On Autosynthesis

Jacob Neumann
Reykjavik University
4 June 2026
SSTT 2026
Ljubljana, Slovenia

Why's it called 'the groupoid model'?

[HS95] defines the **groupoid model** of (Martin-Löf) type theory.

Q: Why's it called 'the groupoid model'?

A1: *it's made of groupoids*: the contexts are groupoids, substitutions are groupoid morphisms (functors), types are families of groupoids, etc.

A2: *it models a synthetic theory of groupoids*: each type A represents a groupoid

- whose 'objects' are the terms $t : A$ and
- whose 'morphisms' from s to t are terms of type $\mathrm{Id}(s, t)$, with
- identity morphisms given by refl, composition and inverses definable by identity elimination (J);
- failure of UIP means there can be multiple parallel morphisms (A is a groupoid, not necessarily a setoid), but UIP 'one level up' means that each hom-set is indeed a *set*

Autosynthesis

A model made of groupoids

which models a type theory of synthetic groupoids

Q1: Are there any other instances of this?

Q2: What does Q1 mean exactly?

CwFs

Defn. A **category with families** (CwF) consists of

- $\mathsf{Con} : \mathsf{Set}$
- $\mathsf{Sub} : \mathsf{Con} \to \mathsf{Con} \to \mathsf{Set}$
- $\mathsf{Ty} : \mathsf{Con} \to \mathsf{Set}$
- $\mathsf{Tm} : (\Gamma : \mathsf{Con}) \to \mathsf{Ty}\ \Gamma \to \mathsf{Set}$

along with various operations making $(\mathsf{Con}, \mathsf{Sub})$ into a category, Ty into a presheaf, and Tm into a *locally representable* presheaf.

Defn. A CwF morphism F consists of components F_{Con}, F_{Sub}, F_{Ty}, F_{Tm} which commute with the CwF operations in the appropriate way ($(F_{\mathsf{Con}}, F_{\mathsf{Sub}})$ is a functor, F_{Ty} a natural transform, etc.)

Concreteness

Defn. A **concrete category** is a category $\mathbb{C}$ equipped with a faithful functor $\mathbb{C} \to$ **Set** (more generally, a *concrete category over* $\mathbb{D}$ is a category $\mathbb{C}$ equipped with faithful $\mathbb{C} \to \mathbb{D}$).

Prop. A functor F is a monomorphism in the category of categories iff

- Injectivity: if $F(I) = F(J)$ for objects I, J, then $I = J$
- Faithfulness: if $F(f) = F(g)$ for parallel morphisms f, g, then $f = g$.

Concrete CwFs

Prop. A CwF morphism F is a monomorphism in the category of CwFs iff

- Injectivity: if $F_{\mathsf{Con}}(\Gamma) = F_{\mathsf{Con}}(\Delta)$, then $\Gamma = \Delta$
- Faithfulness: if $F_{\mathsf{Sub}}(\sigma) = F_{\mathsf{Sub}}(\tau)$, then $\sigma = \tau$
- Local Injectivity: if $F_{\mathsf{Ty}}(A) = F_{\mathsf{Ty}}(B)$, then $A = B$
- Local Faithfulness: if $F_{\mathsf{Tm}}(s) = F_{\mathsf{Tm}}(t)$ for terms s, t, then $s = t$.

Defn. The **set model** [Dyb95] has sets for contexts, functions for substitutions, families of sets for types, and sections of set families as terms.

Source of concrete CwFs: GAT algebras

Generalized algebraic theories provide a very general framework for articulating ‘notions of structure’

- Original formulation: pre-theories and restriction to well-formed theories [Car86]
- Intrinsically-well-formed signature languages [KKA19]

[KKA19] show that every GAT $\mathfrak{G}$ gives rise to a concrete CwF $\mathscr{C}(\mathfrak{G})$:

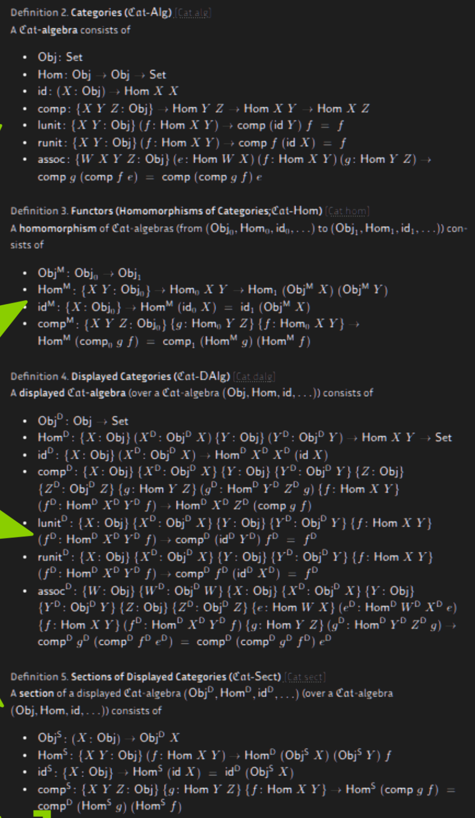
- Contexts are $\mathfrak{G}$-algebras
- Substitutions are $\mathfrak{G}$-algebra homomorphisms
- Types are *displayed* $\mathfrak{G}$-algebras
- Terms are *sections* of displayed $\mathfrak{G}$-algebras

genalg.info

```
def ℭat : GAT := {
    Obj : U,
    Hom : Obj ⇒ Obj ⇒ U,
    id  : (X : Obj) ⇒ Hom X X,
    comp  : {X Y Z : Obj} ⇒
        Hom Y Z ⇒ Hom X Y ⇒ Hom X Z,
    lunit : {X Y : Obj} ⇒ (f : Hom X Y) ⇒
        comp (id Y) f ≡ f,
    runit : {X Y : Obj} ⇒ (f : Hom X Y) ⇒
        comp f (id X) ≡ f,
    assoc : {W X Y Z : Obj} ⇒ (e : Hom W X) ⇒
        (f : Hom X Y) ⇒ (g : Hom Y Z) ⇒
        comp g (comp f e) ≡ comp (comp g f) e
}
```

```
◇
 ▷ U
 ▷ Π 0 (Π 1 U)
 ▷ Π 1 (El (1 @ 0 @ 0))
 ▷ Π 2 (Π 3 (Π 4 (Π (4 @ 1 @ 0) (Π (5 @ 3 @ 2) (El (6 @ 4 @ 2))))))
 ▷ Π 3 (Π 4 (Π (4 @ 1 @ 0) (Eq  (3 @ 2 @ 1 @ 1 @ (4 @ 1) @ 0)  0)))
 ▷ Π 4 (Π 5 (Π (5 @ 1 @ 0) (Eq  (4 @ 2 @ 2 @ 1 @ 0 @ (5 @ 2))  0)))
 ▷ Π 5 (Π 6 (Π 7 (Π 8 (Π (8 @ 3 @ 2) (Π (9 @ 3 @ 2) (Π (10 @ 3 @ 2) (Eq
  (9 @ 6 @ 4 @ 3 @ 0 @ (9 @ 6 @ 5 @ 4 @ 1 @ 2))
  (9 @ 6 @ 5 @ 3 @ (9 @ 5 @ 4 @ 3 @ 0 @ 1) @ 2)))))))))
```

elab



Definition 2. **Categories ($\mathfrak{Cat}$-Alg)** [Cat.alg]

A **$\mathfrak{Cat}$-algebra** consists of

- Obj : Set
- Hom : Obj → Obj → Set
- id : $(X : \text{Obj}) \to \text{Hom}\ X\ X$
- comp : $\{X\ Y\ Z : \text{Obj}\} \to \text{Hom}\ Y\ Z \to \text{Hom}\ X\ Y \to \text{Hom}\ X\ Z$
- lunit : $\{X\ Y : \text{Obj}\}\ (f : \text{Hom}\ X\ Y) \to \text{comp}\ (\text{id}\ Y)\ f\ =\ f$
- runit : $\{X\ Y : \text{Obj}\}\ (f : \text{Hom}\ X\ Y) \to \text{comp}\ f\ (\text{id}\ X)\ =\ f$
- assoc : $\{W\ X\ Y\ Z : \text{Obj}\}\ (e : \text{Hom}\ W\ X)\ (f : \text{Hom}\ X\ Y)\ (g : \text{Hom}\ Y\ Z) \to \text{comp}\ g\ (\text{comp}\ f\ e)\ =\ \text{comp}\ (\text{comp}\ g\ f)\ e$

Definition 3. **Functors (Homomorphisms of Categories;$\mathfrak{Cat}$-Hom)** [Cat.hom]

A **homomorphism** of $\mathfrak{Cat}$-algebras (from $(\text{Obj}_0, \text{Hom}_0, \text{id}_0, \ldots)$ to $(\text{Obj}_1, \text{Hom}_1, \text{id}_1, \ldots)$) consists of

- $\text{Obj}^{\text{M}} : \text{Obj}_0 \to \text{Obj}_1$
- $\text{Hom}^{\text{M}} : \{X\ Y : \text{Obj}_0\} \to \text{Hom}_0\ X\ Y \to \text{Hom}_1\ (\text{Obj}^{\text{M}}\ X)\ (\text{Obj}^{\text{M}}\ Y)$
- $\text{id}^{\text{M}} : \{X : \text{Obj}_0\} \to \text{Hom}^{\text{M}}\ (\text{id}_0\ X)\ =\ \text{id}_1\ (\text{Obj}^{\text{M}}\ X)$
- $\text{comp}^{\text{M}} : \{X\ Y\ Z : \text{Obj}_0\}\ \{g : \text{Hom}_0\ Y\ Z\}\ \{f : \text{Hom}_0\ X\ Y\} \to \text{Hom}^{\text{M}}\ (\text{comp}_0\ g\ f)\ =\ \text{comp}_1\ (\text{Hom}^{\text{M}}\ g)\ (\text{Hom}^{\text{M}}\ f)$

Definition 4. **Displayed Categories ($\mathfrak{Cat}$-DAlg)** [Cat.dalg]

A **displayed $\mathfrak{Cat}$-algebra** (over a $\mathfrak{Cat}$-algebra (Obj, Hom, id, . . .)) consists of

- $\text{Obj}^{\text{D}} : \text{Obj} \to \text{Set}$
- $\text{Hom}^{\text{D}} : \{X : \text{Obj}\}\ (X^{\text{D}} : \text{Obj}^{\text{D}}\ X)\ \{Y : \text{Obj}\}\ (Y^{\text{D}} : \text{Obj}^{\text{D}}\ Y) \to \text{Hom}\ X\ Y \to \text{Set}$
- $\text{id}^{\text{D}} : \{X : \text{Obj}\}\ (X^{\text{D}} : \text{Obj}^{\text{D}}\ X) \to \text{Hom}^{\text{D}}\ X^{\text{D}}\ X^{\text{D}}\ (\text{id}\ X)$
- $\text{comp}^{\text{D}} : \{X : \text{Obj}\}\ \{X^{\text{D}} : \text{Obj}^{\text{D}}\ X\}\ \{Y : \text{Obj}\}\ \{Y^{\text{D}} : \text{Obj}^{\text{D}}\ Y\}\ \{Z : \text{Obj}\}\ \{Z^{\text{D}} : \text{Obj}^{\text{D}}\ Z\}\ \{g : \text{Hom}\ Y\ Z\}\ (g^{\text{D}} : \text{Hom}^{\text{D}}\ Y^{\text{D}}\ Z^{\text{D}}\ g)\ \{f : \text{Hom}\ X\ Y\}\ (f^{\text{D}} : \text{Hom}^{\text{D}}\ X^{\text{D}}\ Y^{\text{D}}\ f) \to \text{Hom}^{\text{D}}\ X^{\text{D}}\ Z^{\text{D}}\ (\text{comp}\ g\ f)$
- $\text{lunit}^{\text{D}} : \{X : \text{Obj}\}\ \{X^{\text{D}} : \text{Obj}^{\text{D}}\ X\}\ \{Y : \text{Obj}\}\ \{Y^{\text{D}} : \text{Obj}^{\text{D}}\ Y\}\ \{f : \text{Hom}\ X\ Y\}\ (f^{\text{D}} : \text{Hom}^{\text{D}}\ X^{\text{D}}\ Y^{\text{D}}\ f) \to \text{comp}^{\text{D}}\ (\text{id}^{\text{D}}\ Y^{\text{D}})\ f^{\text{D}}\ =\ f^{\text{D}}$
- $\text{runit}^{\text{D}} : \{X : \text{Obj}\}\ \{X^{\text{D}} : \text{Obj}^{\text{D}}\ X\}\ \{Y : \text{Obj}\}\ \{Y^{\text{D}} : \text{Obj}^{\text{D}}\ Y\}\ \{f : \text{Hom}\ X\ Y\}\ (f^{\text{D}} : \text{Hom}^{\text{D}}\ X^{\text{D}}\ Y^{\text{D}}\ f) \to \text{comp}^{\text{D}}\ f^{\text{D}}\ (\text{id}^{\text{D}}\ X^{\text{D}})\ =\ f^{\text{D}}$
- $\text{assoc}^{\text{D}} : \{W : \text{Obj}\}\ \{W^{\text{D}} : \text{Obj}^{\text{D}}\ W\}\ \{X : \text{Obj}\}\ \{X^{\text{D}} : \text{Obj}^{\text{D}}\ X\}\ \{Y : \text{Obj}\}\ \{Y^{\text{D}} : \text{Obj}^{\text{D}}\ Y\}\ \{Z : \text{Obj}\}\ \{Z^{\text{D}} : \text{Obj}^{\text{D}}\ Z\}\ \{e : \text{Hom}\ W\ X\}\ (e^{\text{D}} : \text{Hom}^{\text{D}}\ W^{\text{D}}\ X^{\text{D}}\ e)\ \{f : \text{Hom}\ X\ Y\}\ (f^{\text{D}} : \text{Hom}^{\text{D}}\ X^{\text{D}}\ Y^{\text{D}}\ f)\ \{g : \text{Hom}\ Y\ Z\}\ (g^{\text{D}} : \text{Hom}^{\text{D}}\ Y^{\text{D}}\ Z^{\text{D}}\ g) \to \text{comp}^{\text{D}}\ g^{\text{D}}\ (\text{comp}^{\text{D}}\ f^{\text{D}}\ e^{\text{D}})\ =\ \text{comp}^{\text{D}}\ (\text{comp}^{\text{D}}\ g^{\text{D}}\ f^{\text{D}})\ e^{\text{D}}$

Definition 5. **Sections of Displayed Categories ($\mathfrak{Cat}$-Sect)** [Cat.sect]

A **section** of a displayed $\mathfrak{Cat}$-algebra $(\text{Obj}^{\text{D}}, \text{Hom}^{\text{D}}, \text{id}^{\text{D}}, \ldots)$ (over a $\mathfrak{Cat}$-algebra (Obj, Hom, id, . . .)) consists of

- $\text{Obj}^{\text{S}} : (X : \text{Obj}) \to \text{Obj}^{\text{D}}\ X$
- $\text{Hom}^{\text{S}} : \{X\ Y : \text{Obj}\}\ (f : \text{Hom}\ X\ Y) \to \text{Hom}^{\text{D}}\ (\text{Obj}^{\text{S}}\ X)\ (\text{Obj}^{\text{S}}\ Y)\ f$
- $\text{id}^{\text{S}} : \{X : \text{Obj}\} \to \text{Hom}^{\text{S}}\ (\text{id}\ X)\ =\ \text{id}^{\text{D}}\ (\text{Obj}^{\text{S}}\ X)$
- $\text{comp}^{\text{S}} : \{X\ Y\ Z : \text{Obj}\}\ \{g : \text{Hom}\ Y\ Z\}\ \{f : \text{Hom}\ X\ Y\} \to \text{Hom}^{\text{S}}\ (\text{comp}\ g\ f)\ =\ \text{comp}^{\text{D}}\ (\text{Hom}^{\text{S}}\ g)\ (\text{Hom}^{\text{S}}\ f)$

[KKA19, Appendix A]

GAT methodology for designing type theories

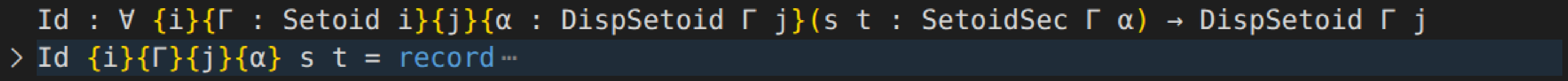
Key Fact Every GAT (indeed, every QIIT signature) has an initial algebra

Method Specify new type theories as GAT (or QIIT) extensions of $\mathfrak{CwF}$, get initial syntax models automatically

Ex. CwFs with identity types:

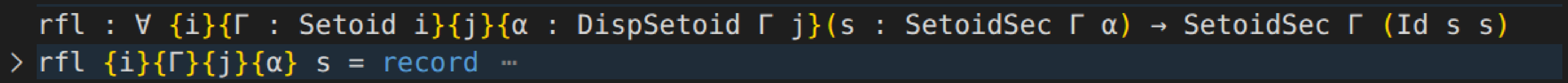
$$
\begin{aligned}
\mathfrak{CwF} &\rhd \Pi\ 25\ (\Pi\ (16 @ 0)\ (\Pi\ (13 @ 1 @ 0)\ (\Pi\ (14 @ 2 @ 1)\ (\mathrm{El}\ (19 @ 3)))))\\
&\rhd \Pi\ 26\ (\Pi\ (17 @ 0)\ (\Pi\ (14 @ 1 @ 0)\ (\mathrm{El}\ (15 @ 2 @ (3 @ 2 @ 1 @ 0 @ 0))\\
&\rhd \cdots
\end{aligned}
$$

Ex. $\mathscr{C}(\mathfrak{Setoid})$ supports identity types



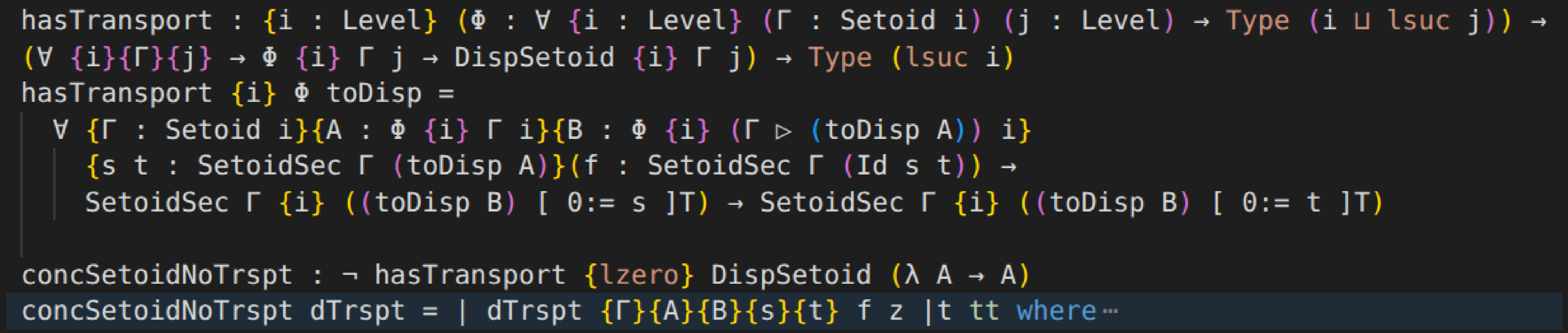
```
  Id : ∀ {i}{Γ : Setoid i}{j}{α : DispSetoid Γ j}(s t : SetoidSec Γ α) → DispSetoid Γ j
> Id {i}{Γ}{j}{α} s t = record…
```

with reflexivity



```
  rfl : ∀ {i}{Γ : Setoid i}{j}{α : DispSetoid Γ j}(s : SetoidSec Γ α) → SetoidSec Γ (Id s s)
> rfl {i}{Γ}{j}{α} s = record …
```

but not transport



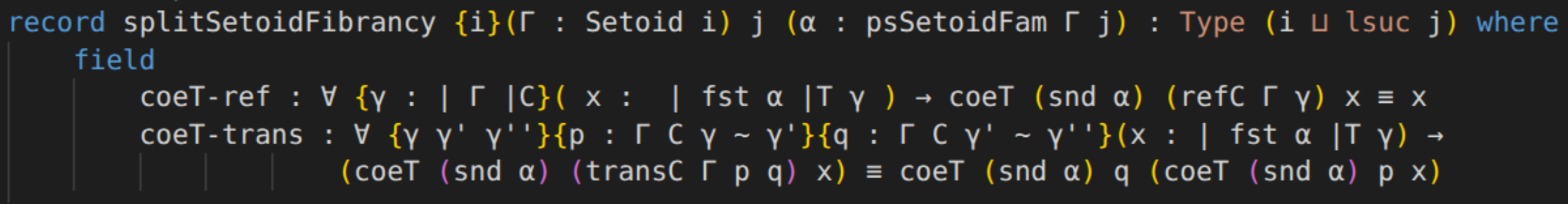
```
hasTransport : {i : Level} (Φ : ∀ {i : Level} (Γ : Setoid i) (j : Level) → Type (i ⊔ lsuc j)) →
(∀ {i}{Γ}{j} → Φ {i} Γ j → DispSetoid {i} Γ j) → Type (lsuc i)
hasTransport {i} Φ toDisp =
  ∀ {Γ : Setoid i}{A : Φ {i} Γ i}{B : Φ {i} (Γ ▷ (toDisp A)) i}
    {s t : SetoidSec Γ (toDisp A)}(f : SetoidSec Γ (Id s t)) →
    SetoidSec Γ {i} ((toDisp B) [ 0:= s ]T) → SetoidSec Γ {i} ((toDisp B) [ 0:= t ]T)

concSetoidNoTrspt : ¬ hasTransport {lzero} DispSetoid (λ A → A)
concSetoidNoTrspt dTrspt = | dTrspt {Γ}{A}{B}{s}{t} f z |t tt where…
```

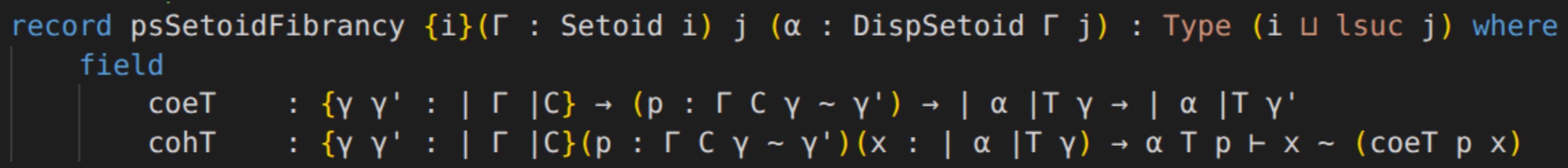
Solution: **Fibrancy!**

$$\begin{tikzcd}
\mathscr{C}_{\mathrm{split}}(\mathfrak{Setoid}) \arrow[d] \\
\mathscr{C}_{\mathrm{ps}}(\mathfrak{Setoid}) \arrow[d] \\
\mathscr{C}(\mathfrak{Setoid})
\end{tikzcd}$$

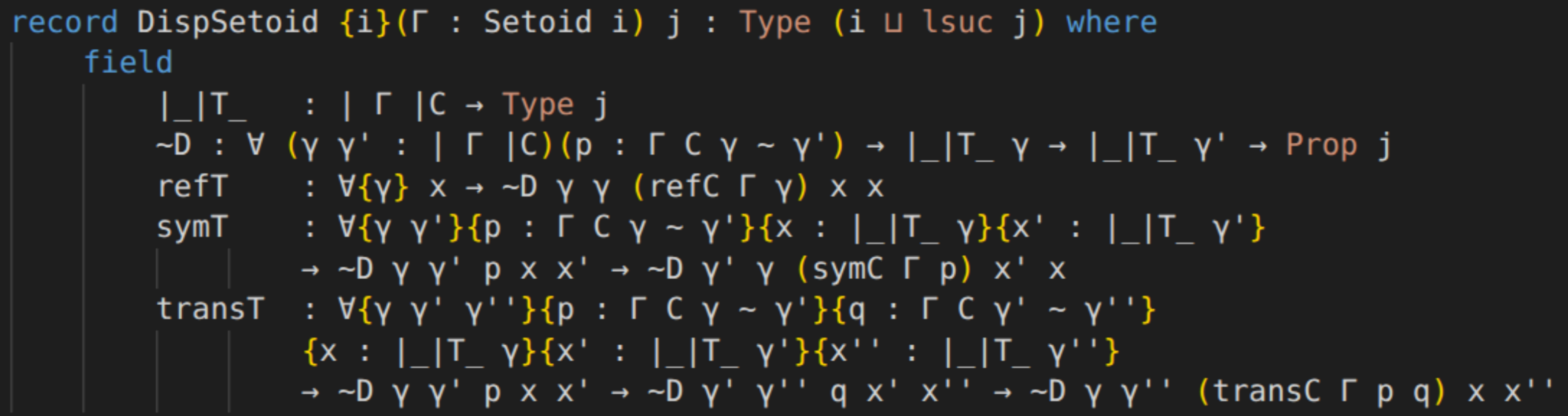
"the setoid model"



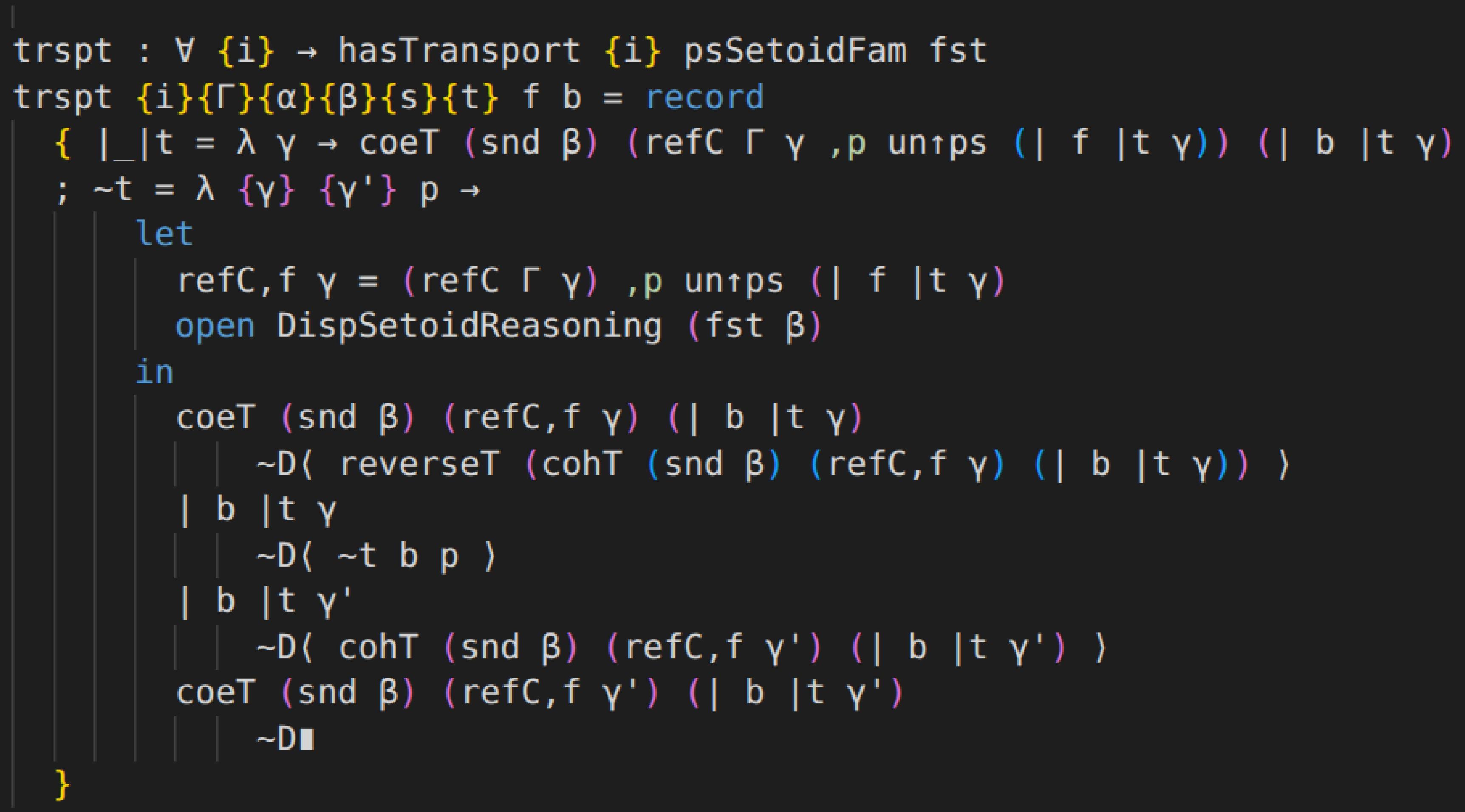
```
record splitSetoidFibrancy {i}(Γ : Setoid i) j (α : psSetoidFam Γ j) : Type (i ⊔ lsuc j) where
    field
        coeT-ref : ∀ {γ : | Γ |C}( x :  | fst α |T γ ) → coeT (snd α) (refC Γ γ) x ≡ x
        coeT-trans : ∀ {γ γ' γ''}{p : Γ C γ ~ γ'}{q : Γ C γ' ~ γ''}(x : | fst α |T γ) →
                    (coeT (snd α) (transC Γ p q) x) ≡ coeT (snd α) q (coeT (snd α) p x)
```



```
record psSetoidFibrancy {i}(Γ : Setoid i) j (α : DispSetoid Γ j) : Type (i ⊔ lsuc j) where
    field
        coeT    : {γ γ' : | Γ |C} → (p : Γ C γ ~ γ') → | α |T γ → | α |T γ'
        cohT    : {γ γ' : | Γ |C}(p : Γ C γ ~ γ')(x : | α |T γ) → α T p ⊢ x ~ (coeT p x)
```



```
record DispSetoid {i}(Γ : Setoid i) j : Type (i ⊔ lsuc j) where
    field
        |_|T_   : | Γ |C → Type j
        ~D : ∀ (γ γ' : | Γ |C)(p : Γ C γ ~ γ') → |_|T_ γ → |_|T_ γ' → Prop j
        refT    : ∀{γ} x → ~D γ γ (refC Γ γ) x x
        symT    : ∀{γ γ'}{p : Γ C γ ~ γ'}{x : |_|T_ γ}{x' : |_|T_ γ'}
                → ~D γ γ' p x x' → ~D γ' γ (symC Γ p) x' x
        transT  : ∀{γ γ' γ''}{p : Γ C γ ~ γ'}{q : Γ C γ' ~ γ''}
                {x : |_|T_ γ}{x' : |_|T_ γ'}{x'' : |_|T_ γ''}
                → ~D γ γ' p x x' → ~D γ' γ'' q x' x'' → ~D γ γ'' (transC Γ p q) x x''
```



```
trspt : ∀ {i} → hasTransport {i} psSetoidFam fst
trspt {i}{Γ}{α}{β}{s}{t} f b = record
  { |_|t = λ γ → coeT (snd β) (refC Γ γ ,p un↑ps (| f |t γ)) (| b |t γ)
  ; ~t = λ {γ} {γ'} p →
      let
        refC,f γ = (refC Γ γ) ,p un↑ps (| f |t γ)
        open DispSetoidReasoning (fst β)
      in
        coeT (snd β) (refC,f γ) (| b |t γ)
            ~D⟨ reverseT (cohT (snd β) (refC,f γ) (| b |t γ)) ⟩
        | b |t γ
            ~D⟨ ~t b p ⟩
        | b |t γ'
            ~D⟨ cohT (snd β) (refC,f γ') (| b |t γ') ⟩
        coeT (snd β) (refC,f γ') (| b |t γ')
            ~D∎
  }
```

Fibrancy unlocks the door to auto-synthesis

- $\mathfrak{Setoid}$
  - Fibrancy gives us transport
  - With transport, we can prove symmetry and transitivity of Id, i.e that the types of the setoid model are synthetic setoids
- $\mathfrak{Grpd}$
  - The groupoid model has split fibrant displayed groupoids as types
  - Fibrancy gives us transport
  - With transport, we can prove symmetry and transitivity of Id, i.e that the types of the groupoid model are synthetic groupoids

Directed analogue

Central question of [Neu25]: how does $\mathfrak{Cat}$ (and $\mathfrak{PreOrd}$) auto-synthesize?

- Polarity
  - Two notions of types, Ty and Ty^{-} (opfibrations and fibrations, respectively), with their own context extension operators
  - Express Ty^{-} in terms of the opposite category operation (on contexts) and its fiberwise version (on types)
  - Hom-type and Π-type formers come with polarity annotations (polarities prevent directed J-rule from proving symmetry)

Directed analogue

Central question of [Neu25]: how does $\mathfrak{Cat}$ (and $\mathfrak{PreOrd}$) auto-synthesize?

- Neutralizing the polarity
  - A calculus for working in groupoid contexts, where we can coerce between the polarities
  - Make polarized Π-types and Hom-types actually useful
  - Characterize the Hom-types of Σ-types
- Synthetic category theory
  - Any type we can write down carries the structure of a synthetic category (composition constructible by J)
  - Functions are synthetic functors (morphism part and functoriality automatic)
  - Directed univalent universe of sets
  - Universal mapping properties (products, exponentials, adjoints) as principles of induction

The general situation

Question For which GATs $\mathfrak{G}$ can we find

- a type theory, i.e. an extension $\mathfrak{GCwF}$ of $\mathfrak{CwF}$, and
- a $\mathfrak{GCwF}$-algebra $\mathscr{M}(\mathfrak{G})$ ("the $\mathfrak{G}$-algebra model")

such that

- $\mathscr{M}(\mathfrak{G})$ is a concrete CwF over $\mathscr{C}(\mathfrak{G})$ ("$\mathscr{M}(\mathfrak{G})$ is made of $\mathfrak{G}$-algebras")
- the (closed) types of the initial $\mathfrak{GCwF}$ are synthetic $\mathfrak{G}$-algebras ("$\mathscr{M}(\mathfrak{G})$ models a type theory of synthetic $\mathfrak{G}$-algebras")?

Some thoughts and questions

- Besides $\mathfrak{Set}$ (which trivially auto-synthesizes), the lower bound seems to be $\mathfrak{Cat}$
  - Cosmology: view the collection of $\mathfrak{G}$-algebras itself as a $\mathfrak{G}$-algebra, to define $\mathsf{Ty}(\Gamma) = \Gamma \to \mathfrak{G}\text{-}\mathsf{Alg}$
  - Fibrancy seems to play a key role. Does fibrancy make sense for non-categorical structures?
- Beyond $\mathfrak{Cat}$
  - 2-category model? n-category model?
  - The $\mathfrak{CwF}$ model? Types in Γ are CwF morphisms from Γ to $\mathscr{C}(\mathfrak{CwF})$? Fibrant displayed CwFs? A type theory for synthetic CwFs?
  - The GAT signature language is itself the initial algebra of a GAT? Does *it* auto-synthesize?

[Car86] John Cartmell.
Generalised algebraic theories and contextual categories.
Annals of pure and applied logic, 32:209–243, 1986.

[Dyb95] Peter Dybjer.
Internal type theory.
In *International Workshop on Types for Proofs and Programs*, pages 120–134. Springer, 1995.

[HS95] Martin Hofmann and Thomas Streicher.
The groupoid interpretation of type theory.
Twenty-five years of constructive type theory (Venice, 1995), 36:83–111, 1995.

[KKA19] Ambrus Kaposi, András Kovács, and Thorsten Altenkirch.
Constructing quotient inductive-inductive types.
Proc. ACM Program. Lang., 3(POPL), jan 2019.

[Neu25] Jacob Neumann.
A Generalized Algebraic Theory of Directed Equality.
PhD thesis, University of Nottingham, 2025.

- The Generalized Algebra website:
  genalg.info
- More about my PhD project:
  jacobneu.phd

Thank you!