]\!][\reflectbox{λ} i./\text{🔒}_{\mathcal{L}}][\reflectbox{λ} j./\text{🔒}_{\mathcal{K}}] \\
\equiv\ & j[1/i][2/\text{🔑}_{\mathcal{K}}][\reflectbox{λ} j./\text{🔒}_{\mathcal{K}}] \\
\equiv\ & j[2/j] \equiv 2
\end{aligned}$$

$$\frac{\Gamma, \text{🔒} \vdash b : B}{\Gamma \vdash \text{🔒}.b : \sqrt{B}} \qquad \frac{\Gamma, i : \mathbb{T} \vdash r : \sqrt{B}}{\Gamma \vdash r(\curlyvee i.) : B[\curlyvee i./\text{🔒}]}$$

Definition

For closed A, define $\mathsf{e} : \sqrt{A} \to A$ by

$$\mathsf{e}(r) :\equiv r(\curlyvee i.)$$

Compare:

$$\mathsf{const} : A \to (C \to A)$$
$$\mathsf{const}(a) :\equiv \lambda c.a$$

Functoriality

$$\frac{\Gamma, \text{🔒} \vdash b : B}{\Gamma \vdash \text{🔒}.b : \sqrt{B}} \qquad \frac{\Gamma, i : \mathbb{T} \vdash r : \sqrt{B}}{\Gamma \vdash r(\curlyvee i.) : B[\curlyvee i./\text{🔒}]}$$

Definition

For closed $f : A \to B$, define $\sqrt{f} : \sqrt{A} \to \sqrt{B}$ by

$$(\sqrt{f})(r) :\equiv \text{🔒}.f(r[i/\text{🔑}](\curlyvee i.))$$

Compare:

$$\begin{aligned} f \circ - &: (C \to A) \to (C \to B) \\ (f \circ -)(r) &:\equiv \lambda c.f(r(c)) \end{aligned}$$

Functoriality

$$\frac{\Gamma, \text{🔒} \vdash b : B}{\Gamma \vdash \text{🔒}.b : \sqrt{B}} \qquad \frac{\Gamma, i : \mathbb{T} \vdash r : \sqrt{B}}{\Gamma \vdash r(⅄i.) : B[⅄i./\text{🔒}]}$$

Definition

For closed $f : A \to B$, define $\sqrt{f} : \sqrt{A} \to \sqrt{B}$ by

$$(\sqrt{f})(r) :\equiv \text{🔒}.f(r[i/\text{🔑}](⅄i.))$$

Start with $r : \sqrt{A}$. To produce $\sqrt{B}$ we need a B after locking our assumptions. There is a function $f : A \to B$ available, so we just need an A. We cannot use e on $r : \sqrt{A}$, because r is locked. We could unlock r as $r[\![i/\text{🔑}]\!] : \sqrt{A}$ if we had an additional assumption $i : \mathbb{T}$. Because we are eliminating $\sqrt{}$, we amazingly do have this assumption. So $(r[\![i/\text{🔑}]\!])(⅄i.) : A$, and we can apply f.

Proposition

For closed types A and B,

$$\mathsf{unsplit} : (\mathbb{T} \to A + B) \to (\mathbb{T} \to A) + (\mathbb{T} \to B)$$

Proof.

$$\mathsf{unsplit}(f) :\equiv \mathsf{case}_{+}(f(i), a.🔒.\mathsf{inl}(\lambda t.a[\![t/🔑]\!]), b.🔒.\mathsf{inr}(\lambda t.b[\![t/🔑]\!]))(\curlyvee i.)$$

□

Proposition

For closed types A, B and P,

$$\mathsf{higherind} : \surd((\mathbb{T} \to A) \to P) \times \surd((\mathbb{T} \to B) \to P) \to (\mathbb{T} \to A + B) \to P$$

Proof.

$$\mathsf{higherind}(g, h, f) :\equiv \mathsf{case}_+(f(i), a.\text{🔒}.g[\![j/\text{🔑}]\!](\curlyvee j.)(\lambda t.a[\![t/\text{🔑}]\!]), b.\text{🔒}.h[\![j/\text{🔑}]\!](\curlyvee j.)(\lambda t.b[\![t/\text{🔑}]\!]))(\curlyvee i.)$$

□

Fix a "notion of composition structure" $\mathsf{C} : (\mathbb{I} \to \mathsf{Set}) \to \mathsf{Set}$.

$$
\begin{aligned}
&\mathsf{isFib} : \textstyle\prod_{(\Gamma:\mathsf{Set})}\prod_{(A:\Gamma\to\mathsf{Set})}\mathsf{Set} \\
&\mathsf{isFib}(\Gamma)(A) :\equiv \textstyle\prod_{(p:\mathbb{I}\to\Gamma)}\mathsf{C}(A \circ p) \\
&\mathsf{Fib} : \mathsf{Set} \to \mathsf{Set} \\
&\mathsf{Fib}(\Gamma) :\equiv \textstyle\sum_{(A:\Gamma\to\mathsf{Set})} \mathsf{isFib}(\Gamma)(A)
\end{aligned}
$$

The [LOPS18] construction of a universe classifying (crisp) fibrations is:

$$
\begin{array}{ccc}
\mathsf{U} & \longrightarrow & \sqrt{\textstyle\sum_{(A:\mathsf{Set})} A} \\
\big\downarrow & \lrcorner & \big\downarrow \sqrt{\mathsf{pr}_1} \\
\mathsf{Set} & \xrightarrow[\mathsf{C}^{\vee}]{} & \sqrt{\mathsf{Set}}
\end{array}
$$

This works out to:

$$\mathsf{U} \equiv \textstyle\sum_{(X:\mathsf{Set})}\sqrt{\mathsf{C}(\lambda j.X[\![j/\text{🔑}]\!])}$$

SDG

We can remove some of the crispness restrictions in David's work.

And try his idea for bundles *with connection*:

$$
\begin{aligned}
B_{\nabla}G &:\equiv (V : BG) \times \Lambda^1(T_{\mathsf{id}}\mathrm{Aut}(V)) \\
&\equiv (V : BG) \times \Lambda^1((\varepsilon : D) \to \mathrm{Aut}(V[\varepsilon/\text{🔑}]) \times \dots)
\end{aligned}
$$

```
data Closure = Closure Env Tm
data RootClosure = RootClosure Env Tm
data Val
  = VPi Closure
  | VLam Closure
  ...
  | VTiny
  | VRoot LockClosure
  | VRootIntro LockClosure

data BindTiny a = BindTiny Lvl a
data Neutral
  = NApp Neutral Val
  ...
  | NVar Lvl [Val]
  | NRootElim (BindTiny Neutral)
```

```
data Env =
  EnvEmpty
  | EnvVal Val Env
  | EnvLock (Val -> Env)

eval env t = case t of
  ...
  App t u      -> apply (eval env t) (eval env u)
  RootElim x t -> freshLvl $ \l ->
    coapply (eval (EnvVal (makeVarLvl l) env) t) l

coapply :: Val -> Lvl -> Val
coapply (VNeutral ne) lvl = VNeutral ...
coapply (VRootIntro (RootClosure cloenv t)) lvl =
  eval (EnvLock (\v -> sub v lvl cloenv)) t
```

Thanks again!