What is an inference rule?

André Hirschowitz, Tom Hirschowitz, and Ambroise Lafont

LIMD seminar, Feb 15, 2024

Type theory

- Invented around 1910 by Russell and Whitehead.
- Way of avoiding paradoxes in naive set theory.
- Use today:
  - in programming language theory: programming languages described as specific type theories,
  - in mechanised mathematics: most mainstream proof assistants use some type theory as their foundation.
- In mechanised mathematics, mostly dependent type theory, our focus today.

Dependent type theory

Feature of proof assistants based on dependent type theory:

dependent sum and product types.

- Dependent sum $\coprod_{x:A} B(x) \approx \exists x : A, B$.
    - Example: $\coprod_{n:\mathbb{N}} \mathrm{vec}(n)$, pairs of some $n \in \mathbb{N}$ and a vector of length n.
    - Example: $\coprod_{x:X} \mathrm{paths}(x, x)$, pairs of a point x in some space X and a loop around x.
- Dependent product $\prod_{x:A} B(x) \approx \forall x : A, B$.
    - Example: $\prod_{x:X} T(x)$, sections of a (tangent?) bundle T.

Example: the axiom of choice in type theory

For any relation $R \subseteq A \times B$:

$$\prod_{a:A} \coprod_{b:B} R(a, b) \to \coprod_{f:A \to B} \prod_{a:A} R(a, f(a)).$$

Current practice

- Define type theories by inference rules.
- Rarely need to say anything about substitution, beyond "such variable is binding in such term".

Example: dependent application

$$\frac{\Gamma \vdash A : \mathbf{type} \qquad \Gamma, a : A \vdash B : \mathbf{type} \qquad \Gamma \vdash M : \prod_{a:A} B \qquad \Gamma \vdash N : A}{\Gamma \vdash M\ N : B[a \mapsto N]}$$

What does such an inference rule really mean, mathematically?

Interpreting inference rules

$$\frac{\Gamma \vdash A : \mathbf{type} \qquad \Gamma, a : A \vdash B : \mathbf{type} \qquad \Gamma \vdash M : \prod_{a:A} B \qquad \Gamma \vdash N : A}{\Gamma \vdash M\ N : B[a \mapsto N]}$$

- Extrinsic (aka old-school) approach.
- Nearly algebraic approaches.
- Fancy approaches.

Extrinsic approach

Two layers:

- Untyped version terms and types.
  Example:

$$\frac{A : \mathbf{type} \qquad B : \mathbf{type}}{\prod_{a:A} B : \mathbf{type}} \qquad\qquad \frac{M : \mathbf{term} \qquad N : \mathbf{term}}{M\ N : \mathbf{term}}$$

- Typing rules as a relation on types and terms.

Issues

- Would like only well-typed terms to exist.
- Unclear notion of model.

Nearly algebraic approaches

A bunch of roughly equivalent formalisms:

- Finite limit sketches (Ehresmann, 1968).
- Essentially algebraic theories (Freyd, 1972).
- Generalised algebraic theories (Cartmell, 1978).
- Inside type theory: fancy inductive types (e.g., inductive-recursive types).

Assessment

- General-purpose.
- Substitution must be defined by hand.

Fancy approaches

Uemura (2019), Gratzer and Sterling (2020), Coraglia and Di Liberti (2021).

- Manage to infer substitution.
- Sophisticated, e.g.,
    - Gratzer and Sterling rely on generalised sketches over the 2-monad of locally cartesian closed categories,
    - Di Liberti and Osmond are currently developing a 2-categorical extension of locally presentable categories for justifying Coraglia and Di Liberti's framework.

A sweet spot?

Our approach:

- Infer substitution.
- Initial-algebra semantics (Goguen, 1974):
  - Define a whole category of models.
  - The desired type theory is the initial one[1].
  - Initiality ≈ recursion principle.

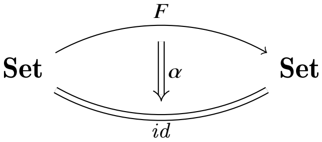
  This is like an implicit definition:
  - we know the desired object exists without constructing it,
  - it has the properties we need.
- Relatively basic category theory: locally presentable categories.

> If we prove that the category of models is locally presentable, then it has an initial object: the desired type theory.

[1] Explain initial objects on the board!

Categorical interlude

Natural transformations, set-based example



- F is a functor: it maps sets to sets and functions to functions;
- α is a natural transformation: for all sets X, we have a map $\alpha_X \colon F(X) \to X$, “naturally” in X.

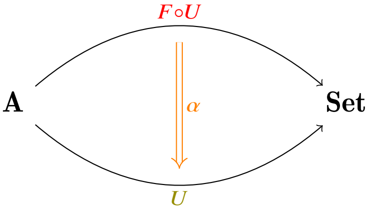
Example: $F(X) = X \times X$

F-algebra structure = binary operation $\alpha_X \colon X^2 \to X$.
Ex: $\alpha_X =$ first projection.

Core infrastructure: algebras as inserters

Starting from $F\colon \mathbf{Set} \to \mathbf{Set}$ as before, consider a category $\mathbf{A}$ with a functor $U\colon \mathbf{A} \to \mathbf{Set}$.

Giving a natural transformation



is equipping each $U(a)$ with F-algebra structure

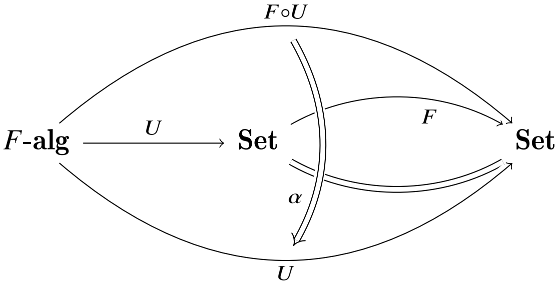
$$F(U(a)) \xrightarrow{\alpha_a} U(a).$$

The category F-**alg** of F-algebras is the universal such $U\colon \mathbf{A} \to \mathbf{Set}$.

Inserters

The category F-**alg** of F-algebras is the universal such $U\colon \mathbf{A} \to \mathbf{Set}$.

This is called an inserter of F and id.



(It “inserts” a 2-cell α between F and id.)

Locally presentable magic

Well-known, not recalled here, please believe me:

- Notions of locally presentable category and accessible functor.
- Notion of continuous functor (preserves limits).

Theorem (Essentially in Adamek and Rosicky, 1994)

For any

- *locally presentable* $\mathbf{C}$ *and* $\mathbf{D}$ *and*
- *accessible* $F, G \colon \mathbf{C} \to \mathbf{D}$,

if G *is continuous, then the inserter* $U \colon \mathbf{A} \to \mathbf{C}$ *of* F *and* G *is accessible and* $\mathbf{A}$ *is locally presentable.*

Example: when $G = id$, F-$\mathbf{alg}$ is locally presentable $\rightsquigarrow$ initial object.

Summary thus far

- Category of algebras as an inserter.
- General result: inserters of locally presentable categories are again locally presentable (roughly).

Next question: how to deal with substitution automatically?

Familial interlude

Well-known equivalence:

- families of sets indexed by some fixed set X,
- pairs of a set Y and a map $Y \to X$.

A well-known axiomatisation of substitution

Contexts form a category $\mathbb{C}$:

- objects are contexts Γ,
- morphisms are assignments $\Gamma \vdash \sigma : \Delta$, i.e.,

$$\Gamma \vdash (M_1, \ldots, M_n) : (x_1 : A_1, \ldots, x_n : A_n)$$

  where, for all i, $\Gamma \vdash M_i : A_i$.

Simply-typed case, but also works with dependent types.
Composition of

$$\Gamma \xrightarrow{\sigma} \Delta \xrightarrow{(M_1,\ldots,M_n)} (x_1 : A_1, \ldots, x_n : A_n)$$

is

$$\Gamma \xrightarrow{(M_1[\sigma],\ldots,M_n[\sigma])} (x_1 : A_1, \ldots, x_n : A_n).$$

A well-known axiomatisation of substitution

- For each context Γ, we have a family $\mathbb{T}(\Gamma)$ given by

$$\mathrm{Terms}(\Gamma) \to \mathrm{Types}(\Gamma).$$

  - A set $\mathrm{Types}(\Gamma)$ of types.
    Example: for $\Gamma = (n : \mathbb{N})$, $\mathrm{vec}\ n$.
  - For each type A, a set $\mathrm{Terms}(\Gamma)_A$ of terms of type A.

- For each assignment $\sigma : \Gamma \to \Delta$, substitution gives maps

$$\begin{array}{cc} \mathrm{Types}(\Delta) \to \mathrm{Types}(\Gamma) & \mathrm{Terms}(\Delta)_A \to \mathrm{Terms}(\Gamma)_{A[\sigma]} \\ A \mapsto A[\sigma] & M \mapsto M[\sigma], \end{array}$$

i.e., a morphism $\mathbb{T}(\sigma) : \mathbb{T}(\Delta) \to \mathbb{T}(\Gamma)$.

Substitution as indexing

In summary, we have a functor

$$\mathbb{T} \colon \mathbb{C}^{op} \to \mathbf{Fam}$$

from contexts to families of sets.

Notation

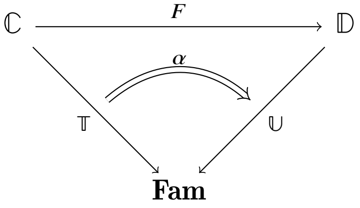
$\mathbb{T} = (\text{Types}, \text{Terms}, \ldots)$, $\mathbb{U} = (\text{Types}', \text{Terms}', \ldots)$, ... hopefully clear from context.

This is how we think of type theories:

- a category $\mathbb{C}$ and
- a functor $\mathbb{C}^{op} \to \mathbf{Fam}$.

Actually, forget the $-^{op}$, just take $\mathbb{C}^{op}$ instead of $\mathbb{C}$ as base category!

Morphism between type theories



- A functor F between categories of contexts.
- For all contexts $\Gamma \in \mathbb{C}$, maps

$$\alpha^0_\Gamma \colon \mathrm{Types}(\Gamma) \to \mathrm{Types}'(F(\Gamma))$$
$$\alpha^1_{\Gamma,A} \colon \mathrm{Terms}(\Gamma)_A \to \mathrm{Terms}'(F(\Gamma))_{\alpha^0_\Gamma(A)}.$$

We obtain a category $\mathbf{Cat}//\mathbf{Fam}$.

A (large) category of type theories

Definition

A type theory is an object $(\mathbb{C}, \mathbb{T}) \in \mathbf{Cat}//\mathbf{Fam}$ with context extension, which I won't explain today unless asked for it.

Type theories form a large category $\mathbf{CwF}$, for (small) categories with families.

A first, naive notion of inference rule

Replacing $\mathbf{Fam}$ with $\mathbf{Set}$, we get a category $\mathbf{Cat}/\!/\mathbf{Set}$.

Definition

A naive inference rule is a pair of functors

$$\mathbf{C} \underset{T}{\overset{S}{\rightrightarrows}} \mathbf{Cat}/\!/\mathbf{Set}$$

satisfying local presentability hypotheses, notably T is continuous.

Example naive inference rule

$$\frac{\Gamma \vdash M : A \qquad \Gamma \vdash N : B}{\Gamma \vdash (M, N) : A \times B}$$

Our functors $S, T \colon \mathbf{CwF}_\times \to \mathbf{Cat}//\mathbf{Set}$ keep the same base category, e.g.,

$$F(\mathbb{C}, \mathbb{T}) = (\mathbb{C}, F' \colon \mathbb{C} \to \mathbf{Set}),$$

with

- $S'(\mathbb{C}, \mathbb{T})(\Gamma) = \coprod_{A,B \in \mathrm{Types}(\Gamma)} \mathrm{Terms}(\Gamma)_A \times \mathrm{Terms}(\Gamma)_B$ and
- $T'(\mathbb{C}, \mathbb{T})(\Gamma) = \coprod_{A,B \in \mathrm{Types}(\Gamma)} \mathrm{Terms}(\Gamma)_{A \times B}$.

Remark

Action on morphisms $\Gamma \to \Delta$ will impose the behaviour of substitution.

Models of a naive inference rule

Definition

Let $R = (S, T)$ be any inference rule. Its category of models is the inserter of S and T.

When S and T keep the same base category, a model is a type theory $(\mathbb{C}, \mathbb{T})$ with a coherent family of morphisms

$$S'(\mathbb{C}, \mathbb{T})(\Gamma) \to T'(\mathbb{C}, \mathbb{T})(\Gamma).$$

Inadequacy of inference rules

Naive inference rules are insufficient, e.g., for dependent application.

$$\frac{\Gamma \vdash A : \mathbf{type} \qquad \Gamma, a : A \vdash B : \mathbf{type} \qquad \Gamma \vdash M : \prod_{a:A} B \qquad \Gamma \vdash N : A}{\Gamma \vdash M\ N : B[a \mapsto N]}$$

Models are type theories $(\mathbb{C}, \mathbb{T})$ with morphisms

$$\coprod_{A,B} \mathrm{Terms}(\Gamma)_{\prod_{a:A} B} \times \mathrm{Terms}(\Gamma)_A \to \coprod_{A,B} \coprod_{N \in \mathrm{Terms}(\Gamma)_A} \mathrm{Terms}(\Gamma)_{B[a \mapsto N]}.$$

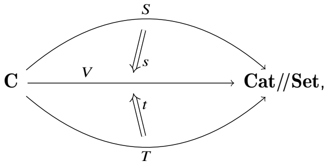
Nothing imposes that A and B remain the same.

Inference rules

In order to express preservation of data between source and target, we refine our definition:

Definition

An inference rule consists of three functors and two natural transformations as in



with T and V continuous.

- In fact, works for any locally presentable category instead of $\mathbf{Cat}/\!/\mathbf{Set}$!
- Enables iteration.

Example: dependent application

$$\frac{\Gamma \vdash A : \mathbf{type} \qquad \Gamma, a : A \vdash B : \mathbf{type} \qquad \Gamma \vdash M : \prod_{a:A} B \qquad \Gamma \vdash N : A}{\Gamma \vdash M\ N : B[a \mapsto N]}$$

- $\mathbf{C} := \mathbf{CwF}_{\prod}$,
- $V'(\mathbb{C}, \mathbb{T})(\Gamma) = \coprod_{A,B} \mathrm{Terms}(\Gamma)_A$,
- $S'(\mathbb{C}, \mathbb{T})(\Gamma) = \coprod_{A,B,N \in V'(\mathbb{C},\mathbb{T})(\Gamma)} \mathrm{Terms}(\Gamma)_{\prod_{a:A} B}$,
- $T'(\mathbb{C}, \mathbb{T})(\Gamma) = \coprod_{A,B,N \in V'(\mathbb{C},\mathbb{T})(\Gamma)} \mathrm{Terms}(\Gamma)_{B[a \mapsto N]}$.

Models

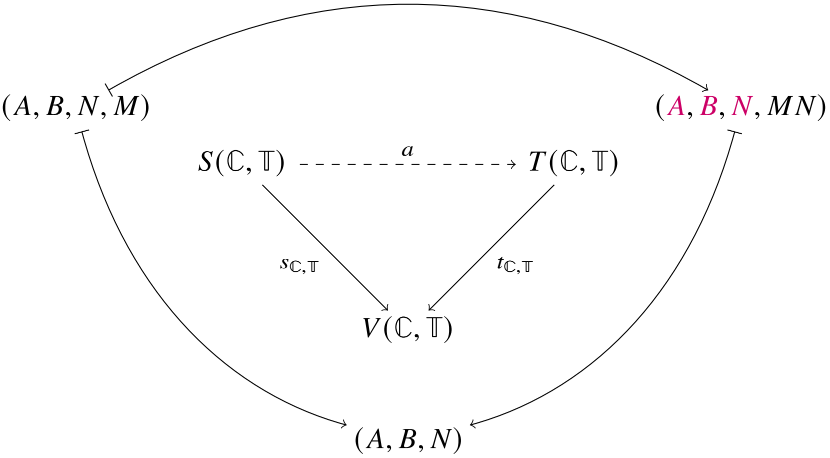
We enforce data preservation in the definition of models:

Definition

A model of $R = (V, S, T, s, t)$ is a type theory $(\mathbb{C}, \mathbb{T})$ with a morphism a making the following diagram commute.

$$\begin{array}{ccc} S(\mathbb{C}, \mathbb{T}) & \overset{a}{\dashrightarrow} & T(\mathbb{C}, \mathbb{T}) \\ & {\scriptstyle s_{\mathbb{C},\mathbb{T}}} \searrow \quad \swarrow {\scriptstyle t_{\mathbb{C},\mathbb{T}}} & \\ & V(\mathbb{C}, \mathbb{T}) & \end{array}$$

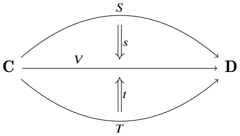
Example: dependent application



Local presentability of models

Proposition

For any



with

- $\mathbf{C}$ and $\mathbf{D}$ locally presentable,
- S, T, and V accessible, and
- T and V continuous,

the category of models is again locally presentable, and the forgetful functor to $\mathbf{C}$ is accessible and creates limits.

Conclusion

- General, rigorous notion of inference rule.
- Proof that models form a locally presentable category.

Not shown, in progress:

- tools to automatically infer that T and V are continuous in practice (cf. technical part.);
- application to quantitative type theory (Atkey, '18).

Thanks for your attention

1. Introduction
2. Exploiting substitution as indexing
3. Naive inference rules
4. Inference rules
5. Conclusion

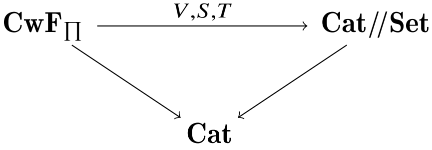
Any questions?

Styles

Our example functors

$$\mathbf{CwF}_{\prod} \to \mathbf{Cat}//\mathbf{Set}$$

preserve the base categoy.



Alternative presentation

Observation: $\mathbf{Cat}/\!/\mathbf{Set} = \oint \mathcal{P}$, where

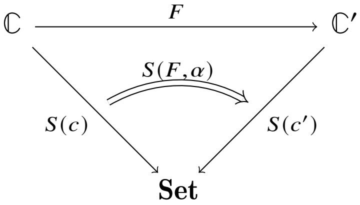
$$\begin{aligned}\mathcal{P} : \mathbf{Cat}^{op} &\to \mathbf{CAT} \\ \mathbb{C} &\mapsto [\mathbb{C}, \mathbf{Set}].\end{aligned}$$

Alternative presentation

Base-preserving functors $\mathbf{CwF}_{\prod} \to \mathbf{Cat}//\mathbf{Set}$ are in 1-1 correspondence with lax global sections S of $\mathcal{P}_U$:

$$\mathbf{CwF}_{\prod}{}^{op} \xrightarrow{U} \mathbf{Cat}^{op} \xrightarrow{\mathcal{P}} \mathbf{CAT}, \text{i.e.,}$$

- for all $c = (\mathbb{C}, X, \prod) \in \mathbf{CwF}_{\prod}$, a functor $S(c) \colon \mathbb{C} \longrightarrow \mathbf{Set}$,
- for all $(F, \alpha) \colon (\mathbb{C}, X, \prod) \to (\mathbb{C}', X', \prod') = c'$, a natural transformation



compatible with composition in $\mathbf{CwF}_{\prod}$.

Consequence

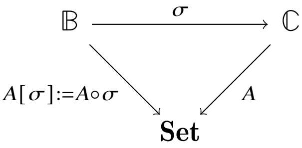
Proving that some base-preserving functor is continuous
$\iff$ proving that the corresponding lax global section of $\mathcal{P}_U$ is continuous.

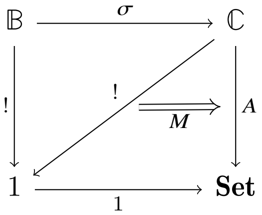
$\rightsquigarrow$ attempt to understand the structure of (continuous) lax global sections.

A CwF of presheaves

Not so well known (variants in Hofmann and Streicher; Harper and Licata; Coraglia and Di Liberti):

- contexts = (small) categories and functors;
- types over $\mathbb{C}$ = functors $\mathbb{C} \to \mathbf{Set}$;
- terms of type $A\colon \mathbb{C} \to \mathbf{Set}$ = global elements $1 \to A$;
- substitution of types and terms = precomposition.

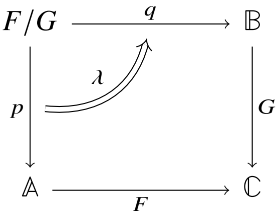




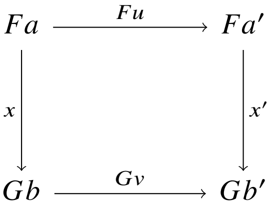
(a global element of $A[\sigma]$)

Categorical interlude: the category of elements as a comma category

Comma category of F and G:



Concretely:



Transfo λ given at $x\colon Fa \to Gb$ by... x itself!

Application: category of elements

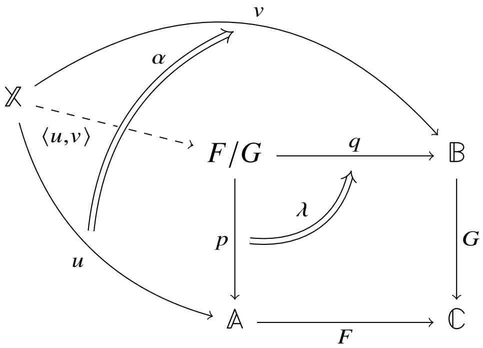
$$\begin{array}{ccc} \{A\} := 1/A & \xrightarrow{\partial_A^1} & \mathbb{C} \\ {\scriptstyle !}\big\downarrow & \partial_A^2 & \big\downarrow{\scriptstyle A} \\ 1 & \xrightarrow[1]{} & \mathbf{Set} \end{array}$$

Concretely:

$$\begin{array}{ccc} 1 & = & 1 \\ {\scriptstyle x}\big\downarrow & & \big\downarrow{\scriptstyle y} \\ A(c) & \xrightarrow[A(f)]{} & A(d) \end{array}$$

Categorical interlude: the category of elements as a comma category

Universal property of the comma category:

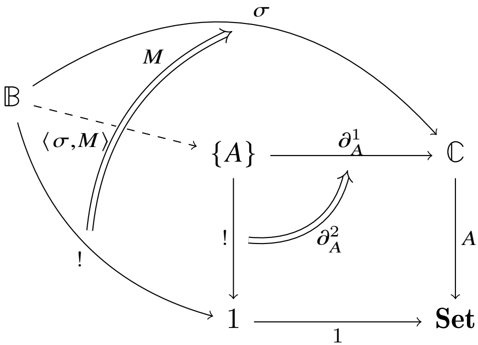


Define $\langle u, v\rangle(x) := \alpha_x : F(u(x)) \to G(v(y))$.

Context extension as category of elements

$$\begin{array}{ccc} \{A\} & \xrightarrow{\partial^1_A} & \mathbb{C} \\ {\scriptstyle !}\downarrow & \partial^2_A : \,! \Rightarrow \partial^1_A & \downarrow{\scriptstyle A} \\ 1 & \xrightarrow[1]{} & \mathbf{Set} \end{array} \qquad \text{cf.} \qquad \begin{array}{ccc} \mathbf{y}_{[0]} & \xrightarrow{A} & \mathbb{T}(\Gamma) \\ {\scriptstyle \mathbf{y}_s}\downarrow & & \downarrow{\scriptstyle \mathbb{T}(\partial^1_A)} \\ \mathbf{y}_{[1]} & \xrightarrow[\partial^2_A]{} & \mathbb{T}(\Gamma.A) \end{array}$$

Universal properties



$$\frac{\Delta \vdash \sigma : \Gamma \qquad \Gamma \vdash A : \mathbf{type} \qquad \Delta \vdash M : A[\sigma]}{\Delta \vdash \langle \sigma, M \rangle : \Gamma.A}$$

By construction $\langle \sigma, M \rangle (b) = M_b \in A(\sigma(b)) = A[\sigma](b)$.