Homotopy Type Theory

Davi Sales Barreira

August 24, 2022

Contents

List of Definitions

List of Theorems

List of Examples

Notation

Perhaps to my own detriment, I'll deviate from the common notation from HoTT and use something more in line with programming. More specifically, Julia programming.

1 Introduction

Type theory has many relations to the foundations of mathematics, and hence, to logic.

First, in type theory, every *type* corresponds to a *proposition*. The basic *judgment* of type theory is written as $a :: A$, and it represents "A has a proof". The a in $a :: A$ is called a *witness*.

But what are propositions and judgements? Think of a proposition as a statement which can be proven, disproven, assumed, and so on. A judgement is a statement about a proposition, e.g. "proposition A has a proof".

Another important aspect to note is the notion of equalities. The proposition of an equality is similar to the $==$ in programming, meaning, we are "testing" whether two variables are the same. While the judgement is akin to assigning (defining) an equality:

```
x = 10 # judgment
y = 10 # judgment
x == y # proposition
```

Since $x == y$ is a proposition, it also means that, in type theory, this is a type. Thus, we can have $a : (x == y)$.

References