Normalization by Evaluation for Sized Dependent Types

Andreas Abel[1] Andrea Vezzosi[1] Theo Winterhalter[2]

[1]Department of Computer Science and Engineering
Chalmers and Gothenburg University, Sweden

[2]École Normale Supérieure de Cachan, France

23rd International Conference on Types for Proofs and Programs
TYPES 2017
Budapest, Hungary, 30 May 2017

Story I: Stratification of Universes

- $\mathsf{Set} : \mathsf{Set}$ (meaning $\mathsf{Type} : \mathsf{Type}$) inconsistent.
- Stratification $\mathsf{Set}_0 : \mathsf{Set}_1 : \mathsf{Set}_2 : \dots$
- Subtyping $\mathsf{Set}_0 \subset \mathsf{Set}_1 \subset \mathsf{Set}_2 \subset \dots$
- Polymorphism $f : \forall \ell \to (A : \mathsf{Set}_\ell) \to A \to A$.
- Level expressions are not unique: $f\,0\,\mathsf{Nat}\,x$ vs. $f\,1\,Nat\,x$.
- Get in the way of equality: $f\,0\,\mathsf{Nat}\,x = f\,1\,Nat\,x$?

Story II: Stratification of Datatypes

- $\mathsf{fix} : ((\mathsf{Nat} \to C) \to (\mathsf{Nat} \to C)) \to \mathsf{Nat} \to C$ inconsistent
- Stratification $\mathsf{Nat}^0 \subset \mathsf{Nat}^1 \subset \mathsf{Nat}^2 \subset \cdots \subset \mathsf{Nat}^\infty$
  $\mathsf{fix} : (\forall i \to (\mathsf{Nat}^i \to C) \to (\mathsf{Nat}^{i+1} \to C)) \to \forall i \to \mathsf{Nat}^i \to C$
- Polymorphism $t : (\forall i \to \mathsf{Tree}^i \to \mathsf{Tree}^i) \to \forall i \to \mathsf{Tree}^i \to \mathsf{Tree}^\infty$
- Size expressions are not unique.
- Get in the way of equality proofs.

Sizes in the Way of Agda

$$
\begin{array}{l}
\text{data Nat} : (i : \text{Size}) \to \text{Set where} \\
\quad \text{zero} : \forall\ i \to \text{Nat}\ (i + 1) \\
\quad \text{suc}\ \ : \forall\ i \to \text{Nat}\ i \to \text{Nat}\ (i + 1)
\end{array}
$$

$$
\text{pred} : \forall\ i \to \text{Nat}\ i \to \text{Nat}\ i
$$

$$
\begin{array}{l}
\text{monus} : \forall\ i\ (x : \text{Nat}\ i)\ j\ (y : \text{Nat}\ j) \to \text{Nat}\ i \\
\text{monus}\ i\ x\ .(j + 1)\ (\text{zero}\ j)\ \ = x \\
\text{monus}\ i\ x\ .(j + 1)\ (\text{suc}\ j\ y) = \text{monus}\ i\ (\text{pred}\ i\ x)\ j\ y
\end{array}
$$

$$
\begin{array}{l}
\text{monus-diag} : \forall\ i \to (x : \text{Nat}\ i) \to \text{zero}\ \infty \equiv \text{monus}\ i\ x\ i\ x \\
\text{monus-diag}\ .(i + 1)\ (\text{zero}\ i)\ \ = \{!!\} \in \text{zero}\ \infty \equiv \text{zero}\ i \\
\text{monus-diag}\ .(i + 1)\ (\text{suc}\ i\ x) = \{!!\} \in \text{zero}\ \infty \equiv \text{monus}\ (i + 1)\ x\ i\ x
\end{array}
$$

What if the offender was absent?

- Curry-style quantification:

$$\frac{t_1 = t_2 : \forall i \to A\,i}{t_1 = t_2 : A\,a}$$

- Great, but want Church-style syntax for type-checking.

$$\frac{t_1 = t_2 : \forall i \to A\,i}{t_1\langle a_1\rangle = t_2\langle a_2\rangle : A\,a}$$

- Semantically, instantiations a_1 and a_2 are irrelevant.
- "Semantic" typing rule (not *type checking* rule):

$$\frac{t : \forall i \to A\,i}{t\langle a\rangle : A\,b}$$

(When) does this make sense?

- ICC(*) (Miquel, Barras, Bernardo, Sheard, Mishram-Linger): "Always, under erasure"
- What about η for unit type $\star : \mathbf{1}$ with $t = t' : \mathbf{1}$?

$$\frac{f : \forall X \to (X \to X) \to C}{f\langle A \to A\rangle(\lambda x.\, x) = f\langle \mathbf{1}\rangle(\lambda x.\, \star) = f\langle A \to A\rangle(\lambda x.\, \lambda y.\, x\,(x\,y))}$$

- Our restriction: the choice of instantiation may not affect η.
- η is directed by the type shape: function type, unit type, Σ-type, other type.
- Here: X is not irrelevant for the shape of $(X \to X) \to C$.

Dependent irrelevant function types

1. The cautious (Pfenning, LiCS 2001)

$$\frac{\bullet x : A \vdash B : \mathsf{Set}}{\vdash (\bullet x : A) \to B : \mathsf{Set}}$$

2. The daredevil (Miquel, LiCS 2000)

$$\frac{x : A \vdash B : \mathsf{Set}}{\vdash (\bullet x : A) \to B : \mathsf{Set}}$$

3. “lagom” (Abel, AIM 2011)

$$\frac{\bullet\bullet x : A \vdash B : \mathsf{Set}}{\vdash (\bullet x : A) \to B : \mathsf{Set}}$$

Shape-irrelevance

- These are shape-irrelevant in i

$$\bullet\bullet i \vdash \mathsf{Nat}^{\,i}$$
$$\bullet\bullet i \vdash \mathsf{Nat}^{\,i} \to \mathsf{Nat}^{\,i+1}$$
$$\bullet\bullet i \vdash \text{if } b \text{ then } \mathsf{Set}_i \text{ else } \mathsf{Set}_i \to \mathsf{Set}_i$$
$$\bullet\bullet i \vdash (x : \mathsf{Nat}^{\,i}) \to \mathsf{Vec}\, A\, x$$

- These are not shape-irrelevant:

$$\bullet\bullet b : \mathsf{Bool} \nvdash \text{if } b \text{ then } \mathsf{Set}_0 \text{ else } \mathsf{Set}_0 \to \mathsf{Set}_0$$
$$\bullet\bullet X : \mathsf{Set}_0 \nvdash X$$
$$\bullet\bullet X : \mathsf{Set}_0 \nvdash X \to X$$

Defining Shapes (in the Model)

- Base types of the same shape:

$$\mathbf{1} \approx \mathbf{1} \qquad \mathsf{Nat}^{\,i} \approx \mathsf{Nat}^{\,j} \qquad \mathsf{Set}_{\ell} \approx \mathsf{Set}_{\ell'}$$

- Function types:

$$\frac{A_1 \approx A_2 \qquad B_1(a) \approx B_2(a) \text{ for all } a \in A_1}{(x : A_1) \to B_1(x) \approx (x : A_2) \to B_2(x)}$$

- Not symmetric!

$$\textit{template} \precapprox \textit{shape}$$

- No syntactic judgement for *same shape*. :(

Finally, Normalization by Evaluation (NbE)!

- TA-NbE (TA = Type Assignment $\neq$ Thorsten Altenkirch)
- Values a are (extended) weak head normal forms.
- Relations $a \in A$ and $a = a' \in A$ between whnfs.
- Reflecting neutral term u as value $\uparrow^{A} u \in A$:

$$\begin{array}{rcl}(\uparrow^{(x:A)\to B(x)} u)(a) & = & \uparrow^{B(a)}(u \downarrow^{A} a) \\ (\uparrow^{(\bullet x:A)\to B(x)} u)(a) & = & \uparrow^{B(a)}(u\langle\downarrow^{A} a\rangle)\end{array}$$

- Reifying value $a \in A$ as normal term $\downarrow^{A} a$:

$$\begin{array}{rcl}\downarrow^{\mathbf{1}} a & = & \star \\ \downarrow^{(x:A)\to B(x)} f & = & \lambda y. \downarrow^{B(\uparrow^{A} y)} f(\uparrow^{A} y)\end{array}$$

Reflection and Reification

Theorem

Let $A \precsim A_1$ and $A \precsim A_2$.

1. *If u_1 and u_2 are equal neutrals then $\uparrow^{A_1} u_1 = \uparrow^{A_2} u_2 \in A$.*
2. *If $a_1 = a_2 \in A$ then $\downarrow^{A_1} a_1$ and $\downarrow^{A_2} a_2$ are equal normal forms.*

Proof.

- Goal $\uparrow^{(\bullet x : A_1) \to B_1(x)} u_1 = \uparrow^{(\bullet x : A_2) \to B_2(x)} u_2 \in (\bullet x : A) \to B(x)$.
- Assume $a_1 \in A$ and $a_2 \in A$.
- Show $(\uparrow^{(\bullet x : A_1) \to B_1(x)} u_1)(a_1) = (\uparrow^{(\bullet x : A_2) \to B_2(x)} u_2)(a_2) \in B(a_1)$.
- Show $\uparrow^{B_1(a_1)}(u_1 \langle \downarrow^{A_1} a_1 \rangle) = \uparrow^{B_2(a_2)}(u_2 \langle \downarrow^{A_2} a_2 \rangle) \in B(a_1)$.

Since $B_1(x)$ and $B_2(x)$ are shape-irrelevant in x, we apply the induction hypothesis with $B(a_1) \precsim B_1(a_1)$ and $B(a_1) \precsim B_2(a_2)$. □

Decidability

- NbE decides definitional equality.
- Type checking (bidirectional) decidable with rule:

$$\frac{\Gamma \vdash t \rightrightarrows (\bullet x : U) \to T(x) \qquad \bullet^{-1}(\Gamma) \vdash u : U}{\Gamma \vdash t\langle u \rangle \rightrightarrows T(u)}$$

Sizes out of the Way!

```
{-# OPTIONS --experimental-irrelevance #-}

data Nat : ..(i : Size) → Set where
  zero : ∀ .i → Nat (i + 1)
  suc  : ∀ .i → Nat i → Nat (i + 1)

pred : ∀ .i → Nat i → Nat i

monus : ∀ .i (x : Nat i) .j (y : Nat j) → Nat i
monus i x .(j + 1) (zero j)  = x
monus i x .(j + 1) (suc j y) = monus i (pred i x) j y

monus-diag : ∀ .i → (x : Nat i) → zero ∞ ≡ monus i x i x
monus-diag .(i + 1) (zero i)  = refl
monus-diag .(i + 1) (suc i x) = monus-diag i x
```

Conclusions

- Irrelevance modality allows us to ignore sizes where they just help the type checker.

$$\mathrm{suc}\langle \mathit{ignoreMe}\rangle n : \mathrm{Nat}^{\mathit{dontIgnoreMe}}$$

- Codomain of dependent irrelevant function type needs to be shape-irrelevant.
- Better semantics for these modalities? (Vezzosi, Nuyts)
- Beware type-theorist! More modalities are coming your way!

These modalities are horribly complicated, can't we get rid of them?

—Phil Wadler (Leuven, 2017-05-18)