Irrelevance in Type Theory

Andreas Abel

Department of Computer Science
Ludwig-Maximilians-University Munich

Logic and Computation Seminar
School of Computer Science
McGill University, Montreal, Canada
21 March 2011

Proof Irrelevance in Agda

```
-- div' n m  computes  ceil(n / (m + 1))
div'  : {i : Size} → Nat i → Nat → Nat i
div'  zero    m = zero
div' (suc n) m = suc (div' (n - m) m)

-- div n m computes floor(n / m)
div : Nat → (m : Nat) → .(zero < m) → Nat
div n  zero    ()
div n (suc m) p  = div' (n - m) m
```

- The outcome of division `div` n m p does not depend on proof $p : 0 < m$.
- p only serves to eliminate impossible cases.
- At runtime p can be absent.

Proofs in Data Structures

```
Bounds = List A
_below_ : A → Bounds → Set
_above_ : A → Bounds → Set

data Tree (β γ : Bounds) : Set where
  leaf : Tree β γ
  node : (a : A) → .(a above β) → .(a below γ) →
         Tree β (a :: γ) → Tree (a :: β) γ → Tree β γ
```

- Proofs are irrelevant for equality: $\mathtt{node}\ a\ p\ q\ l\ r = \mathtt{node}\ a\ p'\ q'\ l\ r$.
- Agda (as opposed to Coq) has proof irrelevance at compile-time.

Irrelevant Function type $.A \to B$

Slogan: A term is a proof if computation does not depend on it.

- Conjunctions can be eliminated:

```
split : (.A → .B → C) → .(A × B) → C
split k (a , b) = k a b
```

- Matching can be translated away:

```
split : (.A → .B → C) → .(A × B) → C
split k p = k (fst p) (snd p)
```

- Disjunctions cannot be eliminated:

```
case : (.A → C) → (.B → C) → .(A ⊎ B) → C
case f g (inl a) = f a
case f g (inr b) = g b
```

Ordinary Dependent Function Type

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

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

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

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

Irrelevant Dependent Function Type

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

$$\text{no rule for } .(x:A) \in \Gamma \qquad \frac{\Gamma,\ .(x:A) \vdash t : B}{\Gamma \vdash \lambda x t : .(x:A) \to B}$$

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

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

Algorithmic $\beta\eta$-equality

- Mutual judgements:

$$\Delta \vdash n : T \longleftrightarrow \Delta' \vdash n' : T' \quad \text{infer mode}$$
$$\Delta \vdash u : U \Longleftrightarrow \Delta' \vdash u' : U' \quad \text{check mode}$$

- Ordinary application:

$$\frac{\Delta \vdash n : (x:U) \to T \longleftrightarrow \Delta' \vdash n' : (x:U') \to T' \qquad \Delta \vdash u : U \Longleftrightarrow \Delta' \vdash u' : U'}{\Delta \vdash n\,u : T[u/x] \longleftrightarrow \Delta' \vdash n'\,u' : T'[u'/x]}$$

- Irrelevant application:

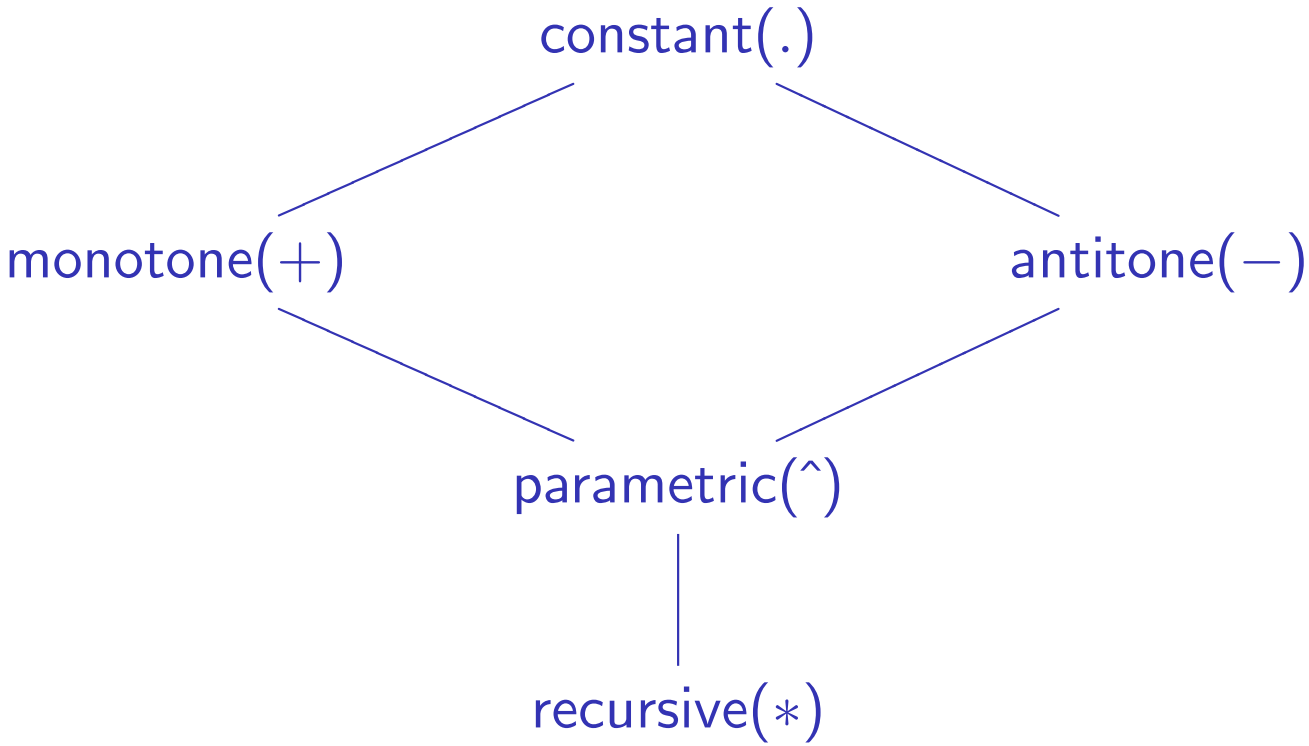
$$\frac{\Delta \vdash n : .(x:U) \to T \longleftrightarrow \Delta' \vdash n' : .(x:U') \to T'}{\Delta \vdash n\,u : T[u/x] \longleftrightarrow \Delta' \vdash n'\,u' : T'[u'/x]}$$

On Paper (FoSSaCS'11)

- Kripke logical relation $\Delta \vdash t : T \mathrel{\overline{\textcircled{S}}} \Delta' \vdash t' : T'$.
- Entails declarative and algorithmic equality.
- Models declarative typing and equality rules.
- Proves consistency, normalization, decidability.

Further Work: Combine with Polarities

- Function lattice:



- p-variant dependent function type $p(x:A) \longrightarrow B$.
- Implemented in MiniAgda.

p-Variant Dependent Function Type

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

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

$$\frac{\Gamma \vdash r : p(x:A) \to B \qquad p^{-1}\Gamma \vdash s : A}{\Gamma \vdash r\, s : B[s/x]}$$

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

Related Work

1. Proof Irrelevance in LF (Pfenning)
2. Bracket Types (Awodey, Bauer)
3. Uniform quantification (Berger, Schwichtenberg)
4. Program extraction in Coq (Letouzey)
5. Implicit Calculus of Constructions (Miquel, Barras, Bernardo)
6. Erasure Pure Type Systems (Mishra-Linger, Sheard)
7. Lightning (Brady, McBride)