There is Life in the Universes Beyond ω

ANONYMOUS AUTHOR(S)

The first draft of Martin-Löf's type theory proposed the assumption Type:Type. Subsequently, universe levels have been introduced to avoid the resulting inconsistencies by assuming Type_i:Type_{i+1}. Proof assistants based on type theory support such universe levels to varying degree, but they impose restrictions that can make coding awkward.

Specifically, we consider the ramifications of Agda's approach to handling levels using a denotational semantics of a stratified version of System F as a motivating example. We propose a simple fix that extends Agda's capabilities for handling universe levels parametrically up to ε_0.

Additional Key Words and Phrases: Dependent types, universes, ordinal numbers

ACM Reference Format:
Anonymous Author(s). 2018. There is Life in the Universes Beyond ω. In *Proceedings of Make sure to enter the correct conference title from your rights confirmation email (Conference acronym 'XX).* ACM, New York, NY, USA, 3 pages. https://doi.org/XXXXXXX.XXXXXXX

1 Introduction

The origin of universe levels.

How do universe levels work in Agda?

What becomes awkward with Agda's approach?

How do we propose to fix it?

Contributions.

2 Preliminaries

Agda, Ordinals, IR-Universes

2.1 Ordinals

Ordinal numbers are an important concept in mathematics and computer science. They are closely related to well-ordered sets as any well-ordered set is order-isomorphic to an ordinal. The significance to computer science is that such well-orders can be used for termination proofs.

The best known construction of ordinals is a set-theoretic one due to von Neumann. It starts with the smallest ordinal 0 represented by the empty set $\emptyset$. To construct the successor of an ordinal α, we define $\alpha + 1 := \alpha \cup \{\alpha\}$. This way, we construct $1, 2, 3, \ldots$. Then, we can scoop them all up into the smallest limit ordinal $\omega = \{0, 1, 2, 3, \ldots\}$, which contains 0 and all finite applications of the successor to it. We continue with $\omega + 1 = \omega \cup \{\omega\}$ and carry on until we build the next limit ordinal $\omega \cdot 2$, then $\omega \cdot 3$, and so on. Constructing the limit at this level yields ω^ω and continuing further leads to ω^{ω^ω}, $\omega^{\omega^{\omega^\omega}}$, and so on. The limit of this sequence is ε_0 which is the smallest ordinal that fulfills the equation $\varepsilon_0 = \omega^{\varepsilon_0}$.

In the context of this paper, we plan to use ordinals as universe levels where the well-ordering avoids collapsing levels. The construction of ordinals does not stop at ε_0, but we wish to carve out a particular set of ordinals that fits well in an implementation context. Concretely, we consider ordinals less than ε_0 as they can be represented by binary trees. To see this, recall that every ordinal α can be written in Cantor normal form

$$\alpha = \omega^{\beta_1} + \omega^{\beta_2} + \cdots + \omega^{\beta_n}$$

for some $n \geq 0$ and ordinals β_i such that $\beta_1 \geq \beta_2 \geq \cdots \geq \beta_n$. If $\alpha < \varepsilon_0$, then it can be shown that each exponent satisfies $\beta_i < \alpha$. If we, again, write $\beta_i = \omega^{\gamma_1} + \cdots + \omega^{\gamma_m}$ in Cantor normal form, then clearly $\gamma_j < \beta_i < \alpha$. As the ordering on ordinals is a well-order, we know this decreasing sequence must terminate and we obtain a finite representation for each ordinal less than ε_0.

?] developed Agda formalizations for three equivalent representations for precisely this set of ordinals. They define addition and multiplication of ordinals, prove the principle of transfinite induction, and use that to prove that the represented subset of ordinals is well-ordered.

We build on one of their representations, called MutualOrd, define some additional operations, and prove some properties which are needed in the context of universe levels.[1]

TODO: Insert

- Definition of MutualOrd
- Explanation
- Example(s)
- any additional properties that we had to prove

2.2 Universes in Agda

TODO: typeset in Agda style

Agda contains significant support for a universe hierarchy [**?**]. It provides an abstract datatype LEVEL of universe levels along with constants ZERO, SUC, and MAX that denote the base-level of the hierarchy, the successor, and the maximum of two levels. It provides a level-parametric type SET which obeys the typing SET i : SET (SUC i) Type formation handles the universe levels in the same way as finitely stratified System F [**?**]. That is,

- if $A_1 : Set\ i_1$ and $A_2 : Set\ i_2$, then $A_1 \longrightarrow A_2 : Set(i_1 \sqcup i_2)$;
- if $\alpha : Set\ i$ is a type variable and $A : Set\ j$, then $\forall\alpha.A : Set(suc\ i \sqcup j)$.

To avoid inconsistencies, Agda does not allow pattern matching on the type LEVEL. However, quantification over LEVEL is allowed and results in a type at level ω (if the level-typed variable is used in a significant way). Unfortunately, levels ω and higher are **not** handled in a parametric way in Agda. Rather there are predefined type constants SET$_\omega$, SET$_{\omega 1}$, SET$_{\omega 2}$, and so on. This design can be inconvenient, as most notions in the standard library (e.g., the equality type) are in a level-parametric way, and thus they cannot be used with SET$_\omega$ and higher universe levels.

[1]Technically, the mechanization of **?**] relies on cubical Agda [**?**]. We back-ported the definitions for MutualOrd to standard Agda, as cubical is not needed for this representation.

2.3 Encoding Universes

3 Running Example

As a running example for demonstrating various encodings, we consider different extensions of finitely stratified System F [**?**]. More precisely, we start from an intrinsically typed encoding of types and expressions, then we construct denotational semantics for different encodings and discuss their respective merits.

We choose this system as it is significant, presents non-trivial challenges, and it has been studied in the literature. Our encoding of syntax is inspired by **?**], who develop the syntactic metatheory of System-Fω (without stratification). It has been picked up by **?**], who give denotational and operational semantics for finitely stratified System-F and develop a logical relation for it. **?**] use a similar syntax representation for a finitely stratified version of System-Fω extended with qualified types. They also develop a denotational semantics for their calculus. All these papers come with Agda formalizations.

4 Constructions

5 Related Work

How do other proof assistants (Coq, Lean) handle universes? Cumulativity, Impact of impredicativity

6 Conclusions