Coherent Categories with Families

jww Ambrus Kaposi

Thorsten Altenkirch

Functional Programming Laboratory
School of Computer Science
University of Nottingham

June 10, 2024

- How to define the syntax of Type Theory?
- How to define the semantics of Type Theory?

Semantics : Categories with Families (CwFs)

record SetTT : Type_1 **where**
  field

- A category of contexts and substitutions

$$\begin{aligned} \mathsf{Con} &: \mathsf{Type} \\ \mathsf{Sub} &: \mathsf{Con} \to \mathsf{Con} \to \mathsf{Type} \end{aligned}$$

- A presheaf of Types

$$\begin{aligned} \mathsf{Ty} &: \mathsf{Con} \to \mathsf{Type} \\ \_[\_] &: \mathsf{Ty}\,\Delta \to \mathsf{Sub}\,\Gamma\,\Delta \to \mathsf{Ty}\,\Gamma \\ \mathsf{A}\,[\,\sigma \mid \delta\,] &: \mathsf{A}\,[\,\sigma \circ \delta\,] \equiv \mathsf{A}\,[\,\sigma\,]\,[\,\delta\,] \end{aligned}$$

- A locally representable presheaf of terms over Ty:

$$\mathsf{Tm} : (\Gamma : \mathsf{Con}) \to \mathsf{Ty}\,\Gamma \to \mathsf{Type}$$

Syntax : Quotient Inductive-Inductive type (QIIT)

data CWF : Type **where**

- A category of contexts and substitutions

$$\begin{aligned} \mathsf{Con} &: \mathsf{Type} \\ \mathsf{Sub} &: \mathsf{Con} \to \mathsf{Con} \to \mathsf{Type} \end{aligned}$$

- A presheaf of Types

$$\begin{aligned} \mathsf{Ty} &: \mathsf{Con} \to \mathsf{Type} \\ \_[\_] &: \mathsf{Ty}\,\Delta \to \mathsf{Sub}\,\Gamma\,\Delta \to \mathsf{Ty}\,\Gamma \\ \mathsf{A}\,[\,\sigma \mid \delta\,] &: \mathsf{A}\,[\,\sigma \circ \delta\,] \equiv \mathsf{A}\,[\,\sigma\,]\,[\,\delta\,] \end{aligned}$$

- A locally presentable presheaf of terms over Ty:

$$\mathsf{Tm} : (\Gamma : \mathsf{Con}) \to \mathsf{Ty}\,\Gamma \to \mathsf{Type}$$

Intrinsic syntax of Type Theory

- Define the initial algebra as a QIIT.

POPL 2016

A., and Ambrus Kaposi. "Type theory in type theory using quotient inductive types." ACM SIGPLAN Notices 51.1 (2016)

- We add Π-types :

$$\begin{aligned}\Pi &: (A : \mathsf{Ty}\,\Gamma)\,(B : \mathsf{Ty}\,(\Gamma \triangleright A)) \to \mathsf{Ty}\,\Gamma \\ \text{lam-app} &: \mathsf{Tm}\,(\Gamma \triangleright A)\,B \cong \mathsf{Tm}\,\Gamma\,(\Pi\,A\,B)\end{aligned}$$

- To avoid the initial algebra to be empty we add a base family

$$\begin{aligned}\mathsf{U} &: \mathsf{Ty}\,\Gamma \\ \mathsf{El} &: \mathsf{Tm}\,\Gamma\,\mathsf{U} \to \mathsf{Ty}\,\Gamma\end{aligned}$$

Truncation

- We need to set-truncate Ty, Tm and Sub:

$$\begin{aligned} \mathsf{isSet}\ A\ &=\ (a\ b : A)\ (p\ q : a \equiv b) \to p \equiv q \\ \mathsf{TySet}\ &:\ \mathsf{isSet}\ (\mathsf{Ty}\ \Gamma) \\ \mathsf{TmSet}\ &:\ \mathsf{isSet}\ (\mathsf{Tm}\ \Gamma\ A) \\ \mathsf{SubSet}\ &:\ \mathsf{isSet}\ (\mathsf{Sub}\ \Gamma\ \Delta) \end{aligned}$$

- Now we want to define the set-theoretic model:

$$\begin{aligned} \mathsf{Con}\ &=\ \mathsf{Set} \\ \mathsf{Sub}\ \Gamma\ \Delta\ &=\ \Gamma \to \Delta \\ \mathsf{Ty}\ \Gamma\ &=\ \Gamma \to \mathsf{Set} \\ \mathsf{Tm}\ \Gamma\ A\ &=\ (\gamma : \Gamma) \to A\ \gamma \end{aligned}$$

  Here $\mathsf{Set} = \mathsf{HSet} = (X : \mathsf{Type}) \times \mathsf{isSet}\ X$.

- But we cannot show

$$\mathsf{isSet}\ (\mathsf{Ty}\ \Gamma)$$

Truncation

- In the POPL paper we used a inductive recursive universe instead.

$$
\begin{aligned}
U &: \mathsf{Set} \\
El &: U \to \mathsf{Set}
\end{aligned}
$$

- How can we interpret type theory in Set and other semantic theories?

Types 2021

A., and Ambrus Kaposi. "A container model of type theory." (2021).

Coherent Type Theory

- To accomodate semantic models, we raise the truncation levels.

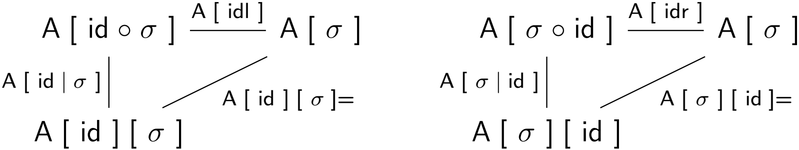
$$\begin{aligned}
&\mathsf{isGpd}\ \mathsf{A} \ = \ (\mathsf{a}\ \mathsf{b} \ : \ \mathsf{A}) \ \to \ \mathsf{isSet}\ (\mathsf{a} \ \equiv \ \mathsf{b}) \\
&\mathsf{TySet} \ : \ \mathsf{isGpd}\ (\mathsf{Ty}\ \Gamma) \\
&\mathsf{TmSet} \ : \ \mathsf{isSet}\ (\mathsf{Tm}\ \Gamma\ \mathsf{A}) \\
&\mathsf{SubSet} \ : \ \mathsf{isSet}\ (\mathsf{Sub}\ \Gamma\ \Delta)
\end{aligned}$$

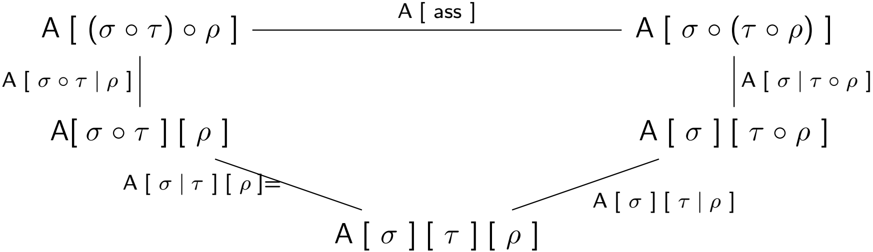
- For the syntax we get isGpd Con as a consequence.
- Now the propositional equations like

$$\mathsf{A}\,[\,\sigma \mid \delta\,] \ : \ \mathsf{A}\,[\,\sigma \circ \delta\,] \ \equiv \ \mathsf{A}\,[\,\sigma\,]\,[\,\delta\,]$$

  become data.
- To have a well behaved theory we need to add (propositional) coherence laws.

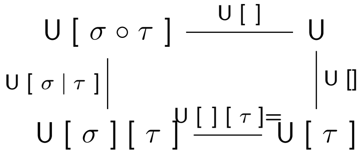
Coherence laws (1)

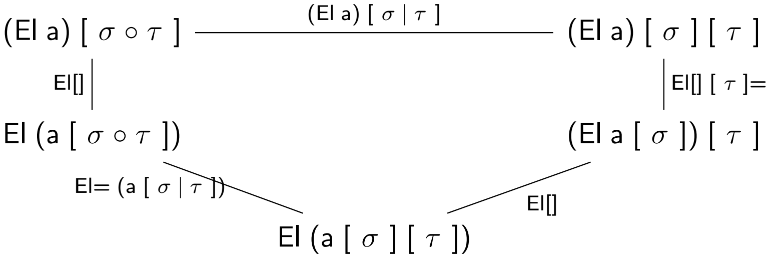




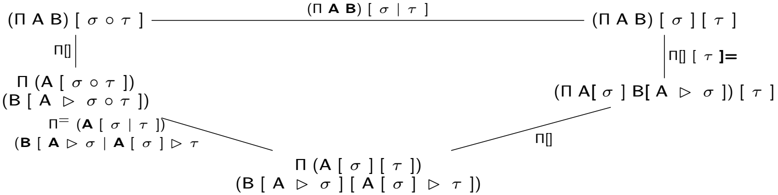
Coherence laws (2)

$$\mathsf{U[]} \ : \ \mathsf{U} \, [\, \sigma \,] \equiv \mathsf{U}$$
$$\mathsf{El[]} \ : \ (\mathsf{El}\ \mathsf{a}) \, [\, \sigma \,] \equiv \mathsf{El}\ (\mathsf{a}\, [\, \sigma \,])$$





Coherence laws (3)

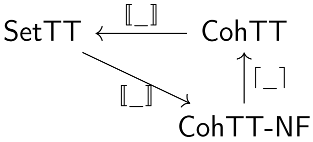


Coherent Type Theory

- CohTT = initial algebra of coherent CwF = a higher inductive type
- Can we interpret the syntax of type theory in CohTT?
- Is the equality of types in CohTT decidable?
- And hence a Set (Hedberg's theorem)?

Coherence theorem

- Theorem in progress:
  CohTT is isomorphic to SetTT.
- Basic idea: Recursive definition of substitution for CohTT: CohTT-NT.
- Normal Types NType are defined inductively without equations.



- Isomorphism: Do we need the coherence equations (1)?
- Some issues with cubical agda dealing with higher equalities using composition.

To inifinity and beyond

- Can't we have completely untruncated Type Theory?

∞-CWFs

Kraus, Nicolai, *Internal ∞-Categorical Models of Dependent Type Theory : Towards 2LTT Eating HoTT*, LICS 2021

Coherence

Uemura, Taichi. "Normalization and coherence for ∞-type theories." arXiv preprint arXiv:2212.11764 (2022).