Lower bounds in type theory with ordinals and universes

Ulrik Torben Buchholtz

Münchenwiler Seminar, June 20, 2014

Outline

1. Context
2. Warmup: ε_0
3. Earlier work: Γ_0
4. Earlier work: $\psi(\varepsilon_{\Omega+1})$
5. Lower bound, $\psi(\Gamma_{\Omega+1})$
6. Notes on the Agda formalization

Outline

1. Context
2. Warmup: ε_0
3. Earlier work: Γ_0
4. Earlier work: $\psi(\varepsilon_{\Omega+1})$
5. Lower bound, $\psi(\Gamma_{\Omega+1})$
6. Notes on the Agda formalization

1. Hancock's conjecture and Γ_0 (Feferman 1982).
2. Recent study of unfoldings and the ordinal $\psi(\Gamma_{\Omega+1})$ (the ordinal of $\mathcal{U}(\mathsf{ID}_1)$ among other systems): another Hancock's conjecture.

We work in intensional type theory with Σ (dependent sum) and Π (dependent function) type formers: If A : Type, and if $x : A \vdash B(x)$: Type, then $\Sigma x : A.\, B(x)$ and $\Pi x : A.\, B(x)$ are types.
Allow free variables $X : A \to$ Type to appear in contexts. If $a : A$, then $X\,a$: Type.

Primitive recursive constructions

Our type system shall include at least the primitive recursive types and constructions, by which we mean those closed under

- $X\,a$, for $X : A \to$ Type and $a : A$;
- 0;
- 1;
- $A + B$;
- $\Sigma x : A.\, B(x)$;
- $u =_A v$;
- $(\mu X : A \to \text{Type}, x : A.\, B(X, x))\,a$ for $a : A$;
- $x :$ Bool $\vdash \mathrm{Atom}(x) :$ Type with $\mathrm{Atom}(\mathit{false}) := 0$ and $\mathrm{Atom}(\mathit{true}) := 1$.

and the corresponding terms.

Proofs of well-foundedness

Suppose we have primitive recursive notions,

- OT : Type,
- $x : \mathsf{OT} \vdash C(x)$: Type, and
- $x : \mathsf{OT}, t : C(x) \vdash x[t] : \mathsf{OT}$

representing ordinal notations, cofinality type and direct predecessor. Then we define:

- $X : \mathsf{OT} \to \mathsf{Type} \vdash \mathrm{Prog}(X)$: Type, where

$$\mathrm{Prog}(X) := \Pi x : \mathsf{OT}.\, (\Pi t : C(x).\, X\, x[t]) \to X\, x.$$

- $X : \mathsf{OT} \to \mathsf{Type}, x : \mathsf{OT} \vdash \mathrm{Acc}(X, x)$: Type, where

$$\mathrm{Acc}(X, x) := \mathrm{Prog}(X) \to X\, x.$$

Outline

Warmup: ε_0

Notation system for ε_0: Terms for 0 and $a + \omega^b$. No need to bother calculating Cantor Normal Form just to find cofinality and predecessors:

$$
\begin{aligned}
C(0) &:= 0 \\
C(a + \omega^0) &:= 1 \\
C(a) &:= \text{Nat} \qquad \text{otherwise.}
\end{aligned}
$$

$$
\begin{aligned}
(a + \omega^0)[*] &:= a \\
(a + \omega^{b+\omega^0})[n] &:= (x \mapsto x + \omega^b)^{(n)}\, a \\
(a + \omega^b)[n] &:= a + \omega^{b[n]} \qquad \text{otherwise.}
\end{aligned}
$$

Define a predicate transformer $X : \mathsf{OT} \to \mathsf{Type}, y : \mathsf{OT} \vdash G(X, y) : \mathsf{Type}$ (G for Gentzen) by:

$$G(X, y) := \Pi x : \mathsf{OT}.\, X\, x \to X(x + \omega^y).$$

We can then prove $X : \mathsf{OT} \to \mathsf{Type}, p : \mathrm{Prog}(X) \vdash \mathrm{Prog}(G(X))$ using a recursion on Nat in the iteration case.

This implies that all terms are accessible (but with more and more complicated proofs).

Outline

Peter G. Hancock developed in his dissertation (2000) a formal proof for the well-foundedness for initial segments of Γ_0 in Martin-Löf type theory with a hierarchy of universes closed under Σ and Π.

A universe is here a type U : Type with a decoding $x : U \vdash T(x)$: Type together with constructions on U reflecting the type formers.

Earlier work: Γ_0, cont.

Notation system for Γ_0: Terms for 0, $a + b$ and $\phi(a, b)$. Need not worry about exact representation as long as we can compute the φ-function (binary Veblen function), and can determine cofinality type (0, 1 or Nat) and predecessors.

Relative to a universe we have small predicates p : OT $\to U$. Then we have a code for progressiveness of small predicates p : OT $\to U \vdash \mathrm{prog}(p) : U$ reflecting Prog.

Earlier work: Γ_0, cont.

Hancock uses the notion of *lens*, where a lens for a function f : OT $\to$ OT relative to a universe is predicate transformer F (from small predicates to small predicates) such that

- Πp : OT $\to U.\,\mathrm{prog}(p) \to \mathrm{prog}(F\,p)$, and
- Πp : OT $\to U.\,\mathrm{prog}(p) \to F\,p \subseteq p \circ f$.

For example, G is a lens for $x \mapsto \omega^x$.

The type of all lenses then belongs to the next universe.

We can show that the accessible notations are closed under any operation that possesses a lens.

There is a construction that given a lens for f produces a lens for the derivative of f.

Earlier work: Γ_0, cont.

The key to Hancock's construction is now to show that the property of a that we have a lens for $x \mapsto \varphi(a, x)$ is progressive (relative to the next universe).

Thus, we need an extra universe to handle an extra layer of the φ-function.

Outline

Earlier work: $\psi(\varepsilon_{\Omega+1})$

Anton Setzer showed in his 1993 Munich dissertation how to prove the well-foundedness of the Bachmann-Howard ordinal in type theory with a generalized positive inductive definition. (In fact, he did a lot more: initial segments of $\psi_{\Omega_1}(\Omega_{I+\omega})$ with I the first recursively inaccessible ordinal.)

Definition

The sets of ordinals $B(\alpha)$ and the ordinals $\psi\alpha$ are defined for all ordinals α by induction on α.

- $0 \in B(\alpha)$ and $\Omega \in B(\alpha)$.
- If $\eta, \xi \in B(\alpha)$, then $\eta + \xi \in B(\alpha)$ and $\varphi\eta\xi \in B(\alpha)$.
- If $\xi < \alpha$ and $\xi \in B(\alpha)$, then $\psi\xi \in B(\alpha)$.
- $\psi\alpha := \min\{\xi \in \mathrm{On} : \xi \notin B(\alpha)\}$.

Then the Howard-Bachmann ordinal is denoted by $\psi\varepsilon_{\Omega+1}$.

Earlier work: $\psi(\varepsilon_{\Omega+1})$, cont.

We now add a new type Ord : Type of tree ordinals, inductively generated by constructors:

- $0 : \mathrm{Ord}$,
- $\mathrm{succ} : \mathrm{Ord} \to \mathrm{Ord}$, and
- $\lim : (\mathsf{Nat} \to \mathrm{Ord}) \to \mathrm{Ord}$.

This suffices to define the fibred type of accessible notations below Ω by labelling a tree ordinal with the predecessors of a notation.

Earlier work: $\psi(\varepsilon_{\Omega+1})$, cont.

Now we define the fibred type OT^+ of notations whose ε-number components below Ω are accessible.

With the Gentzen predicate transformer we obtain accessibility of the elements $\omega_n(\Omega+1) := (x \mapsto \omega^x)^{(n)}(\Omega+1)$ of OT^+.

A standard condensation argument then allows us to obtain accessibility of the elements $\psi(\omega_n(\Omega+1))$.

Outline

Lower bound, $\psi(\Gamma_{\Omega+1})$

In a sense, $\psi(\Gamma_{\Omega+1})$ is the combination of Γ_0 and $\psi(\varepsilon_{\Omega+1})$, and it takes a combination of the two approaches to prove well-foundedness of each initial segment in the type theory with Ord as a basic type and a hierarchy of universes.

Lower bound, $\psi(\Gamma_{\Omega+1})$

Because Hancock's lens-construction only depends on being able to compute φ and having standard cofinality and predecessor constructions, we can obtain accessibility of the elements $\eta_n(\Omega+1) := (x \mapsto \varphi x 0)^{(n)}(\Omega+1)$ of OT$^+$.

Thus we have accessibility of the elements $\psi(\eta_n(\Omega+1))$ of OT, and these form a sequence with limit $\psi\Gamma_{\Omega+1}$.

Outline

Notes on the Agda formalization

To formalize the preceeding in Agda, we need a uniform way of stepping from one universe to the next, *i.e.*, we need a stronger principle than is available in the system under scrutiny.

Agda has a hierarchy of universes, but no way of uniformly stepping from one to the next (it does however allow constructions that work for all universe levels at once, but that's not helpful here).

However, Agda allows a stronger form of inductive definitions, namely inductive-recursive definitions, and we can use these to define a next-universe operator.

A Super-Universe in Agda

```
module UniverseOver (OT : Set) (U : Set) (T : U → Set) where

  mutual

    data set : Set where
      u : set
      t : U → set
      π : (d : set) → (el d → set) → set
      σ : (d : set) → (el d → set) → set
      n₀ : set
      n₁ : set
      nat : set
      ord : set
      ot : set

    el : set → Set
    el u = U
    el (t x) = T x
    el (π d p) = (x : el d) → el (p x)
    el (σ d p) = Σ (el d) λ x → el (p x)
    el n₀ = N₀
    el n₁ = N₁
    el nat = Nat
    el ord = Ord
    el ot = OT
```