Extending Cubical Agda with Internal Parametricity

Antoine Van Muylder[1] Andrea Vezzosi Andreas Nuyts[1]
Dominique Devriese[1]

[1]KU Leuven

TYPES 2022,
20 June 2022

Van Muylder/Nuyts hold a PhD/Postdoctoral Fellowship from the Research Foundation – Flanders (FWO).

Respecting sameness

Natural to ask that terms respect sameness

Respecting sameness

Natural to ask that terms respect sameness

- **Raw DTT.** The subst. operation respects $=$ (judg. equ.)
  $\Gamma \vdash s_i = s_i' : S_i \quad \Gamma, x : S \vdash t : T \quad$ then $\quad \Gamma \vdash \mathrm{t}[s_i] = \mathrm{t}[s_i'] : T[s_i]$

Respecting sameness

Natural to ask that terms respect sameness

- **Raw DTT.** The subst. operation respects $=$ (judg. equ.)
  $\Gamma \vdash s_i = s_i' : S_i \quad \Gamma, x : S \vdash t : T \quad$ then $\quad \Gamma \vdash \mathrm{t}[s_i] = \mathrm{t}[s_i'] : T[s_i]$
  Substitution rule. Valid in all models. Internalize?

Respecting sameness

Natural to ask that terms respect sameness

- **Raw DTT.** The subst. operation respects $=$ (judg. equ.)
  $\Gamma \vdash s_i = s_i' : S_i \quad \Gamma, x : S \vdash t : T \quad$ then $\quad \Gamma \vdash \mathrm{t}[s_i] = \mathrm{t}[s_i'] : T[s_i]$
  Substitution rule. Valid in all models. Internalize?
- **Intensional TT.**

Respecting sameness

Natural to ask that terms respect sameness

- **Raw DTT.** The subst. operation respects $=$ (judg. equ.)
  $\Gamma \vdash s_i = s_i' : S_i \quad \Gamma, x : S \vdash t : T \quad$ then $\quad \Gamma \vdash \mathrm{t}[s_i] = \mathrm{t}[s_i'] : T[s_i]$
  Substitution rule. Valid in all models. Internalize?
- **Intensional TT.**
  1. State it with internal equality $x \equiv_A y$
     $\vdash ? : (s \equiv_S s') \to (t : (x : S) \to Ts) \to t\ s \equiv t\ s'$ (fmap)
     "Function application respects $\equiv$" (prop. equ.)

Respecting sameness

Natural to ask that terms respect sameness

- **Raw DTT.** The subst. operation respects $=$ (judg. equ.)
  $\Gamma \vdash s_i = s_i' : S_i \quad \Gamma, x : S \vdash t : T \quad$ then $\quad \Gamma \vdash \mathrm{t}[s_i] = \mathrm{t}[s_i'] : T[s_i]$
  Substitution rule. Valid in all models. Internalize?
- **Intensional TT.**
  1. State it with internal equality $x \equiv_A y$
     $\vdash ? : (s \equiv_S s') \to (t : (x : S) \to Ts) \to t\ s \equiv t\ s'$ (fmap)
     "Function application respects $\equiv$" (prop. equ.)
  2. Prove it with J primitive.

Respecting sameness

Natural to ask that terms respect sameness

- **Raw DTT.** The subst. operation respects $=$ (judg. equ.)
  $\Gamma \vdash s_i = s_i' : S_i \quad \Gamma, x : S \vdash t : T \quad$ then $\quad \Gamma \vdash \mathrm{t}[s_i] = \mathrm{t}[s_i'] : T[s_i]$
  Substitution rule. Valid in all models. Internalize?
- **Intensional TT.**
  1. State it with internal equality $x \equiv_A y$
     $\vdash ? : (s \equiv_S s') \to (t : (x : S) \to Ts) \to t\ s \equiv t\ s'$ (fmap)
     "Function application respects $\equiv$" (prop. equ.)
  2. Prove it with J primitive.
- **Cubical.** Type-former application respects $\simeq$ (isomorphisms)
  $(F : \mathrm{Mon} \to \mathrm{Grp}) \vdash ? : (A \simeq_{\mathrm{Mon}} B) \to F\,A \simeq_{\mathrm{Grp}} F\,B$
  Valid in some models $\mathrm{psh}(\square_{\mathrm{dM}}), \mathrm{psh}(\square_{\mathrm{c}})$. Internalize?

Respecting sameness

Natural to ask that terms respect sameness

- **Raw DTT.** The subst. operation respects $=$ (judg. equ.)
  $\Gamma \vdash s_i = s_i' : S_i \quad \Gamma, x : S \vdash t : T \quad$ then $\quad \Gamma \vdash \mathrm{t}[s_i] = \mathrm{t}[s_i'] : T[s_i]$
  Substitution rule. Valid in all models. Internalize?
- **Intensional TT.**
  1. State it with internal equality $x \equiv_A y$
     $\vdash ? : (s \equiv_S s') \to (t : (x : S) \to Ts) \to t\ s \equiv t\ s'$ (fmap)
     "Function application respects $\equiv$" (prop. equ.)
  2. Prove it with J primitive.
- **Cubical.** Type-former application respects $\simeq$ (isomorphisms)
  $(F : \mathrm{Mon} \to \mathrm{Grp}) \vdash ? : (A \simeq_{\mathrm{Mon}} B) \to F\,A \simeq_{\mathrm{Grp}} F\,B$
  Valid in some models $\mathrm{psh}(\square_{\mathrm{dM}}), \mathrm{psh}(\square_{\mathrm{c}})$. Internalize?
  1. State it with internal, "synthetic", isomorphisms $p : \mathrm{Path}\ A\ B$
     CTT primitives yield $\circ$, $()^{-1}$, id

Respecting sameness

Natural to ask that terms respect sameness

- **Raw DTT.** The subst. operation respects $=$ (judg. equ.)
  $\Gamma \vdash s_i = s_i' : S_i \quad \Gamma, x : S \vdash t : T \quad$ then $\quad \Gamma \vdash \mathrm{t}[s_i] = \mathrm{t}[s_i'] : T[s_i]$
  Substitution rule. Valid in all models. Internalize?
- **Intensional TT.**
  1. State it with internal equality $x \equiv_A y$
     $\vdash ? : (s \equiv_S s') \to (t : (x : S) \to Ts) \to t\ s \equiv t\ s'$ (fmap)
     "Function application respects $\equiv$" (prop. equ.)
  2. Prove it with J primitive.
- **Cubical.** Type-former application respects $\simeq$ (isomorphisms)
  $(F : \mathrm{Mon} \to \mathrm{Grp}) \vdash ? : (A \simeq_{\mathrm{Mon}} B) \to F\,A \simeq_{\mathrm{Grp}} F\,B$
  Valid in some models $\mathrm{psh}(\square_{\mathrm{dM}}), \mathrm{psh}(\square_{\mathrm{c}})$. Internalize?
  1. State it with internal, "synthetic", isomorphisms $p : \mathrm{Path}\ A\ B$
     CTT primitives yield $\circ$, $()^{-1}$, id
  2. Other primitives (e.g. Glue) ensuring univalence $\mathrm{Path}_{\mathcal{U}}\ A\ B \simeq (A \simeq B)$
     Univalence entails the pcpl.

Respecting relations(1)

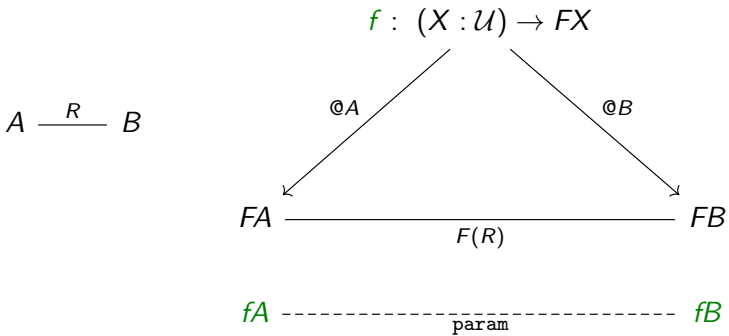
Param. TT. What about *relations*? 2 Observations.

1. Type-formers act on relations

Respecting relations(2)

Based on (1) we formulate parametricity:

2. *Polymorphic functions map type relations to provably related implementations*



The CH system, in Agda

- E. Cavallo, R. Harper achieve "The CH system"
  Internal Parametricity For Cubical Type Theory, LMCS 2021
  Based on Moulin, Bernardy, Coquand primitives

The CH system, in Agda

- E. Cavallo, R. Harper achieve "The CH system"
  Internal Parametricity For Cubical Type Theory, LMCS 2021
  Based on Moulin, Bernardy, Coquand primitives
- To state parametricity: synthetic relations $q : \mathrm{Bridge}\ A\ B$ cf. path

The CH system, in Agda

- E. Cavallo, R. Harper achieve "The CH system"
  Internal Parametricity For Cubical Type Theory, LMCS 2021
  Based on Moulin, Bernardy, Coquand primitives
- To state parametricity: synthetic relations $q : \mathsf{Bridge}\ A\ B$ cf. path
- Other primitives (extent, Gel) grant internal `param`

The CH system, in Agda

- E. Cavallo, R. Harper achieve "The CH system"
  Internal Parametricity For Cubical Type Theory, LMCS 2021
  Based on Moulin, Bernardy, Coquand primitives
- To state parametricity: synthetic relations $q : \mathsf{Bridge}\ A\ B$ cf. path
- Other primitives (extent, Gel) grant internal `param`
- We have implemented CH on top of `agda --cubical`

The CH system, in Agda

- E. Cavallo, R. Harper achieve "The CH system"
  Internal Parametricity For Cubical Type Theory, LMCS 2021
  Based on Moulin, Bernardy, Coquand primitives
- To state parametricity: synthetic relations $q : \mathsf{Bridge}\ A\ B$ cf. path
- Other primitives (extent, Gel) grant internal `param`
- We have implemented CH on top of `agda --cubical`
- We discuss extent, Gel, Bridge in `agda --bridges`
  $\rightsquigarrow$ `param`

Bridges: proofs of relatedness

```
  postulate
    BridgeP : ∀ {ℓ} (A : (x : BI) → Type ℓ) → (a0 : A bi0) → (a1 : A bi1) → Type ℓ
{-
  (A : (x : BI) → Type ℓ) relation ... BI-indexed type.
  BridgeP A a0 a1 = "a0 a1 are related under A"

      Γ , x:BI ⊢ q(x) : A(x)   +judg. endpoints
      ----------------------------------------------INTRO
      Γ ⊢ λ x . q(x) : BridgeP A a₀ a₁

      Γ ⊢ q : BridgeP A a0 a1   Γ ⊢ x : BI     Γ = Γ₁,x,Γ₂  FV(q) ⊆ Γ₁
      ----------------------------------------------ELIM
      Γ ⊢ q x : A x
-}

  refl-bridge : BridgeP (λ _ → A) a₀ a₀
  refl-bridge = λ x → a₀
```

extent: functions preserve bridges

Relational version of funext
Pointwise relatedness of f_0, f_1 implies relatedness

```
-- "related inputs map to related outputs"
Ptwise : (f₀ f₁ : A → B) → Type _
Ptwise f₀ f₁ = ∀ (a₀ a₁ : A) (aa : BridgeP (λ _ → A) a₀ a₁) → BridgeP (λ _ → B) (f₀ a₀) (f₁ a₁)

-- extent : Ptwise → Bridge
bridge-funext : (f₀ f₁ : A → B) → Ptwise f₀ f₁ ≃ BridgeP (λ _ → A → B) f₀ f₁
bridge-funext f₀ f₁ = isoToEquiv (iso
            (λ ptwise → λ x a → primExtent f₀ f₁ ptwise x a)
            (λ bdg a0 a1 aa x → bdg x (aa x))
            (λ bdg → {! ≈ extent-β !} )
            (λ ptwise → refl))

  -- Sometimes a=a'(x). Capture of a' ill defined if x not fresh.
  -- ⊢ primExtent f₀ f₁ ptwise x (a' x) = ptwise (a' bi0) (a' bi1) (λ j → a' j) x
```

Gel: bridges at Type are relations

```
-- Gel : (A₀ → A₁ → Type)  →  Bridge (λ _ → Type) A₀ B₀
relativity : (A₀ → A₁ → Type ℓ) ≃ BridgeP (λ x → Type ℓ) A₀ A₁
relativity = isoToEquiv (iso
  (λ R → λ x → primGel A₀ A₁ R x)
  (λ B → λ a0 a1 → BridgeP (λ x → B x) a0 a1 )
  (λ B →  {! not trivial. BridgeP at ≃ !} )
  (λ R →  {!...!} ))
```

Modular `param`

```
-- polym. functions map related types to related impl.
param-v0 : (F : Type ℓ → Type ℓ) (f : (X : Type ℓ) → F X) →
  (R : BridgeP (λ _ → Type ℓ) A₀ A₁) →
  BridgeP (λ x → F (R x)) (f A₀) (f A₁)
param-v0 F f = λ R → λ x → f ( R x )
```

Each time we use `param-v0` at specific F with custom F_{rel}, we have to prove

$$\ldots \circ F_{\text{rel}} \circ \ldots \equiv (\lambda R\, \lambda x.\, F(Rx))$$

We package such boilerplate proofs in a notion of *native relator*

Modular `param`

```
-- polym. functions map related types to related impl.
param-v0 : (F : Type ℓ → Type ℓ) (f : (X : Type ℓ) → F X) →
  (R : BridgeP (λ _ → Type ℓ) A₀ A₁) →
  BridgeP (λ x → F (R x)) (f A₀) (f A₁)
param-v0 F f = λ R → λ x → f ( R x )
```

Each time we use `param-v0` at specific F with custom F_{rel}, we have to prove

$$\ldots \circ F_{\text{rel}} \circ \ldots \equiv (\lambda R\, \lambda x.\, F(Rx))$$

We package such boilerplate proofs in a notion of *native relator*
$\Rightarrow$ to use internal parametricity:

- (F, F_{rel}) native relator (e.g. by composition)
- `param` applies to all native relators.

Examples

Our `param` theorem grants one-liner proofs for

- Church encodings $\forall(X : \mathit{Type}).X \to X \to X \quad \simeq \quad \texttt{Bool}$
- (Predicative) System F has a parametric model
- (Not yet) the circle encoding $\forall(X_* : \mathit{Type}_*).\,\Omega(X_*) \to X \quad \simeq \quad S^1$
- Much more?

Examples

Our `param` theorem grants one-liner proofs for

- Church encodings $\forall(X : \mathit{Type}).X \to X \to X \quad \simeq \quad \texttt{Bool}$
- (Predicative) System F has a parametric model
- (Not yet) the circle encoding $\forall(X_* : \mathit{Type}_*).\,\Omega(X_*) \to X \quad \simeq \quad S^1$
- Much more?

Thank you

https://github.com/antoinevanmuylder/bridgy-lib