Specifying QIITs using Containers

Stefania Damato   Thorsten Altenkirch

University of Nottingham, UK

Workshop on
Homotopy Type Theory/Univalent Foundations

23rd April 2023

Quotient inductive-inductive types (QIITs)

$$\text{QIITs} = \begin{cases} \text{Quotient inductive types} \\ \\ \end{cases}$$

Quotient inductive-inductive types (QIITs)

$$\text{QIITs} = \begin{cases} \text{Quotient inductive types} \\ \text{Inductive-inductive types} \qquad A : \text{Type} \\ \qquad\qquad\qquad\qquad\qquad\quad B : A \to \text{Type} \end{cases}$$

Quotient inductive-inductive types (QIITs)

$$\text{QIITs} = \begin{cases} \text{Quotient inductive types} \\ \text{Inductive-inductive types} \qquad A : \text{Type} \\ \qquad\qquad\qquad\qquad\qquad\quad B : A \to \text{Type} \end{cases}$$

Example

```
data Con : Set
data Ty : Con → Set

data Con where
   ◇ : Con
   _,_ : (Γ : Con) (A : Ty Γ) → Con
   eq : (Γ : Con) (A : Ty Γ) (B : Ty (Γ , A)) →
        ((Γ , A) , B) ≡ (Γ , Σ Γ A B)

data Ty where
   ι : (Γ : Con) → Ty Γ
   Σ : (Γ : Con) (A : Ty Γ) → Ty (Γ , A) → Ty Γ
```

Specifications of inductive types

Class of types	**Functor type**	**Category theory semantics**	**Type theoretic normal form**	**Universal type**
ordinary inductive types e.g. $\mathbb{N}$: Set	$\mathbf{Set} \to \mathbf{Set}$	initial algebras of endofunctors on **Set**	containers	W-type
inductive families e.g. $\texttt{Fin} : \mathbb{N} \to \mathrm{Set}$	$(\mathbf{I} \to \mathbf{Set}) \to (\mathbf{I} \to \mathbf{Set})$	initial algebras of endofuntors on $\mathbf{Set}^I$	indexed containers	WI-type

Specifications of inductive types

Class of types	**Functor type**	**Category theory semantics**	**Type theoretic normal form**	**Universal type**
ordinary inductive types e.g. $\mathbb{N} : \mathrm{Set}$	$\mathbf{Set} \to \mathbf{Set}$	initial algebras of endofunctors on **Set**	containers	W-type
inductive families e.g. $\mathtt{Fin} : \mathbb{N} \to \mathrm{Set}$	$(\mathbf{I} \to \mathbf{Set}) \to (\mathbf{I} \to \mathbf{Set})$	initial algebras of endofuntors on $\mathbf{Set}^I$	indexed containers	WI-type
QIITs e.g. $\mathtt{Con} : \mathrm{Set},$ $\mathtt{Ty} : \mathtt{Con} \to \mathrm{Set}$	?	?	?	?

Specifications of inductive types

Class of types	Functor type	Category theory semantics	Type theoretic normal form	Universal type
ordinary inductive types e.g. $\mathbb{N} : \text{Set}$	$\mathbf{Set} \to \mathbf{Set}$	initial algebras of endofunctors on **Set**	containers	W-type
inductive families e.g. $\texttt{Fin} : \mathbb{N} \to \text{Set}$	$(\mathbf{I} \to \mathbf{Set}) \to (\mathbf{I} \to \mathbf{Set})$	initial algebras of endofuntors on $\mathbf{Set}^I$	indexed containers	WI-type
QIITs e.g. $\texttt{Con} : \text{Set}$, $\texttt{Ty} : \texttt{Con} \to \text{Set}$	seemingly impossible, but we have an alternative	?	?	?

Representing QIITs [Altenkirch et al., 2018]

1. Category $\mathbf{A_0}$ of sorts.

Representing QIITs [Altenkirch et al., 2018]

1. Category $\mathbf{A_0}$ of sorts.
2. Constructor specification. The n^{th} constructor is specified by two functors

$$L_n \colon \mathbf{A_n} \to \mathbf{Set},$$
$$R_n \colon \int L_n \to \mathbf{Set}.$$

Representing QIITs [Altenkirch et al., 2018]

1. Category $\mathbf{A_0}$ of sorts.
2. Constructor specification. The n^{th} constructor is specified by two functors

$$L_n \colon \mathbf{A_n} \to \mathbf{Set},$$
$$R_n \colon \int L_n \to \mathbf{Set}.$$

3. Category of algebras. $\mathbf{A_{n+1}}$ is the category having
   - objects of type $\sum(A : |\mathbf{A_n}|)(c \colon (x : L_n\, A) \to R_n(A, x))$
   - morphisms $(A, c) \to (A', c')$ are morphisms $f \colon A \to A'$ in $\mathbf{A_n}$ such that

$$\begin{array}{ccc} (x : L_n\, A) & \xrightarrow{L_n\, f} & ((L_n\, f)\, x : L_n\, A') \\ {\scriptstyle c}\downarrow & & \downarrow{\scriptstyle c'} \\ R_n(A, x) & \xrightarrow[R_n\, \bar{f}]{} & R_n(A', (L_n\, f)\, x) \end{array}$$

where $\bar{f}$ is the morphism in $\int L_n$ determined by f.

Specifications of inductive types (revisited)

Class of types	**Functor type**	**Category theory semantics**	**Type theoretic normal form**	**Universal type**
ordinary inductive types e.g. $\mathbb{N}$: Set	$\mathbf{Set} \to \mathbf{Set}$	initial algebras of endofunctors on **Set**	containers	W-type
inductive families e.g. $\texttt{Fin} : \mathbb{N} \to \mathsf{Set}$	$(\mathbf{I} \to \mathbf{Set}) \to (\mathbf{I} \to \mathbf{Set})$	initial algebras of endofuntors on $\mathbf{Set}^I$	indexed containers	WI-type
QIITs e.g. $\texttt{Con} : \mathsf{Set}$, $\texttt{Ty} : \texttt{Con} \to \mathsf{Set}$	seemingly impossible, but we have an alternative	?	?	?

Specifications of inductive types (revisited)

Class of types	**Representation**	**Category theory semantics**	**Type theoretic normal form**	**Universal type**
ordinary inductive types e.g. $\mathbb{N} : \mathrm{Set}$	functor $\mathbf{Set} \to \mathbf{Set}$	initial algebras of endofunctors on **Set**	containers	W-type
inductive families e.g. $\mathtt{Fin} : \mathbb{N} \to \mathrm{Set}$	functor $(\mathbf{I} \to \mathbf{Set}) \to (\mathbf{I} \to \mathbf{Set})$	initial algebras of endofuntors on $\mathbf{Set}^I$	indexed containers	WI-type
QIITs e.g. $\mathtt{Con} : \mathrm{Set}$, $\mathtt{Ty} : \mathtt{Con} \to \mathrm{Set}$	sequence of functors L_n and R_n and sequence of categories of dialgebras	initial object in last constructed category of dialgebras $\mathbf{A_n}$	?	?

Specifications of inductive types (revisited)

Class of types	**Representation**	**Category theory semantics**	**Type theoretic normal form**	**Universal type**
ordinary inductive types e.g. $\mathbb{N}$: Set	functor $\mathbf{Set} \to \mathbf{Set}$	initial algebras of endofunctors on **Set**	containers	W-type
inductive families e.g. $\texttt{Fin} : \mathbb{N} \to \text{Set}$	functor $(\mathbf{I} \to \mathbf{Set}) \to (\mathbf{I} \to \mathbf{Set})$	initial algebras of endofuntors on $\mathbf{Set}^I$	indexed containers	WI-type
QIITs e.g. $\texttt{Con} : \text{Set}$, $\texttt{Ty} : \texttt{Con} \to \text{Set}$	sequence of functors L_n and R_n and sequence of categories of dialgebras	initial object in last constructed category of dialgebras $\mathbf{A_n}$	representations constructed via generalised containers	QW-type

Generalised containers

We require $L_n \colon \mathbf{A_n} \to \mathbf{Set}$ and $R_n \colon \int L_n \to \mathbf{Set}$ to be **generalised container functors** (+ other restrictions on R_n).

Generalised containers

We require $L_n \colon \mathbf{A_n} \to \mathbf{Set}$ and $R_n \colon \int L_n \to \mathbf{Set}$ to be **generalised container functors** (+ other restrictions on R_n).

Definition

A *generalised container* $S \triangleleft P$ over a category $\mathbf{C}$ is a pair $S : \mathrm{Set}$ and $P \colon S \to |\mathbf{C}|$.

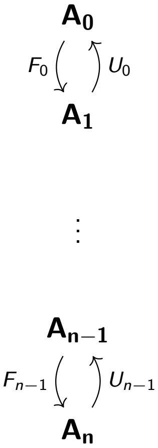
Definition

The *generalised container extension functor* associated to $S \triangleleft P$ and having type $\mathbf{C} \to \mathbf{Set}$, is defined by

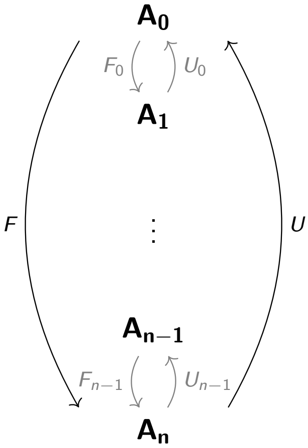
$$[\![S \triangleleft P]\!]\, X := \sum (s : S)(\mathbf{C}(P\, s, X))$$

on objects $X : |\mathbf{C}|$.

Illustration



Illustration



Conclusion

- QIITs combine set-truncated equalities with induction-induction.
- We can represent QIITs semantically as initial dialgebras.
- **Conjecture**: The categories of dialgebras having an initial object are those whose constructors are restricted to **generalised container functors**.
- **Question**: What does a **universal QW-type** look like?

Conclusion

- QIITs combine set-truncated equalities with induction-induction.
- We can represent QIITs semantically as initial dialgebras.
- **Conjecture**: The categories of dialgebras having an initial object are those whose constructors are restricted to **generalised container functors**.
- **Question**: What does a **universal QW-type** look like?

Thank you!

References

Altenkirch, T., Capriotti, P., Dijkstra, G., Kraus, N., and Nordvall Forsberg, F. (2018).
Quotient inductive-inductive types.
In Baier, C. and Dal Lago, U., editors, *FoSSACS*, pages 293–310. Springer.

Altenkirch, T. and Kaposi, A. (2021).
A container model of type theory.
In *TYPES 2021*.

Fiore, M. P., Pitts, A. M., and Steenkamp, S. (2021).
Quotients, inductive types, and quotient inductive types.
Log. Methods Comput. Sci., 18.

Kaposi, A., Kovács, A., and Altenkirch, T. (2019).
Constructing quotient inductive-inductive types.
Proc. ACM Program. Lang., 3(POPL).