Eötvös Loránd University

Faculty of Informatics

Dept. of Programming Language and Compilers

Integers as a Higher Inductive Type

Supervisor:
Ambrus Kaposi Dr.
Associate Professor

Author:
Zoltán Balázs
Computer Science MSc

Budapest, 2024

Contents

Chapter 1

Introduction

1.1 Type Definition

In Homotopy Type Theory, we can define integers in numerous ways, all with their own pros and cons. Following !NAME!, our higher inductive type definition will be the following:

```
data ℤh : Set where
  zero : ℤh
  succ : ℤh → ℤh
  pred : ℤh → ℤh
  sec : (z : ℤh) → pred (succ z) ≡ z
  ret : (z : ℤh) → succ (pred z) ≡ z
  coh : (z : ℤh) → congS succ (sec z) ≡ ret (succ z)
```

With this definition, we have the base element *zero*, as well as *succ* and *pred* as constructors, to increment and decrement the integer value respectively. We then postulate that they are inverse to each other with the inclusion of *sec* (*section*, often seen in the Agda Cubical library as *predSuc*) and *ret* (*retraction*, often seen as *sucPred*). With a *coh* (*coherence*) constructor, we define an equivalence of equivalences. This constructor will introduce most of the challenges when trying to prove anything with our Integer definition. With the inclusion of this last condition, we also say that succ is a half-adjoint equivalence, something that we can use to our advantage in Cubical Agda:

```
isHAℤh : isHAEquiv succ
isHAℤh .isHAEquiv.g    = pred
isHAℤh .isHAEquiv.linv = sec
isHAℤh .isHAEquiv.rinv = ret
isHAℤh .isHAEquiv.com  = coh
```

Using this, we can define a sort of inverse coherence rule, a coherence that inverses the equivalence by applying pred to ret, and checking that it is equal to passing the pred value to sec:

```
hoc : (z : ℤh) → congS pred (ret z) ≡ sec (pred z)
hoc = com-op isHAℤh
```

This will be useful later on, when defining operations on our Integer type.

1.2 Commutative Ring

Our question is the following: Is this definition of integers a correct one, is it a set with decidable equality, and if so, do they form a commutative ring? (While this should be fairly obvious, integers do form a commutative ring, with the higher inductive type definition of integers, this hasn't been formally proven yet.) Moreover, what does it mean for integers to form a commutative ring? We will have to prove the following:

- The set and two binary operations (here: addition and multiplication) form a ring:
  - The set and addition form an abelian group:
    - Addition is associative
    - The identity element exists
    - An inverse element exists
    - Addition is commutative
  - The set is monoid under multiplication:
    - Multiplication is associative
    - The indetity element exists
  - Multiplication distributes over addition

- Multiplication is commutative

Before we prove these, we will dive into proving some other useful properties first.

List of Figures

List of Tables

List of Codes