A General Construction of Strict Models in HoTT

Niyousha Najmaei, **Niels van der Weide**

Let's start with some history

It was 2016 and Altenkirch and Kaposi published a paper[1]

- **Main contribution of that paper**: we can construct type theory as a quotient inductive-inductive type (QIIT)
- Very nice and extendible construction

[1] "Type theory in type theory using quotient inductive types", Altenkirch, Kaposi

Let's start with some history

It was 2016 and Altenkirch and Kaposi published a paper[1]

- **Main contribution of that paper**: we can construct type theory as a quotient inductive-inductive type (QIIT)
- Very nice and extendible construction

But there was a challenge...

- If one uses their QIIT, then one can only eliminate into types whose identity types are "trivial" (i.e., unique inhabitant)
- However, this is problematic in homotopy type theory (HoTT), where various types do not satisfy that condition

[1] "Type theory in type theory using quotient inductive types", Altenkirch, Kaposi

Let's start with some history

It was 2016 and Altenkirch and Kaposi published a paper[1]

- **Main contribution of that paper**: we can construct type theory as a quotient inductive-inductive type (QIIT)
- Very nice and extendible construction

But there was a challenge...

- If one uses their QIIT, then one can only eliminate into types whose identity types are "trivial" (i.e., unique inhabitant)
- However, this is problematic in homotopy type theory (HoTT), where various types do not satisfy that condition
- **Consequence**: they couldn't construct interpretation of the syntax in sets, i.e., they can only define **"strict" models**

[1] "Type theory in type theory using quotient inductive types", Altenkirch, Kaposi

Let's start with some history

It was 2016 and Altenkirch and Kaposi published a paper[1]

- **Main contribution of that paper**: we can construct type theory as a quotient inductive-inductive type (QIIT)
- Very nice and extendible construction

But there was a challenge...

- If one uses their QIIT, then one can only eliminate into types whose identity types are "trivial" (i.e., unique inhabitant)
- However, this is problematic in homotopy type theory (HoTT), where various types do not satisfy that condition
- **Consequence**: they couldn't construct interpretation of the syntax in sets, i.e., they can only define **"strict" models**

To get the "standard model", Altenkirch and Kaposi were forced to use an inductive-recursive type instead of sets (as in HoTT)

[1] "Type theory in type theory using quotient inductive types", Altenkirch, Kaposi

Nothing happened for a while until...

At HoTT/UF2023, Gratzer, Gylterud, Mörtberg, and Stenholm presented another strict model[2]

- **Key idea**: identify the universe of iterative sets, and show that they form a strict model

[2] "The Category of Iterative Sets in Homotopy Type Theory and Univalent Foundations", Gratzer, Gylterud, Mörtberg, Stenholm

[3] "The type theoretic interpretation of constructive set theory", Aczel

Nothing happened for a while until...

At HoTT/UF2023, Gratzer, Gylterud, Mörtberg, and Stenholm presented another strict model[2]

- **Key idea**: identify the universe of iterative sets, and show that they form a strict model
- This approach works, because iterative sets form a set (in the sense of HoTT)
- Their approach is inspired by Aczel's work on the relation between type theory and set theory[3]

[2] "The Category of Iterative Sets in Homotopy Type Theory and Univalent Foundations", Gratzer, Gylterud, Mörtberg, Stenholm

[3] "The type theoretic interpretation of constructive set theory", Aczel

Nothing happened for a while until...

At HoTT/UF2023, Gratzer, Gylterud, Mörtberg, and Stenholm presented another strict model[2]

- **Key idea**: identify the universe of iterative sets, and show that they form a strict model
- This approach works, because iterative sets form a set (in the sense of HoTT)
- Their approach is inspired by Aczel's work on the relation between type theory and set theory[3]

This might feel very similar to what Altenkirch and Kaposi did:

identify a universe

and construct a strict model by restricting to that universe

[2] "The Category of Iterative Sets in Homotopy Type Theory and Univalent Foundations", Gratzer, Gylterud, Mörtberg, Stenholm

[3] "The type theoretic interpretation of constructive set theory", Aczel

And then there was another model

At HoTT/UF2024, Damato presented a strict container model[4].

- Starting point: an inductive-recursive universe for types
- Use this universe to construct a **set** of all containers
- The right adjoint splitting (i.e., delaying substitution) is used

This model has similar vibes to what we saw before, but it is not presented as a "container universe of containers"

[4] "Containers, Cubes, and Coherences: Containers in Homotopy Type Theory", Damato

And then there was another model

At HoTT/UF2024, Damato presented a strict container model[4].

- Starting point: an inductive-recursive universe for types
- Use this universe to construct a **set** of all containers
- The right adjoint splitting (i.e., delaying substitution) is used

This model has similar vibes to what we saw before, but it is not presented as a "container universe of containers"
Note: Damato talked about a different approach 7 minutes ago

[4] "Containers, Cubes, and Coherences: Containers in Homotopy Type Theory", Damato

Other Examples of Interest

And there are other examples of interest

- Presheaves[5]
- Realisability (assemblies)[6]

A similar approach should be usable for these

[5] "Lifting Grothendieck universes", Hofmann, Streicher

[6] "An extended calculus of constructions", Luo

Overview of this Talk

Goal of this talk: we present a general construction of strict models within HoTT

- Starting point: a "weak" model. Types are not necessarily a set, substitution laws up to isomorphisms

Overview of this Talk

Goal of this talk: we present a general construction of strict models within HoTT

- Starting point: a "weak" model. Types are not necessarily a set, substitution laws up to isomorphisms
- Same idea as before: **restrict the model to the universe**
- We show that we get a category with families (under mild conditions): types are a set and substitution laws up to equality

Overview of this Talk

Goal of this talk: we present a general construction of strict models within HoTT

- Starting point: a "weak" model. Types are not necessarily a set, substitution laws up to isomorphisms
- Same idea as before: **restrict the model to the universe**
- We show that we get a category with families (under mild conditions): types are a set and substitution laws up to equality
- Our construction is formalised in the Rocq proof assistant using the UniMath library

Main Idea

Let $\chi : \mathcal{E} \to \mathcal{C}^{\to}$ be a comprehension category with a universe $\mathcal{U}$

- Types in χ are **not** guaranteed to form a set
- Terms in χ do form a set

Main Idea

Let $\chi : \mathcal{E} \to \mathcal{C}^{\to}$ be a comprehension category with a universe $\mathcal{U}$

- Types in χ are **not** guaranteed to form a set
- Terms in χ do form a set
- **Main idea**: identify a submodel of χ where types are **terms of type** $\mathcal{U}$
- Under mild conditions (see next slide) this forms a CwF

Conditions for the Universe

We require the following conditions on the universe:

- Contains the unit type: needed for the empty context
- Closed under $\sum$-types: needed for context extension

We won't give the definitions here.

Main Theorem

Theorem

Let $\chi : \mathcal{E} \to \mathcal{C}^{\to}$ be a comprehension category such that

- *χ has $\sum$-types and unit types.*
- *χ has a universe $\mathcal{U}$*
- *$\mathcal{U}$ contains the unit type and $\sum$-types*

Then χ induces a category with families.

Universes in Comprehension Categories

Definition

Suppose, we have a comprehension category $\chi : \mathcal{E} \to \mathcal{C}^{\to}$. A **universe** for χ consists of

- a type $\mathcal{U}$ in the empty context
- a map **el**: sends terms $t : \mathbf{tm}(\Gamma, \mathcal{U})$ to types $\mathbf{el}(t) : \mathbf{ty}(\Gamma)$

We require **el** to be coherently stable under substitution

Similar definitions can be found throughout the literature[7] [8] [9]

[7] "The groupoid-syntax of type theory is a set", Altenkirch, Kaposi, Xie

[8] "Principles of Dependent Type Theory", Angiuli, Gratzer

[9] "The internal languages of univalent categories", Van der Weide

Overview of the Construction

Let $\chi : \mathcal{E} \to \mathcal{C}^{\to}$ be a comprehension category with a universe $\mathcal{U}$. We define a CwF $\overline{\chi}$ as follows:

- Contexts: $\mathbf{tm}([], \mathcal{U})$
- Types in context Γ: $\mathbf{tm}(\mathbf{el}(\Gamma), \mathcal{U})$ (I am too lazy to add weakening here, because that isn't really the point)
- Terms in context Γ of type A: $\mathbf{tm}(\mathbf{el}(\Gamma), \mathbf{el}(A))$

[10] "A C-system defined by a universe category", Voevodsky

Overview of the Construction

Let $\chi : \mathcal{E} \to \mathcal{C}^{\to}$ be a comprehension category with a universe $\mathcal{U}$. We define a CwF $\overline{\chi}$ as follows:

- Contexts: $\mathbf{tm}([], \mathcal{U})$
- Types in context Γ: $\mathbf{tm}(\mathbf{el}(\Gamma), \mathcal{U})$ (I am too lazy to add weakening here, because that isn't really the point)
- Terms in context Γ of type A: $\mathbf{tm}(\mathbf{el}(\Gamma), \mathbf{el}(A))$

Note:

- Since terms in a comprehension category form a set, we indeed have a set of types
- Substitution laws for types hold strictly in the new model
- Similar to **externalisation of internal categories** and Voevodsky's strictification construction[10]

[10] "A C-system defined by a universe category", Voevodsky

Type Formers

The CwF $\overline{\chi}$ inherits type formers from $\mathcal{U}$

- if $\mathcal{U}$ is closed under a certain type former, then $\overline{\chi}$ can be equipped with such types
- for instance: if $\mathcal{U}$ is closed under $\prod$-types, then $\overline{\chi}$ interprets $\prod$-types

Type Formers

The CwF $\overline{\chi}$ inherits type formers from $\mathcal{U}$

- if $\mathcal{U}$ is closed under a certain type former, then $\overline{\chi}$ can be equipped with such types
- for instance: if $\mathcal{U}$ is closed under $\prod$-types, then $\overline{\chi}$ interprets $\prod$-types

Note:

- $\overline{\chi}$ always interprets $\sum$-types and the unit type
- $\overline{\chi}$ always is democratic

Conclusion

Main points:

- We presented a general construction of strict models from comprehension categories with a universe
- Our construction works in HoTT and we don't need to assume that types form a set in the given model
- Our construction is formalised in the Rocq proof assistant using UniMath

Conclusion

Main points:

- We presented a general construction of strict models from comprehension categories with a universe
- Our construction works in HoTT and we don't need to assume that types form a set in the given model
- Our construction is formalised in the Rocq proof assistant using UniMath

The essence:

- For the type theorist: restricting to a universe gives a model of type theory (useful for strictification)
- For the category theorist: externalisation of internal categories (internal to comprehension categories)