Quotient inductive-inductive types in the setoid model

Ambrus Kaposi and Zongpu Xie
Eötvös Loránd University, Budapest

TYPES 2021
online (Leiden)
14 June 2021
The setoid model

A closed type is a setoid: a set with an equivalence relation.

The setoid model justifies (Hofmann’s thesis 1995)

- function extensionality
- propositional extensionality
- quotient types

It is a strict model in an intensional metatheory with SProp (Altenkirch LICS 1999).

- model construction
- setoid type theory
- Agda has SProp!

Goal of this talk: show that the setoid model justifies QIITs.
Quotient inductive-inductive types (QIITs)

Inductive types with possibly

- multiple sorts
- indexed over each other
- equality constructors

Examples:

- constructive ordinals (Kraus, Nordvall Forsberg, Chuangjie TODAY)
- Cauchy reals
- partiality monad
- syntax of type theory
- initial object for a generalised algebraic theory (Cartmell 1986) / essentially algebraic theory (Freyd 1972)

More constructive than quotients: avoid axiom of choice.
Example QIIT

\[
\begin{align*}
\text{Con} & : \text{Set} \\
\text{Ty} & : \text{Con} \rightarrow \text{Set} \\
\bullet & : \text{Con} \\
- \triangleright - & : (\gamma : \text{Con}) \rightarrow \text{Ty} \gamma \rightarrow \text{Con} \\
\text{U} & : \text{Ty} \gamma \\
\text{El} & : \text{Ty} (\gamma \triangleright \text{U}) \\
\Sigma & : (a : \text{Ty} \gamma) \rightarrow \text{Ty} (\gamma \triangleright a) \rightarrow \text{Ty} \gamma \\
\text{eq} & : \gamma \triangleright \Sigma a b = \gamma \triangleright a \triangleright b
\end{align*}
\]
When does a model support …

1. … inductive types?
   - Inductive types can be reduced to W types (in ITT)
     (Jasper Hugunin: Why not W? TYPES 2020)

2. Indexed / mutual inductive types?
   - Indexed W types can be reduced to W types (in ITT)

3. IITs?
   - Finitary IITs can be reduced to indexed W types (in ETT)
     (Kaposi, Kovács, Lafont TYPES 2019)

4. QIITs?
   - QIITs can be reduced to a universal QIIT (in ETT)
     (Kaposi, Kovács LICS 2020)
Computation rules?

Conservativity of ETT over ITT + funext + UIP
(Hofmann 1995, Winterhalter, Sozeau, Tabareau 2019):

\[
\frac{\Gamma \vdash_{\text{ITT}} A \quad \Gamma \vdash_{\text{ETT}} t : A}{\Gamma \vdash_{\text{ITT}} t' : A}
\]

For any QIIT Ω, the type

“the universal QIIT exists ⇒ Ω exists”

can be expressed in ITT.

If we prove that the setoid model supports the universal QIIT, then it supports all QIITs with propositional computation rules.
The universal QIIT

is a syntax for a small type theory:

- CwF
- U,El
- \( \Pi \) with domain in \( U \)
- Id with reflection
- \( \Pi \) with metatheoretic domain
- \( U \) is closed under \( \Pi \) with metatheoretic domain

4 sorts, 19 operators, 22 equations.
Example QIIT

Con : Set
Ty   : Con → Set
•    : Con
→ ▷ → : (γ : Con) → Ty γ → Con
U    : Ty γ
El   : Ty (γ ▷ U)
Σ    : (a : Ty γ) → Ty (γ ▷ a) → Ty γ
eq    : γ ▷ Σ a b = γ ▷ a ▷ b
The implementation \text{IIT}

|\text{Con}| : \text{Set}

\sim_{\text{Con}} : |\text{Con}| \rightarrow |\text{Con}| \rightarrow \text{SProp}

|\text{Ty}| : |\text{Con}| \rightarrow \text{Set}

\sim_{\text{Ty}} : \gamma \sim_{\text{Con}} \gamma' \rightarrow |\text{Ty}| \gamma \rightarrow |\text{Ty}| \gamma' \rightarrow \text{SProp}

|\bullet| : |\text{Con}|

\sim_{\bullet} : |\bullet| \sim_{\text{Con}} |\bullet|

\vdash : |\text{Con}| \rightarrow |\text{Ty}| \gamma \rightarrow |\text{Con}|

\dashed : (\bar{\gamma} : \gamma \sim_{\text{Con}} \gamma') \rightarrow \sim_{\text{Ty}} \bar{\gamma} \alpha \alpha' \rightarrow (\gamma |\dashed| \alpha) \sim_{\text{Con}} (\gamma' |\dashed| \alpha')

\ldots

|\text{eq}| : \gamma |\dashed| |\Sigma| a b \sim_{\text{Con}} \gamma |\dashed| a |\dashed| b

\sim_{\text{Con}}, \sim_{\text{Ty}} \text{ are reflexive, symmetric and transitive}

\text{coe}_{\text{Ty}} : \gamma \sim_{\text{Con}} \gamma' \rightarrow |\text{Ty}| \gamma \rightarrow |\text{Ty}| \gamma'

\text{coh}_{\text{Ty}} : (\bar{\gamma} : \gamma \sim_{\text{Con}} \gamma')(\alpha : |\text{Ty}| \gamma) \rightarrow \sim_{\text{Ty}} \bar{\gamma} \alpha (\text{coe}_{\text{Ty}} \bar{\gamma} \alpha)
The QIIT in the setoid model

1. The type formation rules and constructors are defined in the empty context using the implementation IIT.

2. The recursor for a Con-Ty algebra *in the empty context* is defined by recursion-recursion on this IIT.

3. An algebra in an arbitrary context is turned into an algebra in the empty context, i.e. we turn a type \( \Gamma \vdash C \) into a type \( \cdot \vdash \Pi(x : K \Gamma).C[x] \)

4. Now we can use the recursor in any context. Its \( \beta \) rules are definitional. Its substitution laws are proven by induction-induction on the IIT.

5. Uniqueness is proved by induction-induction on the IIT.

For Con-Ty, we formalised this in Agda.
For the universal QIIT, we formalised steps 1,2,5 in Agda.
Summary

- The setoid model is poor man’s cubical model.
- QIITs: free generalised algebraic theories.
- QIITs with propositional computation rules can be reduced to a universal QIIT.
- The universal QIIT can be defined in the setoid model (WIP: lifting to arbitrary contexts).
- Further work: give a direct definition of a QIIT in the setoid model for an arbitrary signature. This would give definitional $\beta$ rules.