What is an inference rule?

André Hirschowitz, Tom Hirschowitz, and Ambroise Lafont

LHC days, June 5, 2024

Dependent types

Ubiquitous in the community:

- in programming language theory: programming languages described as specific type theories (after Harper),
- in mechanised mathematics: most mainstream proof assistants use some type theory as their foundation.

Current practice to define dependent type theories

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

Although sometimes not obvious.

Example: dependent application

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

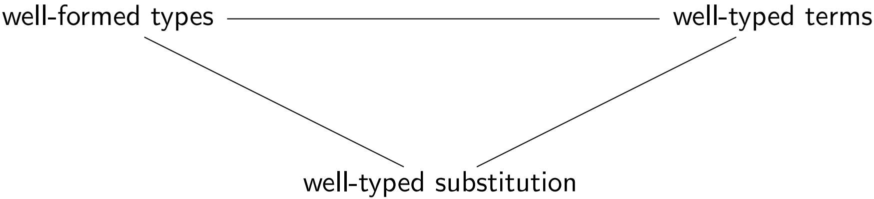
Question

What are the mathematical foundations of inference rules and substitution inference?

Difficulty of intrinsically-typed definition

Circularity!

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



(Not super precise, but you get the idea Johnny?)

This work

Goal

Abstract notion of inference rule with generic

- construction of category of models,
- initial model.

Also, on suitable instances: substitution inference.

Bonus

- Locally presentable category of models $\rightsquigarrow$
  - complete and cocomplete,
  - well-powered and well-copowered,
  - (epi, mono) orthogonal factorisation system(s).
- Forgetful functor
  - is an accessible right adjoint and
  - creates limits.

Related work

- 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.
- No substitution inference on well-typed terms.

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.
- No substitution inference:
  - explicit substitution,
  - usual recursive definition $\rightsquigarrow$ equations.

Fancy approaches

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

- Manage to infer substitution.
- High-brow category theory, e.g., involve 2-categories:
  - 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.

Prerequisites

- Locally presentable categories.
- Accessible functors.

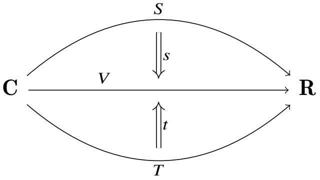
Let's pretend you all know them: may be taken as black boxes.

Inference rules

Let $\mathbf{C}$ and $\mathbf{R}$ be locally presentable.

Definition

An $\mathbf{R}$-valued inference rule over $\mathbf{C}$ consists of three accessible functors and two natural transformations as in



with T and V continuous.

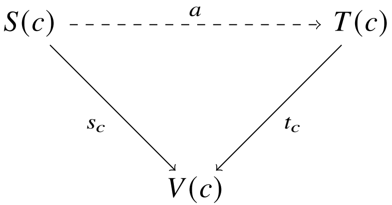
Intuition

V: metavariables, S: premises, T: conclusion.

Models

Definition

A model of $R = (V, S, T, s, t)$ is an object $c \in \mathbf{C}$ with (S, T)-dialgebra structure a making the following diagram commute.



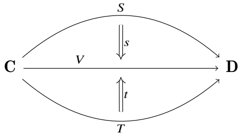
Model morphism: dialgebra morphism.
Let R-**alg** denote the category of models and morphisms between them.

Maps premises to conclusion, over fixed metavariables.

Local presentability of models

Proposition

For any



with

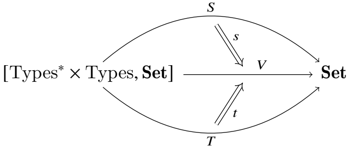
- $\mathbf{C}$ and $\mathbf{D}$ locally presentable,
- S, T, and V accessible, and
- T and V continuous,

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

Example: simply-typed application

$$\frac{\Gamma \vdash M : A \to B \qquad \Gamma \vdash N : A}{\Gamma \vdash M\ N : B}$$

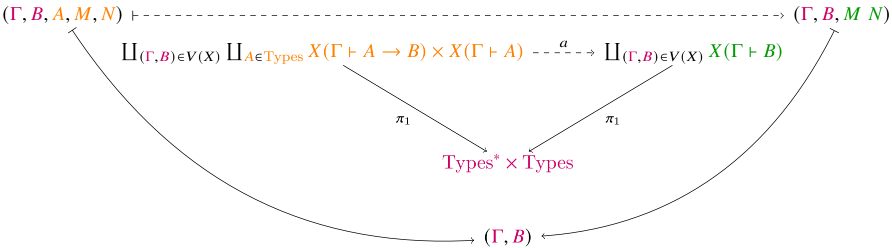
Let Types := simple types over some fixed base types.



- Notation: $X(\Gamma \vdash A) := X(\Gamma, A)$, for any $X \in [\mathrm{Types}^* \times \mathrm{Types}, \mathbf{Set}]$.
- Metavariables: Γ and B $\quad\rightsquigarrow\quad$ $V(X) = \mathrm{Types}^* \times \mathrm{Types}$.
- $S(X) = \coprod_{(\Gamma,B)\in V(X)} \coprod_{A\in\mathrm{Types}} X(\Gamma \vdash A \to B) \times X(\Gamma \vdash A)$.
- $T(X) = \coprod_{(\Gamma,B)\in V(X)} X(\Gamma \vdash B)$.

Example: simply-typed application

Models.



Equivalently:

$$\prod_{\Gamma,B} \quad \coprod_A X(\Gamma \vdash A \to B) \times X(\Gamma \vdash A) \quad \to \quad X(\Gamma \vdash B).$$

metavariables premises conclusion

Substitution inference: overview

- Take object of $\mathbf{R}$ to consist of a category + indexed set or similar.
- Carefully craft V, S, T to encode the desired substitution behaviour semantically.
- Prove continuity of V, T.

Substitution inference by example

- Indexed sets.
- Families.
- Indexed families.
- Categories with families (CwFs).
- Inference rule for dependent application.

Indexed sets

Definition

An indexed set consists of

- a category $\mathbb{C}$, and
- a functor $\mathbb{T} \colon \mathbb{C} \to \mathbf{Set}$.

Intuition:

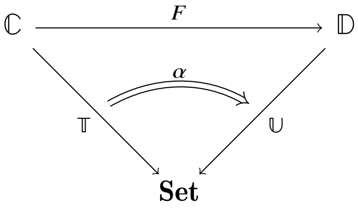
- $\mathbb{C}$: context category.
- $\mathbb{T}(\Gamma)$: family of “things” with free variables in Γ.
- $\mathbb{T}$ on morphisms: substitution.

Indexed sets

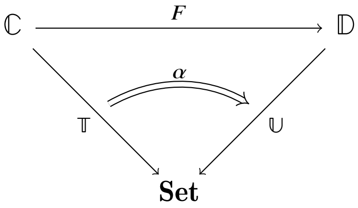
Definition

A morphism $(\mathbb{C}, \mathbb{T}) \to (\mathbb{D}, \mathbb{U})$ consists of

- a functor F between context categories and,
- for all contexts $\Gamma \in \mathbb{C}$, a map $\alpha_\Gamma \colon \mathbb{T}(\Gamma) \to \mathbb{U}(F(\Gamma))$.



Indexed sets



Proposition

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

Proposition (Makkai and Paré)

$\mathbf{Cat}//\mathbf{Set}$ is locally presentable.

Families

Definition of families

(Contravariant) presheaves over the category $[0] \xrightarrow{s} [1]$.

Intuition

For any $X \in \mathbf{Fam}$: $X[0]$: “types”, $X[1]$: “terms”.

Notation

X_A := fibre of $X[1]$ over any $A \in X[0]$.

Yoneda lemma in this case

- Types ≅ morphisms $\mathbf{y}_{[0]} \to X$.
- Terms ≅ morphisms $\mathbf{y}_{[1]} \to X$.
- Type of $\mathbf{y}_{[1]} \xrightarrow{f} X$ = $\mathbf{y}_{[0]} \xrightarrow{\mathbf{y}_s} \mathbf{y}_{[1]} \xrightarrow{f} X$.

Indexed families

Replace $\mathbf{Set}$ with $\mathbf{Fam}$.

- $\mathbb{T} : \mathbb{C} \longrightarrow \mathbf{Fam}$.
- $\mathbb{T}(\Gamma)[0]$: types in Γ.
- $\mathbb{T}(\Gamma)[1]$: terms in Γ.

$\rightsquigarrow$

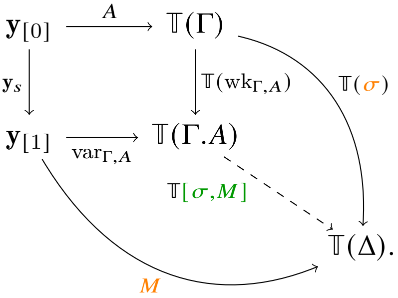
Locally presentable category $\mathbf{Cat}/\!/\mathbf{Fam}$.

Categories with families (CwFs)

A standard notion of model for type theory.

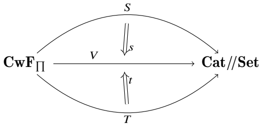
Definition

- An indexed family $\mathbb{T} \colon \mathbb{C} \to \mathbf{Fam}$.
- Context extension:
  for all $A \in \mathbb{T}(\Gamma)[0]$, $\Gamma.A$, $\mathrm{var}_{\Gamma,A}$, and $\mathrm{wk}_{\Gamma,A}$ such that



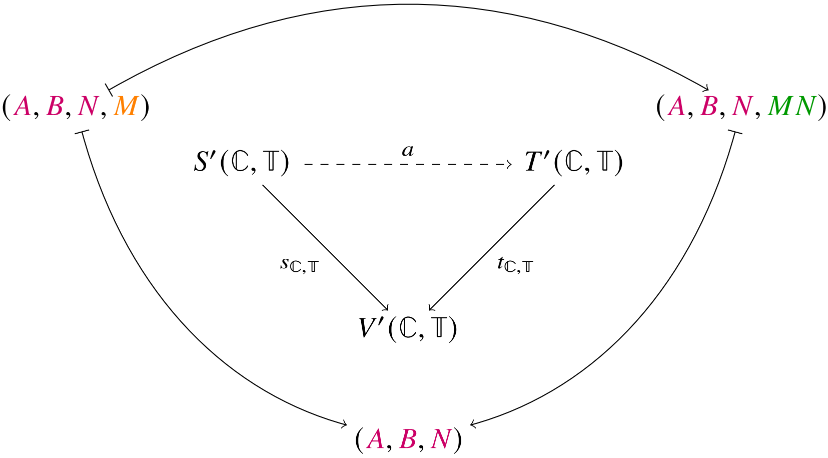
Example: dependent application (supposing we have already done $\prod$)

$$\frac{\Gamma \vdash A : \mathbf{type} \qquad \Gamma.A \vdash B : \mathbf{type} \qquad \Gamma \vdash M : \prod_A B \qquad \Gamma \vdash N : A}{\Gamma \vdash M\ N : B[N]}$$



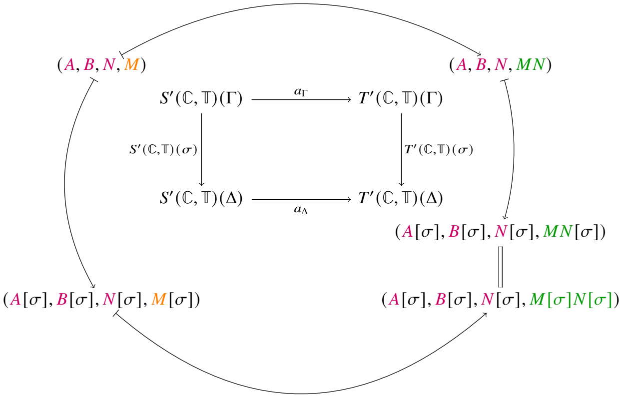
- For each $K \in \{V, S, T\}$, $K(\mathbb{C}, \mathbb{T}) = (\mathbb{C}, K'(\mathbb{C}, \mathbb{T}) : \mathbb{C} \to \mathbf{Set})$.
- $V'(\mathbb{C}, \mathbb{T})(\Gamma) = \coprod_{A,B} \mathbb{T}(\Gamma)_A$,
- $S'(\mathbb{C}, \mathbb{T})(\Gamma) = \coprod_{A,B,N \in V'(\mathbb{C},\mathbb{T})(\Gamma)} \mathbb{T}(\Gamma)_{\prod_A B}$,
- $T'(\mathbb{C}, \mathbb{T})(\Gamma) = \coprod_{A,B,N \in V'(\mathbb{C},\mathbb{T})(\Gamma)} \mathbb{T}(\Gamma)_{B[N]}$.

Example: models of dependent application



Example: models of dependent application

Naturality of a implements substitution inference.



Substitution inference: recap

- Take object of $\mathbf{R}$ to consist of a category + indexed set or similar.
- Carefully craft V, S, T to encode the desired substitution behaviour semantically.
- Prove continuity of V, T.

Next step

Find expressive constructions of functors under which continuous functors are closed.

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,
- application to quantitative type theory (Atkey, '18).