On Shape Irrelevance and Polymorphism in Type Theory

Andreas Abel
(joint work in progress with Gabriel Scherer)

Department of Computer Science
Ludwig-Maximilians-University Munich

14th Agda Intensive Meeting (AIM XIV)
Shonan Village Center
near Tokyo, Japan
9 September 2011

Type systems for computational irrelevance

- Separate computationally relevant parts from "administrative" (computationally irrelevant) parts.
- Used for:
  1. Extracting programs
  2. Strengthening equational theory (ignore irrelevant parts during equality checking)
  3. Pruning terms, reducing memory footprint
- Kinds of irrelevance:
  1. Proof arguments (like $x \neq 0$ in division)
  2. Type arguments (like A in $\mathsf{append}\, A\, xs\, ys$)
  3. "Forced" arguments (like n in $\mathsf{vcons}\, A\, n\, x\, xs$)
  4. Termination evidence: universe levels and sizes

The Agda Pipeline

Frontend	$\vdots$	Parsing Scope checking
	Abstract syntax	
Type checker	$\Downarrow$	Type reconstruction Internal erasure Termination inference
	Internal syntax	
Backend	$\vdots$	Extraction Optimizations Code generation

Internal Erasure vs. Extraction

	Erasure	Extraction
Free vars	yes (terms)	no (programs)
Evaluation	under binder	funs = black boxes
Equality	intensional ($\beta\eta$), decid.	observational, undecid.
Empty types	hypothetical inhabitants	no closed inhabitants
Identity proofs	relevant	irrelevant (= refl)
Accessibility proofs	relevant (termination!)	irrelevant

Extraction can erase all inhabitants of propositional types, i. e., with at most one closed inhabitant.

ICC* and EPTS

- Barras and Bernardo (FoSSaCS 2008) and Sheard and Mishra-Linger (FoSSaCS 2008)

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

no rule for $[x\!:\!A] \in \Gamma$ $$\frac{\Gamma,\ [x\!:\!A] \vdash t : B}{\Gamma \vdash \lambda x t : [x\!:\!A] \to B}$$

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

- *Resurrection* $(-)^{\oplus}$ (Pfenning 2001) turns irrelevant assumptions $[x\!:\!A]$ into relevant ones $(x\!:\!A)$.
- Irrelevant function argument can be relevant in function codomain.

$$\lambda A \lambda x.x : [A : \mathsf{Set}] \to A \to A$$

Equality in ICC^*

- Equality in ICC^* is untyped $\beta\eta$ after erasure.
- Does not scale to typed η-equality with unit type $\top$ in the presence of large eliminations.
- Given $h : [A : \mathsf{Set}] \to (A \to A) \to \mathsf{Bool}$, then?

$$h\ (\mathbb{N} \to \mathbb{N})\ (\lambda x \lambda y.\, x\, y) = h\ \top\ (\lambda x.\, ()) : \mathsf{Bool}$$

- Algorithm would check heterogeneous $\lambda y.\, x\, y : \mathbb{N} \to \mathbb{N} = () : \top$?
- But then $t : A = () : \top = t' : A'$, inconsistent!

Irrelevance in Agda 2.2.10

- Irrelevant function arguments need to be irrelevant in codomain.

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

- Type $.(A : \mathsf{Set}) \to A \to A$ ill-formed.
- Equality is typed $\beta\eta$, ignoring irrelevant arguments.
- No need for heterogeneous equality.

Shape-directed η-equality

- η-laws are applied according to the type shape: function type, record type (e.g., unit type), other type.
- Exact type not necessary.
- Exploited by Harper/Pfenning's simply-typed equality check for LF.
- More subtle with large eliminations!

$$\begin{array}{lcl} T & : & \mathsf{Bool} \to \mathsf{Set} \\ T\ \mathsf{true} & = & \top \\ T\ \mathsf{false} & = & \mathbb{N} \to \mathbb{N} \end{array}$$

- Shape of $T\ u$ depends on value of u.

Shape-Irrelevance

- A function is shape-irrelevant if the value of the argument does not influence the (deep) shape of the result.
- Prime example: Data type constructors.

$$
\begin{array}{lcl}
\mathsf{List} & : & \hat{}(A : \mathsf{Set}) \to \mathsf{Set} \\
\mathsf{Vec} & : & \hat{}(A : \mathsf{Set}) \to \hat{}(n : \mathbb{N}) \to \mathsf{Set} \\
\Sigma & : & \hat{}(A : \mathsf{Set}) \to \hat{}(B : A \to \mathsf{Set}) \to \mathsf{Set}
\end{array}
$$

- Parameters in data constructors and projections are irrelevant!

$$
\begin{array}{lcl}
\mathsf{nil} & : & .(A : \mathsf{Set}) \to \mathsf{List}\, A \\
\mathsf{vcons} & : & .(A : \mathsf{Set}) \to .(n : \mathbb{N}) \to A \to \mathsf{Vec}\, A\, n \to \mathsf{Vec}\, A\, (\mathsf{suc}\, n) \\
\_,\_ & : & .(A : \mathsf{Set}) \to .(B : A \to \mathsf{Set}) \to (a : A) \to B\, a \to \Sigma\, A\, B
\end{array}
$$

Typing rules for irrelevance and shape-irrelevance

- Function classifier $p \;::=\; . \quad$ irrelevant function
  $\mid\; \hat{}\quad$ shape-irrelevant function
  $\mid\; \epsilon\quad$ ordinary (relevant) function
- "Going types" $p^{\oplus}$ turns "." into "^".

$$\frac{\Gamma \vdash A : \mathsf{Set} \qquad \Gamma, p^{\oplus}(x\!:\!A) \vdash B : \mathsf{Set}}{\Gamma \vdash p(x\!:\!A) \to B : \mathsf{Set}} \qquad \frac{\Gamma, p(x\!:\!A) \vdash t : B}{\Gamma \vdash \lambda x t : p(x\!:\!A) \to B}$$

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

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

- Γ^{p} makes assumptions available that are at least as relevant as p.

Examples

- Type of $\mathsf{nil} : .(A\!:\!\mathsf{Set}) \to \mathsf{List}\,A$ well-formed

$$\dfrac{\dfrac{\mathsf{List} : \hat{}(A\!:\!\mathsf{Set}) \to \mathsf{Set} \qquad A : \mathsf{Set} \vdash A : \mathsf{Set}}{\hat{}(A\!:\!\mathsf{Set}) \vdash \mathsf{List}\,A}}{\vdash .(A\!:\!\mathsf{Set}) \to \mathsf{List}\,A}$$

- Irrelevantly quantified variables my appear shape-irrelevantly in codomain. Then, List's argument is shape-irrelevant.
- Universe-polymorphic lists $\mathsf{UList} : .(i\!:\!\mathsf{Level}) \to \hat{}(A\!:\!\mathsf{Set}\,i) \to \mathsf{Set}\,i$.

$$\dfrac{\dfrac{\dfrac{\mathsf{Set} : \hat{}(i\!:\!\mathsf{Level}) \to \mathsf{Set}\,(i+1) \qquad i\!:\!\mathsf{Level} \vdash i : \mathsf{Level}}{\hat{}(i\!:\!\mathsf{Level}) \vdash \mathsf{Set}\,i}}{\hat{}(i\!:\!\mathsf{Level}) \vdash \mathsf{Set}\,i \to \mathsf{Set}\,i}}{\vdash .(i\!:\!\mathsf{Level}) \to \mathsf{Set}\,i \to \mathsf{Set}\,i}$$

Shape-directed equality

- Equality judgement $\Gamma \vdash t = t' : A$ relaxed: A is the common shape of t and t'.

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

- Note: $\Gamma, \hat{}(x\!:\!A) \vdash B$, hence $B[u/x]$ and $B[u'/x]$ same shape.

$$\frac{\Gamma \vdash t = t' : \hat{}(x\!:\!A) \to B \qquad \Gamma^{\hat{}} \vdash u = u' : A}{\Gamma \vdash t\, u = t'\, u' : B[u/x]}$$

Unification

- Unification finds parameters.

$$\frac{\mathsf{nil}\ \_1 \rightrightarrows \mathsf{List}\ \_1 \qquad \dfrac{\_1 = A}{\mathsf{List}\ \_1 = \mathsf{List}\ A}}{\mathsf{nil}\ \_1 \Leftarrow \mathsf{List}\ A}$$

- Irrelevant parameters are not uniquely determined

$$\frac{\mathsf{nil}\ \_1\ \_2 \rightrightarrows \mathsf{UList}\ \_1\ \_2 \qquad \dfrac{\_2 = A : \mathsf{Set}\ i \qquad \text{no eq. for}\ \_1}{\mathsf{UList}\ \_1\ \_2 = \mathsf{UList}\ i\ A}}{\mathsf{nil}\ \_1\ \_2 \Leftarrow \mathsf{UList}\ i\ A}$$

Related Work

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