A model of type theory with quotient inductive-inductive types

Szumi Xie

Eötvös Loránd University, Budapest

j.w.w. Ambrus Kaposi

TYPES 2020

Torino, 2 March 2020

SZÉCHENYI 2020

Magyarország Kormánya

Európai Unió
Európai Szociális Alap

BEFEKTETÉS A JÖVŐBE

Overview

1. Quotient inductive-inductive types (QIITs)
2. Specification of QIITs in a model
3. The setoid model
4. Implementation of the universal QIIT in the setoid model
5. Conclusion

Plan

1. Quotient inductive-inductive types (QIITs)
2. Specification of QIITs in a model
3. The setoid model
4. Implementation of the universal QIIT in the setoid model
5. Conclusion

Examples of QIITs

$$
\begin{aligned}
\mathsf{Con} &: \mathsf{Set} \\
\mathsf{Ty} &: \mathsf{Con} \to \mathsf{Set} \\
\bullet &: \mathsf{Con} \\
- \rhd - &: (\Gamma : \mathsf{Con}) \to \mathsf{Ty}\ \Gamma \to \mathsf{Con} \\
\mathsf{U} &: \mathsf{Ty}\ \Gamma \\
\mathsf{El} &: \mathsf{Ty}\ (\Gamma \rhd \mathsf{U}) \\
\Sigma &: (A : \mathsf{Ty}\ \Gamma) \to \mathsf{Ty}\ (\Gamma \rhd A) \to \mathsf{Ty}\ \Gamma \\
\mathsf{eq} &: \Gamma \rhd \Sigma\ A\ B = \Gamma \rhd A \rhd B
\end{aligned}
$$

Other examples: Cauchy real numbers, partiality monad, intrinsic syntax for programming languages

What is a QIIT?

- A context in the universal QIIT.
- Universal QIIT is a syntax for a small type theory
- Example: $\mathsf{Nat} : \mathsf{U}$, $\mathsf{zero} : \mathsf{El}\ \mathsf{Nat}$, $\mathsf{suc} : \mathsf{Nat} \Rightarrow \mathsf{El}\ \mathsf{Nat}$
- All QIITs can be constructed from the universal QIIT (Kaposi, Kovács, Altenkirch, POPL 2019)

Is this definition circular?

- A model of type theory supports the universal QIIT:
  - notion of algebra
  - notion of homomorphism
  - there is an algebra (constructor)
  - for every other algebra there is a homomorphism from the constructor to that algebra (recursor)
  - the recursor is unique

Is this definition circular?

- A model of type theory supports the universal QIIT:
  - notion of algebra
  - notion of homomorphism
  - there is an algebra (constructor)
  - for every other algebra there is a homomorphism from the constructor to that algebra (recursor)
  - the recursor is unique
- Rest of this talk:
  - How to express all of these for a model of type theory
  - Define a model which supports the universal QIIT
  - Everything was formalised in Agda

Plan

1. Quotient inductive-inductive types (QIITs)
2. Specification of QIITs in a model
3. The setoid model
4. Implementation of the universal QIIT in the setoid model
5. Conclusion

Model of type theory (i)

CwF with extra structure:

$$
\begin{aligned}
\mathsf{Con} &: \mathsf{Set} \\
\mathsf{Ty} &: \mathsf{Con} \to \mathsf{Set} \\
\mathsf{Sub} &: \mathsf{Con} \to \mathsf{Con} \to \mathsf{Set} \\
\mathsf{Tm} &: (\Gamma : \mathsf{Con}) \to \mathsf{Ty}\ \Gamma \to \mathsf{Set} \\
\Pi &: (A : \mathsf{Ty}\ \Gamma) \to \mathsf{Ty}\ (\Gamma \triangleright A) \to \mathsf{Ty}\ \Gamma \\
\mathsf{lam} &: \mathsf{Tm}\ (\Gamma \triangleright A)\ B \to \mathsf{Tm}\ \Gamma\ (\Pi\ A\ B) \\
\mathsf{app} &: \mathsf{Tm}\ \Gamma\ (\Pi\ A\ B) \to \mathsf{Tm}\ (\Gamma \triangleright A)\ B \\
&\vdots
\end{aligned}
$$

Model of type theory (ii)

Con	$: \mathbb{N} \to \mathsf{Set}$
Ty	$: \mathbb{N} \to \mathsf{Con}\, i \to \mathsf{Set}$
Sub	$: \mathsf{Con}\, i \to \mathsf{Con}\, j \to \mathsf{Set}$
Tm	$: (\Gamma : \mathsf{Con}\, i) \to \mathsf{Ty}\, j\, \Gamma \to \mathsf{Set}$
id	$: \mathsf{Sub}\, \Gamma\, \Gamma$
$- \circ -$	$: \mathsf{Sub}\, \Theta\, \Delta \to \mathsf{Sub}\, \Gamma\, \Theta \to \mathsf{Sub}\, \Gamma\, \Delta$
ass	$: (\sigma \circ \delta) \circ \nu = \sigma \circ (\delta \circ \nu)$
idl	$: \mathsf{id} \circ \sigma = \sigma$
idr	$: \sigma \circ \mathsf{id} = \sigma$
$-[-]$	$: \mathsf{Ty}\, i\, \Delta \to \mathsf{Sub}\, \Gamma\, \Delta \to \mathsf{Ty}\, i\, \Gamma$
$-[-]$	$: \mathsf{Tm}\, \Delta\, A \to (\sigma : \mathsf{Sub}\, \Gamma\, \Delta) \to \mathsf{Tm}\, \Gamma\, (A[\sigma])$
$[\mathsf{id}]$	$: A[\mathsf{id}] = A$
$[\circ]$	$: A[\sigma \circ \delta] = A[\sigma][\delta]$
$[\mathsf{id}]$	$: t[\mathsf{id}] = t$
$[\circ]$	$: t[\sigma \circ \delta] = t[\sigma][\delta]$
$\cdot$	$: \mathsf{Con}\, 0$
ϵ	$: \mathsf{Sub}\, \Gamma\, \cdot$
$\cdot\eta$	$: (\sigma : \mathsf{Sub}\, \Gamma\, \cdot) = \epsilon$
$- \rhd -$	$: (\Gamma : \mathsf{Con}\, i) \to \mathsf{Ty}\, j\, \Gamma \to \mathsf{Con}\, (i \sqcup j)$
$-,-$	$: (\sigma : \mathsf{Sub}\, \Gamma\, \Delta) \to \mathsf{Tm}\, \Gamma\, (A[\sigma]) \to \mathsf{Sub}\, \Gamma\, (\Delta \rhd A)$
p	$: \mathsf{Sub}\, (\Gamma \rhd A)\, \Gamma$
q	$: \mathsf{Tm}\, (\Gamma \rhd A)\, (A[\mathsf{p}])$
$\rhd\beta_1$	$: \mathsf{p} \circ (\sigma, t) = \sigma$
$\rhd\beta_2$	$: \mathsf{q}[\sigma, t] = t$
$\rhd\eta$	$: (\mathsf{p}, \mathsf{q}) = \mathsf{id}$
$,\circ$	$: (\sigma, t) \circ \nu = (\sigma \circ \nu, t[\nu])$
Π	$: (A : \mathsf{Ty}\, i\, \Gamma) \to \mathsf{Ty}\, j\, (\Gamma \rhd A) \to \mathsf{Ty}\, (i \sqcup j)\, \Gamma$
lam	$: \mathsf{Tm}\, (\Gamma \rhd A)\, B \to \mathsf{Tm}\, \Gamma\, (\Pi\, A\, B)$
app	$: \mathsf{Tm}\, \Gamma\, (\Pi\, A\, B) \to \mathsf{Tm}\, (\Gamma \rhd A)\, B$
$\Pi\beta$	$: \mathsf{app}\, (\mathsf{lam}\, t) = t$
$\Pi\eta$	$: \mathsf{lam}\, (\mathsf{app}\, t) = t$
$\Pi[]$	$: (\Pi\, A\, B)[\sigma] = \Pi\, (A[\sigma])\, (B[\sigma^\uparrow])$
$\mathsf{lam}[]$	$: (\mathsf{lam}\, t)[\sigma] = \mathsf{lam}\, (t[\sigma^\uparrow])$

Σ	$: (A : \mathsf{Ty}\, i\, \Gamma) \to \mathsf{Ty}\, j\, (\Gamma \rhd A) \to \mathsf{Ty}\, (i \sqcup j)\, \Gamma$
$-,-$	$: (u : \mathsf{Tm}\, \Gamma\, A) \to \mathsf{Tm}\, \Gamma\, (B[\mathsf{id}, u]) \to \mathsf{Tm}\, \Gamma\, (\Sigma\, A\, B)$
projl	$: \mathsf{Tm}\, \Gamma\, (\Sigma\, A\, B) \to \mathsf{Tm}\, \Gamma\, A$
projr	$: (t : \mathsf{Tm}\, \Gamma\, (\Sigma\, A\, B)) \to \mathsf{Tm}\, \Gamma\, (B[\mathsf{id}, \mathsf{projl}\, t])$
$\Sigma\beta_1$	$: \mathsf{projl}\, (u, v) = u$
$\Sigma\beta_2$	$: \mathsf{projr}\, (u, v) = v$
$\Sigma\eta$	$: (\mathsf{projl}\, t, \mathsf{projr}\, t) = t$
$\Sigma[]$	$: (\Sigma\, A\, B)[\sigma] = \Sigma\, (A[\sigma])\, (B[\sigma^\uparrow])$
$,[]$	$: (u, v)[\sigma] = (u[\sigma], v[\sigma])$
$\top$	$: \mathsf{Ty}\, 0\, \Gamma$
tt	$: \mathsf{Tm}\, \Gamma\, \top$
$\top\eta$	$: (t : \mathsf{Tm}\, \Gamma\, \top) = \mathsf{tt}$
$\top[]$	$: \top[\sigma] = \top$
$\mathsf{tt}[]$	$: \mathsf{tt}[\sigma] = \mathsf{tt}$
U	$: (i : \mathbb{N}) \to \mathsf{Ty}\, (i + 1)\, \Gamma$
El	$: \mathsf{Tm}\, \Gamma\, (\mathsf{U}\, i) \to \mathsf{Ty}\, i\, \Gamma$
c	$: \mathsf{Ty}\, i\, \Gamma \to \mathsf{Tm}\, \Gamma\, (\mathsf{U}\, i)$
$\mathsf{U}\beta$	$: \mathsf{El}\, (\mathsf{c}\, A) = A$
$\mathsf{U}\eta$	$: \mathsf{c}\, (\mathsf{El}\, a) = a$
$\mathsf{U}[]$	$: (\mathsf{U}\, i)[\sigma] = (\mathsf{U}\, i)$
$\mathsf{El}[]$	$: (\mathsf{El}\, a)[\sigma] = \mathsf{El}\, (a[\sigma])$
Bool	$: \mathsf{Ty}\, 0\, \Gamma$
true	$: \mathsf{Tm}\, \Gamma\, \mathsf{Bool}$
false	$: \mathsf{Tm}\, \Gamma\, \mathsf{Bool}$
if	$: (C : \mathsf{Ty}\, i\, (\Gamma \rhd \mathsf{Bool})) \to \mathsf{Tm}\, \Gamma\, (C[\mathsf{id}, \mathsf{true}]) \to \mathsf{Tm}\, \Gamma\, (C[\mathsf{id}, \mathsf{false}]) \to (t : \mathsf{Tm}\, \Gamma\, \mathsf{Bool}) \to \mathsf{Tm}\, \Gamma\, (C[\mathsf{id}, t])$
$\mathsf{Bool}\beta_1$	$: \mathsf{if}\, C\, u\, v\, \mathsf{true} = u$
$\mathsf{Bool}\beta_2$	$: \mathsf{if}\, C\, u\, v\, \mathsf{false} = v$
$\mathsf{Bool}[]$	$: \mathsf{Bool}[\sigma] = \mathsf{Bool}$
$\mathsf{true}[]$	$: \mathsf{true}[\sigma] = \mathsf{true}$
$\mathsf{false}[]$	$: \mathsf{false}[\sigma] = \mathsf{false}$
$\mathsf{if}[]$	$: (\mathsf{if}\, C\, u\, v\, t)[\sigma] = \mathsf{if}\, (C[\sigma^\uparrow])\, (u[\sigma])\, (v[\sigma])\, (t[\sigma])$

A model supports a QIIT: algebra

$$
\begin{aligned}
\mathsf{Con} &: \mathsf{Set} \\
\mathsf{Ty} &: \mathsf{Con} \to \mathsf{Set} \\
\mathsf{Sub} &: \mathsf{Con} \to \mathsf{Con} \to \mathsf{Set} \\
\mathsf{Tm} &: (\Gamma : \mathsf{Con}) \to \mathsf{Ty}\ \Gamma \to \mathsf{Set} \\
\bullet &: \mathsf{Con} \\
- \triangleright - &: (\Gamma : \mathsf{Con}) \to \mathsf{Ty}\ \Gamma \to \mathsf{Con} \\
\mathsf{U} &: \mathsf{Ty}\ \Gamma \\
\mathsf{El} &: \mathsf{Tm}\ \Gamma\ \mathsf{U} \to \mathsf{Ty}\ \Gamma \\
\Pi &: (a : \mathsf{Tm}\ \Gamma\ \mathsf{U}) \to \\
&\quad \mathsf{Ty}\ (\Gamma \triangleright \mathsf{El}\ a) \to \mathsf{Ty}\ \Gamma \\
&\vdots
\end{aligned}
$$

$$
\begin{aligned}
\mathsf{Con}^{\mathsf{s}} &: \mathsf{Ty}\ \ \bullet \\
\mathsf{Ty}^{\mathsf{s}} &: \mathsf{Ty}\ \ (\bullet \triangleright \mathsf{Con}^{\mathsf{s}}) \\
\mathsf{Sub}^{\mathsf{s}} &: \mathsf{Ty}\ \ (\bullet \triangleright \mathsf{Con}^{\mathsf{s}} \triangleright \mathsf{Con}^{\mathsf{s}}[\epsilon]) \\
\mathsf{Tm}^{\mathsf{s}} &: \mathsf{Ty}\ \ (\bullet \triangleright \mathsf{Con}^{\mathsf{s}} \triangleright \mathsf{Ty}^{\mathsf{s}}) \\
\bullet^{\mathsf{s}} &: \mathsf{Tm}\ \bullet\ \mathsf{Con}^{\mathsf{s}} \\
\triangleright^{\mathsf{s}} &: \mathsf{Tm}\ (\bullet \triangleright \mathsf{Con}^{\mathsf{s}} \triangleright \mathsf{Ty}^{\mathsf{s}})\ (\mathsf{Con}^{\mathsf{s}}[\epsilon]) \\
\mathsf{U}^{\mathsf{s}} &: \mathsf{Tm}\ (\bullet \triangleright \mathsf{Con}^{\mathsf{s}})\ \mathsf{Ty}^{\mathsf{s}} \\
\mathsf{El}^{\mathsf{s}} &: \mathsf{Tm}\ (\bullet \triangleright \mathsf{Con}^{\mathsf{s}} \triangleright \mathsf{Tm}^{\mathsf{s}}[\mathsf{id}, \mathsf{U}^{\mathsf{s}}])\ (\mathsf{Ty}^{\mathsf{s}}[\mathsf{wk}^1]) \\
\Pi^{\mathsf{s}} &: \mathsf{Tm}\ (\bullet \triangleright \mathsf{Con}^{\mathsf{s}} \triangleright \mathsf{Tm}^{\mathsf{s}}[\mathsf{id}, \mathsf{U}^{\mathsf{s}}] \\
&\qquad\qquad \triangleright \mathsf{Ty}^{\mathsf{s}}[\epsilon, \triangleright^{\mathsf{s}}[\mathsf{wk}^1, \mathsf{El}^{\mathsf{s}}]])\ (\mathsf{Ty}^{\mathsf{s}}[\mathsf{wk}^2]) \\
&\vdots
\end{aligned}
$$

A model supports a QIIT: implementation

```
Πᵀ : ∀ Conˢ Tyˢ Tmˢ → ▶ᵀ Conˢ Tyˢ → ∀ Uˢ → Elᵀ Con
Πᵀ Conˢ Tyˢ Tmˢ ▶ˢ Uˢ Elˢ = Tm Πᶜ Πᴿ
  module Πᵀ where
  open Conᵀ
  open Tyᵀ Conˢ
  open Tmᵀ Conˢ Tyˢ
  open ▶ᵀ  Conˢ Tyˢ
  open Uᵀ  Conˢ Tyˢ
  open Elᵀ Conˢ Tyˢ Tmˢ Uˢ
  Π-Γ = Conˢ
  Π-a = Tmˢ [ < Uˢ > ]T
  Π-B = Tyˢ [ ε ,⟨ Ty-Γ ⟩ ▶ˢ [ wk¹ Π-a ,⟨ ▶-A ⟩ El

  Πᶜ  = • ▷ Π-Γ ▷ Π-a ▷ Π-B
  Πᴿ  = Tyˢ [ wk² Π-a Π-B ]T
```

A model supports a QIIT: implementation (arbitrary model)

```
Πᵀ : ∀ Conˢ Tyˢ Tmˢ → ▶ᵀ Conˢ Tyˢ → ∀ Uˢ → Elᵀ Con
Πᵀ Conˢ Tyˢ Tmˢ ▶ˢ Uˢ Elˢ = Tm Πᶜ Πᴿ
  module Πᵀ where
  open Conᵀ
  open Tyᵀ Conˢ
  open Tmᵀ Conˢ Tyˢ
  open ▶ᵀ  Conˢ Tyˢ
  open Uᵀ  Conˢ Tyˢ
  open Elᵀ Conˢ Tyˢ Tmˢ Uˢ
  Π-Γ = Conˢ
  Π-a = Tmˢ [ < Uˢ > ]T
  Π-B = Tyˢ [ ε , coe (ap (Tm _) ([][]T ■ ap (_ [_
                      (▶ˢ [ wk , Elˢ ]t) ]T
  Πᶜ  = • ▷ Π-Γ ▷ Π-a ▷ Π-B
  Πᴿ  = Tyˢ [ wk² ]T
```

A model supports a QIIT: homomorphism

$$
\begin{aligned}
\mathsf{Con}^{\mathsf{M}} &: \mathsf{Con}_1 \to \mathsf{Con}_2 \\
\mathsf{Ty}^{\mathsf{M}} &: \mathsf{Ty}_1\ \Gamma \to \mathsf{Ty}_2\ (\mathsf{Con}^{\mathsf{M}}\ \Gamma) \\
\mathsf{Sub}^{\mathsf{M}} &: \mathsf{Sub}_1\ \Gamma\ \Delta \to \\
&\quad\ \mathsf{Sub}_2\ (\mathsf{Con}^{\mathsf{M}}\ \Gamma)\ (\mathsf{Con}^{\mathsf{M}}\ \Delta) \\
\mathsf{Tm}^{\mathsf{M}} &: \mathsf{Tm}_1\ \Gamma\ A \to \\
&\quad\ \mathsf{Tm}_2\ (\mathsf{Con}^{\mathsf{M}}\ \Gamma)\ (\mathsf{Ty}^{\mathsf{M}}\ A) \\
\bullet^{\mathsf{M}} &: \mathsf{Con}^{\mathsf{M}}\ \bullet_1 = \bullet_2 \\
\rhd^{\mathsf{M}} &: \mathsf{Con}^{\mathsf{M}}\ (\Gamma \rhd_1 A) = \\
&\quad\ (\mathsf{Con}^{\mathsf{M}}\ \Gamma) \rhd_2 (\mathsf{Ty}^{\mathsf{M}}\ A) \\
&\ \vdots
\end{aligned}
$$

$$
\begin{aligned}
\mathsf{Con}^{\mathsf{M}} &: \mathsf{Tm}\ (\bullet \rhd \mathsf{Con}_1)\ (\mathsf{Con}_2\ [\epsilon]) \\
\mathsf{Ty}^{\mathsf{M}} &: \mathsf{Tm}\ (\bullet \rhd \mathsf{Con}_1 \rhd \mathsf{Ty}_1)\ (\mathsf{Ty}_2\ [\epsilon, \mathsf{Con}^{\mathsf{M}}\ [\mathsf{wk}^1]]) \\
\mathsf{Sub}^{\mathsf{M}} &: \mathsf{Tm}\ (\bullet \rhd \mathsf{Con}_1 \rhd \mathsf{Con}_1\ [\epsilon] \rhd \mathsf{Sub}_1) \\
&\qquad (\mathsf{Sub}_2\ [\epsilon, \mathsf{Con}^{\mathsf{M}}\ [\mathsf{wk}^2], \mathsf{Con}^{\mathsf{M}}\ [\epsilon, \mathsf{v}^1]]) \\
\mathsf{Tm}^{\mathsf{M}} &: \mathsf{Tm}\ (\bullet \rhd \mathsf{Con}_1 \rhd \mathsf{Ty}_1 \rhd \mathsf{Tm}_1) \\
&\qquad (\mathsf{Tm}_2\ [\epsilon, \mathsf{Con}^{\mathsf{M}}\ [\mathsf{wk}^2], \mathsf{Ty}^{\mathsf{M}}\ [\mathsf{wk}^1]]) \\
\bullet^{\mathsf{M}} &: \mathsf{Tm}\ \bullet\ (\mathsf{Id}\ (\mathsf{Con}^{\mathsf{M}}\ [\epsilon, \bullet_1])\ \bullet_2) \\
\rhd^{\mathsf{M}} &: \mathsf{Tm}\ (\bullet \rhd \mathsf{Con}_1 \rhd \mathsf{Ty}_1) \\
&\qquad (\mathsf{Id}\ (\mathsf{Con}^{\mathsf{M}}\ [\epsilon, \rhd_1]) \\
&\qquad\quad (\rhd_2\ [\epsilon, \mathsf{Con}^{\mathsf{M}}\ [\mathsf{wk}^1], \mathsf{Ty}^{\mathsf{M}}])) \\
&\ \vdots
\end{aligned}
$$

Plan

1. Quotient inductive-inductive types (QIITs)
2. Specification of QIITs in a model
3. The setoid model
4. Implementation of the universal QIIT in the setoid model
5. Conclusion

The setoid model

```
record Con i : Set (lsuc i) where
  field
    |_|C : Set i
    _C_~_ : |_|C → |_|C → Prop i
    refC : ∀ γ → _C_~_ γ γ
    symC : ∀{γ γ'} → _C_~_ γ γ' → _C_~_ γ' γ
    transC : ∀{γ γ' γ''} → _C_~_ γ γ' → _C_~_ γ' γ'' → _C_~_ γ γ''
  infix 4 |_|C
  infix 5 _C_~_
open Con public

record Tms {i j}(Γ : Con i)(Δ : Con j) : Set (i ⊔ j) where
  field
    |_|s : | Γ |C → | Δ |C
    ~s   : {γ γ' : | Γ |C} → Γ C γ ~ γ' → Δ C (|_|s γ) ~ (|_|s γ')
  infix 4 |_|s
open Tms public

record Ty {i}(Γ : Con i) j : Set (i ⊔ lsuc j) where
  constructor mkTy
  field
    |_|T_   : | Γ |C → Set j
    _T_⊢_~_ : ∀{γ γ'}(p : Γ C γ ~ γ') → |_|T_ γ → |_|T_ γ' → Prop j
    refT    : ∀{γ} α → _T_⊢_~_ (refC Γ γ) α α
    symT    : ∀{γ γ'}{p : Γ C γ ~ γ'}{α : |_|T_ γ}{α' : |_|T_ γ'}
            → _T_⊢_~_ p α α' → _T_⊢_~_ (symC Γ p) α' α
    transT  : ∀{γ γ' γ''}{p : Γ C γ ~ γ'}{q : Γ C γ' ~ γ''}
              {α : |_|T_ γ}{α' : |_|T_ γ'}{α'' : |_|T_ γ''}
            → _T_⊢_~_ p α α' → _T_⊢_~_ q α' α'' → _T_⊢_~_ (transC Γ p q) α α''
    coeT    : {γ γ' : | Γ |C} → Γ C γ ~ γ' → |_|T_ γ → |_|T_ γ'
    cohT    : {γ γ' : | Γ |C}(p : Γ C γ ~ γ')(α : |_|T_ γ) → _T_⊢_~_ p α (coeT p α)
  infix 4 |_|T_
  infix 5 _T_⊢_~_
open Ty public

record Tm {i}(Γ : Con i){j}(A : Ty Γ j) : Set (i ⊔ j) where
  field
    |_|t : (γ : | Γ |C) → | A |T γ
    ~t   : {γ γ' : | Γ |C}(p : Γ C γ ~ γ') → A T p ⊢ (|_|t γ) ~ (|_|t γ')
  infix 4 |_|t
open Tm public
```

Plan

1. Quotient inductive-inductive types (QIITs)
2. Specification of QIITs in a model
3. The setoid model
4. Implementation of the universal QIIT in the setoid model
5. Conclusion

Implementation IIT (i)

```
data Con  : Set₁
data Ty   : Con → Set₁
data Sub  : Con → Con → Set₁
data Tm   : (Γ : Con) → Ty Γ → Set₁

data Con~ : Con → Con → Prop₁
data Ty~  : ∀ {Γ₀ Γ₁} → Con~ Γ₀ Γ₁ → Ty Γ₀ → Ty Γ₁ → Prop₁
data Sub~ : ∀ {Γ₀ Γ₁} → Con~ Γ₀ Γ₁ → ∀ {Δ₀ Δ₁} → Con~ Δ₀ Δ₁ → Sub Γ₀ Δ
data Tm~  : ∀ {Γ₀ Γ₁} (Γ₀₁ : Con~ Γ₀ Γ₁) {A₀ A₁} → Ty~ Γ₀₁ A₀ A₁ → Tm

  ⋮

data Ty where
  U      : ∀ {Γ} → Ty Γ
  El     : ∀ {Γ} → Tm Γ U → Ty Γ
  Π      : ∀ {Γ}(a : Tm Γ U) → Ty (Γ ▶ El a) → Ty Γ
  Id     : ∀ {Γ}(a : Tm Γ U)(u v : Tm Γ (El a)) → Ty Γ
  _[_]T  : ∀ {Γ Δ} → Ty Δ → Sub Γ Δ → Ty Γ
  coerce : ∀ {Γ₀ Γ₁} → Con~ Γ₀ Γ₁ → Ty Γ₀ → Ty Γ₁

  ⋮
```

Implementation IIT (ii)

```
data Ty~ where
  rflT   : ∀ {Γ A} → Ty~ {Γ}{Γ} rflC A A
  symT   : ∀ {Γ₀ Γ₁ Γ₀₁ A₀ A₁} → Ty~ {Γ₀}{Γ₁} Γ₀₁ A₀ A₁ →
  trsT   : ∀ {Γ₀ Γ₁ Γ₂ Γ₀₁ Γ₁₂}{A₀ A₁ A₂} → Ty~ {Γ₀}{Γ₁} Γ
           → Ty~ (trsC Γ₀₁ Γ₁₂) A₀ A₂
  cohT   : ∀ {Γ₀ Γ₁}(Γ₀₁ : Con~ Γ₀ Γ₁)(A : Ty Γ₀) → Ty~ Γ

  U~     : ∀ {Γ₀ Γ₁ Γ₀₁} → Ty~ {Γ₀}{Γ₁} Γ₀₁ U U
  El~    : ∀ {Γ₀ Γ₁ Γ₀₁}{t₀ : Tm Γ₀ U}{t₁ : Tm Γ₁ U} → Tm~

    ⋮

  U[]    : ∀ {Γ Δ}{σ : Sub Γ Δ} → Ty~ rflC (U [ σ ]T) U
  El[]   : ∀ {Γ Δ}{σ : Sub Γ Δ}{a : Tm Δ U}
           → Ty~ rflC (El a [ σ ]T) (El (coerce rflC U[] (

    ⋮
```

Constructors

```
Tyˢ : Tyᵀ Conˢ
| Tyˢ |T       _ ,Σ Γˢ    = S.Ty   Γˢ
_T_⊢_~_ Tyˢ (_ ,p Γˢ~) = S.Ty~ Γˢ~
refT Tyˢ _ = S.rflT
symT Tyˢ   = S.symT
transT Tyˢ = S.trsT
coeT Tyˢ (_ ,p Γ~ˢ) = S.coerce Γ~ˢ
cohT Tyˢ (_ ,p Γ~ˢ) = S.cohT   Γ~ˢ

Uˢ : Uᵀ Conˢ Tyˢ
| Uˢ |t _ = S.U
~t Uˢ   _ = S.U~

Elˢ : Elᵀ Conˢ Tyˢ Tmˢ Uˢ
| Elˢ |t (_ ,Σ Aˢ)  = S.El  Aˢ
~t Elˢ   (_ ,p A~ˢ) = S.El~ A~ˢ

U[]ˢ : U[]ᵀ Conˢ Tyˢ Subˢ []Tˢ Uˢ
| U[]ˢ |t _ = liftp S.U[]
~t U[]ˢ = _

El[]ˢ : El[]ᵀ Conˢ Tyˢ Subˢ Tmˢ []Tˢ []ˢ Uˢ Elˢ U[]ˢ
| El[]ˢ |t _ = liftp S.El[]
~t El[]ˢ = _
```

Recursor

```
recIx  : S.Ix → Prop₁
recCon : S.Con → | Conˢ |T _
recTy  : ∀{Γ}   → S.Ty  Γ   → | Tyˢ  |T _ ,Σ recCon Γ
recSub : ∀{Γ Δ} → S.Sub Γ Δ → | Subˢ |T _ ,Σ recCon Γ ,Σ recCon Δ
recTm  : ∀{Γ A} → S.Tm  Γ A → | Tmˢ  |T _ ,Σ recTy A
rec⌞⌟  : {i : S.Ix} → S.⌞ i ⌟ → recIx i

recIx (S.Con~I Γ₀ Γ₁)          = Conˢ T _ ⊢ recCon Γ₀ ~ recCon Γ₁
recIx (S.Ty~I  Γ₀₁ A₀ A₁)      = Tyˢ T _ ,p rec⌞⌟ Γ₀₁ ⊢ recTy A₀ ~ recTy A₁
recIx (S.Sub~I Γ₀₁  Δ₀₁  σ₀ σ₁) = Subˢ T _ ,p rec⌞⌟ Γ₀₁ ,p rec⌞⌟ Δ₀₁ ⊢ recSub σ₀ ~ recSub σ₁
recIx (S.Tm~I  Γ₀₁ A₀₁ t₀ t₁)   = Tmˢ T _ ,p rec⌞⌟ Γ₀₁ ,p rec⌞⌟ A₀₁ ⊢ recTm t₀ ~ recTm t₁

recCon S.•       = | •ˢ |t _
recCon (_ S.▶ A) = | ▶ˢ |t (_ ,Σ recTy A)

recTy S.U        = | Uˢ  |t _
recTy (S.El a)   = | Elˢ |t (_ ,Σ recTm a)
recTy (S.coerce Γ₀₁ A) = coeT Tyˢ (_ ,p rec⌞⌟ Γ₀₁) (recTy A)

 ⋮

rec⌞⌟ S.•~              = ~t •ˢ _
rec⌞⌟ (S.▶~ _ A₀₁)      = ~t ▶ˢ (_ ,p rec⌞⌟ A₀₁)
rec⌞⌟ S.rflC            = refT   Conˢ _
rec⌞⌟ (S.symC Γ₀₁)      = symT   Conˢ (rec⌞⌟ Γ₀₁)
rec⌞⌟ (S.trsC Γ₀₁ Γ₁₂) = transT Conˢ (rec⌞⌟ Γ₀₁) (rec⌞⌟ Γ₁₂)

rec⌞⌟ S.U~ = ~t Uˢ _
rec⌞⌟ (S.El~ {Γ₀₁ = Γ₀₁} a₀₁) = ~t Elˢ (_ ,p rec⌞⌟ Γ₀₁ ,p rec⌞⌟ a₀₁)
rec⌞⌟ (S.cohT _ _)      = cohT   Tyˢ _ _
rec⌞⌟ S.rflT            = refT   Tyˢ _
rec⌞⌟ (S.symT A₀₁)      = symT   Tyˢ (rec⌞⌟ A₀₁)
rec⌞⌟ (S.trsT A₀₁ A₁₂) = transT Tyˢ (rec⌞⌟ A₀₁) (rec⌞⌟ A₁₂)
rec⌞⌟ S.U[]  = | U[]ˢ  |t _ .unliftp
rec⌞⌟ S.El[] = | El[]ˢ |t _ .unliftp

 ⋮

recConˢ : Conᴹᵀ S.Conˢ Conˢ
| recConˢ |t (_ ,Σ Γ)   = recCon  Γ
~t recConˢ   (_ ,p Γ₀₁) = rec⌞⌟ Γ₀₁

recTyˢ : Tyᴹᵀ S.Conˢ Conˢ recConˢ S.Tyˢ Tyˢ
| recTyˢ |t (_ ,Σ A)   = recTy  A
~t recTyˢ   (_ ,p A₀₁) = rec⌞⌟ A₀₁

 ⋮
```

Uniqueness

```
uniqCon : ∀       Γ → Conˢ T _                             ⊢ | Conᴹ |t (_ ,Σ Γ) ~ recCon Γ
uniqTy  : ∀{Γ}    A → Tyˢ  T _ ,p uniqCon Γ                ⊢ | Tyᴹ  |t (_ ,Σ A) ~ recTy  A
uniqSub : ∀{Γ Δ} σ → Subˢ T _ ,p uniqCon Γ ,p uniqCon Δ ⊢ | Subᴹ |t (_ ,Σ σ) ~ recSub σ
uniqTm  : ∀{Γ A} t → Tmˢ  T _ ,p uniqCon Γ ,p uniqTy  A ⊢ | Tmᴹ  |t (_ ,Σ t) ~ recTm  t

uniqCon S.• = | •ᴹ |t _ .unliftp
uniqCon (Γ S.▶ A)
  = transT Conˢ (| ▶ᴹ |t _ .unliftp) (~t ▶ˢ (_ ,p uniqCon Γ ,p uniqTy A))

uniqTy S.U = transT Tyˢ (| Uᴹ |t _ .unliftp) (~t Uˢ (_ ,p uniqCon _))
uniqTy (S.El a)
  = transT Tyˢ (| Elᴹ |t _ .unliftp)
               (~t Elˢ (_ ,p uniqCon _
                          ,p transT Tmˢ (symT Tmˢ (cohT Tmˢ _ _)) (uniqTm a)))
uniqTy (S.Π a B)
  = transT Tyˢ (| Πᴹ |t _ .unliftp)
               (~t Πˢ (_ ,p uniqCon _
                         ,p transT Tmˢ (symT Tmˢ (cohT Tmˢ _ _)) (uniqTm a)
                         ,p transT Tyˢ (symT Tyˢ (cohT Tyˢ _ _))
                           (transT Tyˢ (symT Tyˢ (cohT Tyˢ _ _)) (uniqTy B))))
uniqTy (S.Id a u v)
  = transT Tyˢ (| Idᴹ |t _ .unliftp)
               (~t Idˢ (_ ,p uniqCon _
                          ,p transT Tmˢ (symT Tmˢ (cohT Tmˢ _ _)) (uniqTm a)
                          ,p transT Tmˢ (symT Tmˢ (cohT Tmˢ _ _)) (uniqTm u)
                          ,p transT Tmˢ (symT Tmˢ (cohT Tmˢ _ _)) (uniqTm v)))
uniqTy (A S.[ σ ]T)
  = transT Tyˢ (| []Tᴹ |t _ .unliftp)
               (~t []Tˢ (_ ,p uniqCon _ ,p uniqCon _ ,p uniqTy A ,p uniqSub σ))
uniqTy (S.coerce Γ₀₁ A)
  = transT Tyˢ (~t Tyᴹ (_ ,p S.symC Γ₀₁ ,p S.symT (S.cohT _ _)))
   (transT Tyˢ (uniqTy A) (cohT Tyˢ _ _))

 ⋮
```

Plan

1. Quotient inductive-inductive types (QIITs)
2. Specification of QIITs in a model
3. The setoid model
4. Implementation of the universal QIIT in the setoid model
5. Conclusion

Conclusion

- A QIIT is a context in the universal QIIT
- All QIITs can be reduced to the universal QIIT
- We showed that the setoid model of type theory supports the universal QIIT
- All of this formalised in Agda, links in the pdf abstract

Further work

- Infinitary QIITs
- Equalities of sorts
- Reduction rules of arbitrary QIITs constructed from the universal QIIT

$$\begin{tikzcd}
\text{any QIIT} \arrow[r] & \text{univ. QIIT} \arrow[r] & \text{impl. IIT}
\end{tikzcd}$$

no β-rules

$$\begin{tikzcd}
\text{any QIIT} \arrow[rr] & & \text{impl. IIT}
\end{tikzcd}$$

β-rules?