Parametric Dependent Function Types

Andreas Abel

Department of Computer Science
Ludwig-Maximilians-University Munich

Dependently Typed Programming
Satellite Workshop at FLoC 2010, Edinburgh, UK
9 July 2010

Parametric Function Types

- Generalize notion of *type argument* to *parametric argument*.
- Parametric arguments:
  1. Are only present to enable type checking.
  2. Have no computational significance (cannot be matched on).
  3. Can be skipped in equality checking.
  4. Can be erased during program extraction.
- User-controlled extraction.
- Foundations: Miquel (2001), Barras/Bernardo (2008)
- Automatic extraction: Letouzey (2002), Brady/McBride/McKinna (2003)

Church-Style Type Theory (Agda,Coq)

$$\frac{\mathsf{map} : (A\,B : \mathsf{Set}) \to (A \to B) \to \mathsf{List}\,A \to \mathsf{List}\,B \qquad \mathsf{Nat} : \mathsf{Set} \qquad \mathsf{Bool} : \mathsf{Set} \qquad f : \mathsf{Nat} \to \mathsf{Bool}}{\mathsf{map}\,\mathsf{Nat}\,\mathsf{Bool}\,f : \mathsf{List}\,\mathsf{Nat} \to \mathsf{List}\,\mathsf{Bool}}$$

$$\frac{A : \mathsf{Set}}{\mathsf{nil}\,A : \mathsf{Vec}\,A\,\mathsf{zero}} \qquad \frac{A : \mathsf{Set} \qquad n : \mathsf{Nat} \qquad a : A \qquad as : \mathsf{Vec}\,A\,n}{\mathsf{cons}\,A\,n\,a\,as : \mathsf{Vec}\,A\,(\mathsf{succ}\,n)}$$

$$\frac{A : \mathsf{Set} \qquad P : A \to \mathsf{Set} \qquad a : A \qquad p : P\,a}{\mathsf{exI}\,A\,P\,a\,p : \exists x{:}A.\,P\,x}$$

$$\frac{A : \mathsf{Set} \qquad P : A \to \mathsf{Set} \qquad a : A \qquad p : P\,a}{\mathsf{subI}\,A\,P\,a\,p : \{x{:}A \mid P\,x\}}$$

Dependent Function Type

$$\frac{\Gamma \vdash A : \mathsf{Set} \qquad \Gamma, (x\!:\!A) \vdash B : \mathsf{Set}}{\Gamma \vdash (x\!:\!A) \to B : \mathsf{Set}}$$

$$\frac{(x\!:\!A) \in \Gamma}{\Gamma \vdash x : A} \qquad \frac{\Gamma, (x\!:\!A) \vdash t : B}{\Gamma \vdash \lambda x\!:\!A.\, t : (x\!:\!A) \to B}$$

$$\frac{\Gamma \vdash r : (x\!:\!A) \to B \qquad \Gamma \vdash s : A}{\Gamma \vdash r\, s : B[s/x]}$$

$$\frac{\Gamma \vdash t : A \qquad \Gamma \vdash A = B : \mathsf{Set}}{\Gamma \vdash t : B}$$

Curry-Style (Miquel 2001)

$$\frac{\mathsf{map} : [A\,B : \mathsf{Set}] \to (A \to B) \to \mathsf{List}\,A \to \mathsf{List}\,B \qquad \mathsf{Nat} : \mathsf{Set} \quad \mathsf{Bool} : \mathsf{Set} \quad f : \mathsf{Nat} \to \mathsf{Bool}}{\mathsf{map} \quad f : \mathsf{List}\,\mathsf{Nat} \to \mathsf{List}\,\mathsf{Bool}}$$

$$\frac{A : \mathsf{Set}}{\mathsf{nil} \quad : \mathsf{Vec}\,A\,\mathsf{zero}} \qquad \frac{A : \mathsf{Set} \quad n : \mathsf{Nat} \quad a : A \quad as : \mathsf{Vec}\,A\,n}{\mathsf{cons} \quad a\,as : \mathsf{Vec}\,A\,(\mathsf{succ}\,n)}$$

$$\frac{A : \mathsf{Set} \quad P : A \to \mathsf{Set} \quad a : A \quad p : P\,a}{\mathsf{exI} \quad p : \exists x{:}A.\,P\,x}$$

$$\frac{A : \mathsf{Set} \quad P : A \to \mathsf{Set} \quad a : A \quad p : P\,a}{\mathsf{subI} \quad a \quad : \{x{:}A \mid P\,x\}}$$

Polymorphic Function Type

$$\frac{\Gamma \vdash A : \mathsf{Set} \qquad \Gamma, (x\!:\!A) \vdash B : \mathsf{Set}}{\Gamma \vdash [x\!:\!A] \to B : \mathsf{Set}}$$

$$\frac{(x\!:\!A) \in \Gamma}{\Gamma \vdash x : A} \qquad \frac{\Gamma, (x\!:\!A) \vdash t : B}{\Gamma \vdash t : [x\!:\!A] \to B}\ x \notin \mathsf{FV}(t)$$

$$\frac{\Gamma \vdash r : [x\!:\!A] \to B \qquad \Gamma \vdash s : A}{\Gamma \vdash r : B[s/x]}$$

$$\frac{\Gamma \vdash t : A \qquad \Gamma \vdash B : \mathsf{Set}}{\Gamma \vdash t : B}\ A =_{\beta\eta} B$$

Erasure Annotations (Barras/Bernardo 2008)

$$\frac{\mathsf{map} : [A\,B : \mathsf{Set}] \to (A \to B) \to \mathsf{List}\,A \to \mathsf{List}\,B \quad \mathsf{Nat} : \mathsf{Set} \quad \mathsf{Bool} : \mathsf{Set} \quad f : \mathsf{Nat} \to \mathsf{Bool}}{\mathsf{map}\,[\mathsf{Nat}]\,[\mathsf{Bool}]\,f : \mathsf{List}\,\mathsf{Nat} \to \mathsf{List}\,\mathsf{Bool}}$$

$$\frac{A : \mathsf{Set}}{\mathsf{nil}\,[A] : \mathsf{Vec}\,A\,\mathsf{zero}} \qquad \frac{A : \mathsf{Set} \quad n : \mathsf{Nat} \quad a : A \quad as : \mathsf{Vec}\,A\,n}{\mathsf{cons}\,[A]\,[n]\,a\,as : \mathsf{Vec}\,A\,(\mathsf{succ}\,n)}$$

$$\frac{A : \mathsf{Set} \quad P : A \to \mathsf{Set} \quad a : A \quad p : P\,a}{\mathsf{exI}\,[A]\,[P]\,[a]\,p : \exists x{:}A.\,P\,x}$$

$$\frac{A : \mathsf{Set} \quad P : A \to \mathsf{Set} \quad a : A \quad p : P\,a}{\mathsf{subI}\,[A]\,[P]\,a\,[p] : \{x{:}A \mid P\,x\}}$$

Church-Style Polymorphic Function Type

$$\frac{\Gamma \vdash A : \mathsf{Set} \qquad \Gamma, (x\!:\!A) \vdash B : \mathsf{Set}}{\Gamma \vdash [x\!:\!A] \to B : \mathsf{Set}}$$

$$\frac{(x\!:\!A) \in \Gamma}{\Gamma \vdash x : A} \qquad \frac{\Gamma, (x\!:\!A) \vdash t : B}{\Gamma \vdash [\lambda x\!:\!A]t : [x\!:\!A] \to B}\; x \notin \mathsf{FV}(t^*)$$

$$\frac{\Gamma \vdash r : [x\!:\!A] \to B \qquad \Gamma \vdash s : A}{\Gamma \vdash r\,[s] : B[s/x]}$$

$$\frac{\Gamma \vdash t : A \qquad \Gamma \vdash B : \mathsf{Set}}{\Gamma \vdash t : B}\; A^* =_{\beta\eta} B^*$$

Implicit Erasure (MiniAgda)

$$\frac{\mathsf{map} : [A\, B : \mathsf{Set}] \to (A \to B) \to \mathsf{List}\, A \to \mathsf{List}\, B \qquad \mathsf{Nat} : \mathsf{Set} \qquad \mathsf{Bool} : \mathsf{Set} \qquad f : \mathsf{Nat} \to \mathsf{Bool}}{\mathsf{map}\, \mathsf{Nat}\, \mathsf{Bool}\, f : \mathsf{List}\, \mathsf{Nat} \to \mathsf{List}\, \mathsf{Bool}}$$

$$\frac{A : \mathsf{Set}}{\mathsf{nil}\, A : \mathsf{Vec}\, A\, \mathsf{zero}} \qquad \frac{A : \mathsf{Set} \qquad n : \mathsf{Nat} \qquad a : A \qquad as : \mathsf{Vec}\, A\, n}{\mathsf{cons}\, A\, n\, a\, as : \mathsf{Vec}\, A\, (\mathsf{succ}\, n)}$$

$$\frac{A : \mathsf{Set} \qquad P : A \to \mathsf{Set} \qquad a : A \qquad p : P\, a}{\mathsf{exI}\, A\, P\, a\, p : \exists x{:}A.\, P\, x}$$

$$\frac{A : \mathsf{Set} \qquad P : A \to \mathsf{Set} \qquad a : A \qquad p : P\, a}{\mathsf{subI}\, A\, P\, a\, p : \{x{:}A \mid P\, x\}}$$

Parametric Function Type

$$\frac{\Gamma \vdash A : \mathsf{Set} \qquad \Gamma, (x\!:\!A) \vdash B : \mathsf{Set}}{\Gamma \vdash [x\!:\!A] \to B : \mathsf{Set}}$$

$$\frac{(x\!:\!A) \in \Gamma}{\Gamma \vdash x : A} \qquad \frac{\Gamma, [x\!:\!A] \vdash t : B}{\Gamma \vdash \lambda x t : [x\!:\!A] \to B}$$

$$\frac{\Gamma \vdash r : [x\!:\!A] \to B \qquad \Gamma^{\oplus} \vdash s : A}{\Gamma \vdash r\, s : B[s/x]}$$

$$\frac{\Gamma \vdash t : A \qquad (\Gamma^{\oplus} \vdash A : \mathsf{Set}) = (\Gamma^{\oplus} \vdash B : \mathsf{Set})}{\Gamma \vdash t : B}$$

Resurrection $(-)^{\oplus}$ (Pfenning 2001) turns assumptions $[x\!:\!A]$ into $(x\!:\!A)$.

Heterogeneous Definitional Equality

$$\frac{\Gamma, [x\!:\!A] \vdash t : B \qquad \Gamma^{\oplus} \vdash s : A}{(\Gamma \vdash (\lambda x t)\, s : B[s/x]) = (\Gamma \vdash t[s/x] : B[s/x])}$$

$$\frac{(\Gamma, [x\!:\!A] \vdash t : B) = (\Gamma', [x\!:\!A'] \vdash t' : B')}{(\Gamma \vdash \lambda x t : [x\!:\!A] \to B) = (\Gamma' \vdash \lambda x t' : [x\!:\!A'] \to B')}$$

$$\frac{\begin{array}{c}(\Gamma \vdash r : [x\!:\!A] \to B) = (\Gamma' \vdash r' : [x\!:\!A'] \to B') \\ \Gamma^{\oplus} \vdash s : A \qquad \Gamma'^{\oplus} \vdash s' : A'\end{array}}{(\Gamma \vdash r\, s : B[s/x]) = (\Gamma' \vdash r'\, s' : B'[s'/x])}$$

$$\frac{\begin{array}{c}(\Gamma \vdash t : A) = (\Gamma' \vdash t' : A') \\ (\Gamma \vdash A : \mathit{Set}) = (\Gamma \vdash B : \mathrm{Set}) \qquad (\Gamma' \vdash A' : \mathit{Set}) = (\Gamma' \vdash B' : \mathrm{Set})\end{array}}{(\Gamma \vdash t : B) = (\Gamma' \vdash t' : B')}$$

Conclusions

- MiniAgda: implementation of parametric functions (work in progress).
- Future work:
  1. Implement extraction to Haskell.
  2. Work out semantics of heterogeneous definitional equality.
  3. Port to Agda.
  4. Proof search / aggressive unification for erased meta-variables.