

Project Description:

Towards an implementation of guarded Cubical Type Theory

Philipp Stassen

April 30, 2020

Introduction

The goal is to further understand guarded Cubical Type Theory by contributing towards its implementation. This can be achieved both in theory and in practice: By proving meta theoretical results, such as decidability of type-checking and addressing the coherence issues, as well as practically by verifying proofs and programs with GCTT.

Dependent type theories are the foundation for many proof assistants (Agda, Coq,...). Ever since, one focal point of discussion has been the subtle equality: strong extensional equalities versus more restricted intensional equalities, of which the latter relishes the benefits of good computational behaviour. It was only until in 2007 when the new homotopy type theory established a stronger notion of equality, while still offering the merits of its intensional counterparts. Yet the interpretation of such a type theory has proven to be problematic; as the initial simplicial model is not constructive, it cannot serve as base for an implementation. Up until now the simplicial model has not been constructivized.

However, in 2013, Coquand, Huber, Mörtberg and Cohen published their paper about a constructive interpretation of a type theory similar to HoTT in cube categories ([CCHM16]) – using cubes, the obstacles making simplices problematic in constructivisation disappear.

Most notably the cubical Agda proof assistant (still in beta) supports the new constructors. Currently, this implementation is still incomplete. Computations that in theory compute (as the Brunerie numbers in [Bru16]) do not halt.

There are several approaches to enable coinduction for type theories, some of which are also implemented in various proof assistants. To ensure the well-definedness, function definitions need to be verified as *productive*. Aiming towards a more accessible coinduction, the authors of *Programming and reasoning with guarded recursion for coinductive types* ([CBGB15]) introduce the new “*later*” modality and provide denotational semantics, supported by an experimental implementation in Agda – thus guaranteeing productivity via *typings*. Building on that paper, the results were extended to dependent type theories in [BGC⁺16] and eventually to cubical type theory in [BBC⁺19].

Background

Having worked on coherence problems in my master thesis, I am familiar with standard papers like *Syntax and semantics of dependent types* ([Hof97b]) and *Extensional Constructs in Intensional Type Theory* ([Hof97a]). I have basic understanding of the literature mentioned in the introduction; to further deepen my knowledge in the semantics of homotopy type theory, more research will be necessary. Additionally, the homotopic interpretation requires elaborate knowledge about infinity-, cube- and simplex categories.

Work Plan

Depending on the current research status and future research breakthroughs, flexibility in adjusting the work plan is mandatory, hence guaranteeing my most efficient contributions. This said, I will outline the focal points of my work plan in the following:

1. **Study** metatheoretical properties of type theory in general as well as CTT and GCTT in particular. Moreover, analyze the relationship between the constructive cubical set models presented in [BBC⁺19] with the simplicial set model of [BM13].
2. **Compute** using GCTT and CTT, exploring the computational merits of the cubical implementation. Testing and experimenting will deepen my general understanding, for CTT in particular formalizations are already feasible.
3. **Develop** better support for CTT in modern proof assistants, finding solutions to problems related to typechecking, unification, etc, and thus propelling developments also relevant for GCTT.
4. **Establish** decidable type-checking and canonicity for GCTT and start working on a type checker. Furthermore, as suggested in [BBC⁺19], investigate how guarded recursion and higher inductive types interact.
5. **Document** the results and make them available for a broader audience.

Conclusion

My vast experience in type theories and the subtleties of categorical semantics as well as my familiarity with Agda will benefit the progress of this project - and thus contribute towards an implementation of guarded Cubical Type Theory.

References

- [BBC⁺19] Lars Birkedal, Aleš Bizjak, Ranald Clouston, Hans Bugge Grathwohl, Bas Spitters, and Andrea Vezzosi. Guarded cubical type theory. *J. Autom. Reason.*, 63(2):211–253, August 2019.
- [BGC⁺16] Ales Bizjak, Hans Bugge Grathwohl, Ranald Clouston, Rasmus Ejlers Møgelberg, and Lars Birkedal. Guarded dependent type theory with coinductive types. *CoRR*, abs/1601.01586, 2016.
- [BM13] Lars Birkedal and R.E. Møgelberg. Intensional type theory with guarded recursive types qua fixed points on universes. pages 213–222, 06 2013.
- [Bru16] Guillaume Brunerie. On the homotopy groups of spheres in homotopy type theory, 2016.
- [CBGB15] Ranald Clouston, Ales Bizjak, Hans Bugge Grathwohl, and Lars Birkedal. Programming and reasoning with guarded recursion for coinductive types. *CoRR*, abs/1501.02925, 2015.
- [CCHM16] C. Cohen, T. Coquand, S. Huber, and A. Mörtberg. Cubical Type Theory: a constructive interpretation of the univalence axiom. *ArXiv e-prints*, November 2016.
- [Hof97a] Martin Hofmann. *Extensional Constructs in Intensional Type Theory*. Springer Publishing Company, Incorporated, 1st edition, 1997.
- [Hof97b] Martin Hofmann. Syntax and semantics of dependent types. In *Semantics and Logics of Computation*, pages 79–130. Cambridge University Press, 1997.