Finiteness in Cubical Type Theory

Donnacha Oisín Kidney[1] and Gregory Provan[2]

[1] University College Cork o.kidney@cs.ucc.ie
[2] University College Cork g.provan@cs.ucc.ie

Abstract. We study five different notions of finiteness in Cubical Type Theory and prove the relationship between them. In particular we show that any totally ordered Kuratowski finite type is manifestly Bishop finite.
We also prove closure properties for each finite type, and classify them topos-theoretically. For instance we show that the category of decidable Kuratowski finite sets (also called the category of cardinal finite sets) form a Π-pretopos.
We then develop a parallel classification for the countably infinite types, as well as a proof of the countability of $A^\star$ for a countable type A.
We formalise our work in Cubical Agda, and we implement a library for proof search (including combinators for level-polymorphic fully generic currying), and demonstrate how it can be used to both prove properties and synthesise full functions given desired properties.

1 Introduction

1.1 Foreword

In constructive mathematics we are often preoccupied with *why* something is true. Take finiteness, for example. There are a handful of ways to demonstrate some type is finite: we could provide a surjection from another finite type; we could show that any collection of its elements larger than some bound contains duplicates; or that any stream of its elements contain duplicates.

Classically, all of these proofs end up proving the same thing: that our type is finite. Constructively (in Martin-Löf Type Theory [3] at least), however, all three of the statements above construct a different version of finiteness. *How* we show that some type is finite has a significant impact on the type of finiteness we end up dealing with.

Homotopy Type Theory [5] adds another wrinkle to the story. Firstly, in HoTT we cannot assume that every type is a (homotopy) set: this means that the finiteness predicates above can be further split into versions which apply to sets only, and those that apply to all types. Secondly, HoTT gives us a principled and powerful way to construct quotients, allowing us to regain some of the flexibility of classical definitions by "forgetting" the parts of a proof we would be forced to remember in MLTT.

Finally, the other important property of constructive mathematics is that we can actually *compute* with proofs. Cubical Type Theory [1], and its implementation in Cubical Agda [6], realise this property even in the presence of univalence, giving computational content to programs written in HoTT.

1.2 Contributions

In this work we will examine five notions of finiteness in Homotopy Type Theory, the relationships between them, and their topos-theoretic characterisation. We also briefly examine a definition for countable sets, comparing it to the manifestly Bishop finite sets. Our work is formalised in Cubical Agda, where we also develop a library for proof search based on the finiteness predicates.

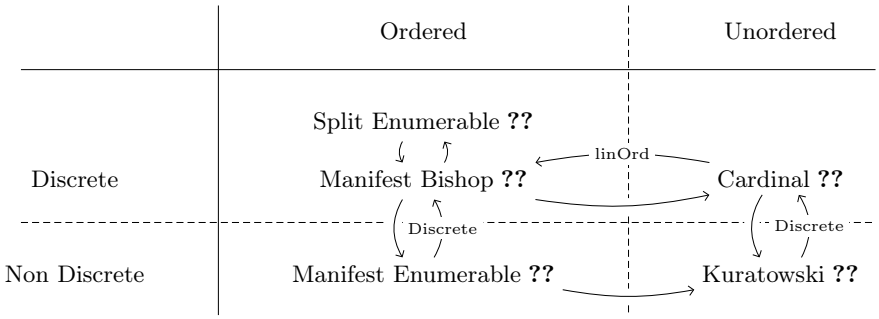

Fig. 1: Classification of the strong definitions of finiteness, according to whether they are discrete (imply decidable equality) and whether they induce a linear order.

The finiteness predicates we are interested in are organised in Figure 1. We will explore two aspects of each predicate: its relation with the other predicates, and its topos-theoretic classification.

When we say "relation" we are referring to the arrows in Figure 1. The proofs, then, amount to a function which inhabits each arrow. Each unlabelled arrow is a weakening: i.e., every manifest Bishop finite set is manifest enumerable. The labelled arrows are strengthening proofs: i.e. every manifest enumerable set *with decidable equality* is manifest Bishop finite.

We will then characterise each predicate as some form of topos. Our proofs follow the structure of [5, Chapters 9, 10] and [4]. This means we will define first the precategory of sets, and then the category of sets, and prove the required closures and limits to get us to a topos. We say "some form of" topos here because of course the category of sets in HoTT do *not*, in fact, form a topos, rather a ΠW-pretopos.

redo these last two paragraphs

After the finite predicates, we will briefly look at the infinite countable types, and classify them in a parallel way to the finite predicates.

All of our work is formalised in Cubical Agda [6]. We will make mention of the few occasions where the formalisation of some proof is of interest, but the main place where we will discuss the code is in the final section, where we implement a library for proof search, based on omniscience and exhaustibility. We also use this section to demonstrate some *computation* using univalence: our proof search library can deal seamlessly with functions, testing them for equality, synthesising them, etc.

1.3 Related Work

1.4 Notation

2 Finiteness Predicates

Definition 1 (Split Enumerable Set)**.**

$$\mathcal{E}!(A) = \Sigma(xs : \mathbf{List}(A)), \Pi(x : A), x \in xs \tag{1}$$

We call the first component of this pair the "support" list, and the second component the "cover" proof. This definition is what is referred to as `Listable` in [2].

3 Relations Between Each Finiteness Definition

4 Topos

5 Countably Infinite Types

6 Search

References

1. Cohen, C., Coquand, T., Huber, S., Mörtberg, A.: Cubical Type Theory: A constructive interpretation of the univalence axiom. arXiv:1611.02108 [cs, math] p. 34 (Nov 2016)
2. Firsov, D., Uustalu, T.: Dependently typed programming with finite sets. In: Proceedings of the 11th ACM SIGPLAN Workshop on Generic Programming - WGP 2015. pp. 33–44. ACM Press, Vancouver, BC, Canada (2015). https://doi.org/10.1145/2808098.2808102
3. Martin-Löf, P.: Intuitionistic Type Theory. Padua (Jun 1980)
4. Rijke, E., Spitters, B.: Sets in homotopy type theory. Mathematical Structures in Computer Science **25**(5), 1172–1202 (Jun 2015). https://doi.org/10.1017/S0960129514000553
5. Univalent Foundations Program, T.: Homotopy Type Theory: Univalent Foundations of Mathematics. https://homotopytypetheory.org/book, Institute for Advanced Study (2013)
6. Vezzosi, A., Mörtberg, A., Abel, A.: Cubical Agda: A Dependently Typed Programming Language with Univalence and Higher Inductive Types. Proc. ACM Program. Lang. **3**(ICFP), 87:1–87:29 (Jul 2019). https://doi.org/10.1145/3341691