∞-type Café 暑期学校讲座摘要

Jun 13, 2023

Tesla Zhang
teslaz@cmu.edu

Contents

全文使用香蕉空间风格的术语翻译, 部分尚未登录到网站中, 以下是一个参考表.

中文	英语
繁饰	elaboration
立方类型论	cubical type theory
高阶归纳类型	higher inductive type
归纳族	indexed family
泛等公理	univalence axiom
三向类型检查	tridirectional type checking
叠加态求值	glued evaluation
δ-规约	δ-reduction
语义正规化	semantic normalization
逆归纳类型	coinductive type
模式匹配	pattern matching
惰性求值	lazy evaluation
不稳定类型	extension type
δ-可化简式	δ-redex

1. 立方类型论入门

这是「前置基础课程」环节的讲座, 因此会比较入门, 但也需要读者有一定的类型论基础.

1.1. 前置知识

- MLTT – 熟悉 (有对应的讲座)
- HoTT – 了解 (也有对应的讲座)

1.2. 大纲

- 给 HoTT 设计构造模型的基本思路
- 立方类型论的语法要素, 高阶归纳类型的语法
- 立方类型论解决的一般类型论中的问题
- 如何使用立方类型论进行定理证明

HoTT 看起来很美好, 但是它也有显然的问题: MLTT 中的等号一般是作为归纳族的特例实现, 而这个实现不允许额外的实例, 因此泛等公理没有插脚的余地. 除此之外高阶归纳类型也需要对归纳类型的语法进行大幅修改.

立方类型论为了解决这些问题, 重新定义了等号, 并给出了 `J` 的新实现, 而引入的新语法也使得高阶归纳类型的实现变得简单, 且给出了一些 MLTT/HoTT 中的定理的新证明, 以及一些类型的性质更好的定义.

新的等号使得高阶归纳类型可以写在语法里, 泛等公理也可以借助宇宙类型的规则实现.

但很遗憾, 此举添加了过多的表达式, 以及引入了一种性质类似内存泄漏的构造, 且目前没有非常可靠的解决方法.

2. Lamett 项目介绍

这是 Aya 团队的暑期项目. 名字来源于 cooltt 项目, 我们这个没有 cooltt 那么😎所以就叫 lamett 了.

2.1. 前置知识

- 立方类型论 – 熟悉 (可以通过前面的讲座熟悉)
- 实现 MLTT – 了解

2.2. 大纲

- 立方类型论的编程实现思路, 不稳定类型 (extension type)
- 借助 Tarski 宇宙复用类型检查的代码
- 如何将这种思路延拓到一般的编程问题上

为了给 Aya 实现前沿的语言特性, 我们决定在一个简单的语言上先把这些功能做出来. 主要特色是通用的不稳定类型, 和弱 Tarski 宇宙的繁饰, 全都是抄的 cooltt.

3. 实现类型论中的高级特性

3.1. 前置知识

- 实现 MLTT – 非常熟悉

3.2. 大纲

介绍如下功能的代码实现:

- 隐式参数
  - 为什么需要带上作用域?
  - 什么是 `lossy-unification`?
- 三向类型检查
- (逆)归纳类型
  - 如何实现(逆)模式匹配?
  - 如何检查模式匹配是否完全?
- 叠加态求值 (glued evaluation)
  - 什么是 δ-规约?
  - 语义正规化有什么潜在的 (显然的) 性能问题?
  - 如何在惰性求值的元语言中实现叠加态求值

实现 MLTT 有很多教程, 而且语义正规化的算法已经非常标准, 但求解隐式参数的算法却寥寥无几. 尤其是 Agda 编程语言中的求解算法, 能力极强又脱离形式化, 使得它的实现非常难以理解. 而且很多 MLTT 的实现教程使用了 $\mathcal{U} : \mathcal{U}$ 规则以简化问题, 但这个规则在实际数学证明中是不可用的, 而加入分层宇宙后还需要第三个类型检查函数, 这虽然在 Mini-TT 中有所展示但并不算广为人知.

归纳类型本身是个对编程来说非常重要的功能, 可它的实现却没有那么简单. 逆归纳类型由于目前暂时没有「族」的版本所以还不算很复杂, 但它却没有普通的归纳类型一般流行, 导致它的乐趣被雪藏. 这背后有着非常优雅的对偶, 不仅仅是语义上的, 还有类型检查算法上的.

叠加态求值是一种在有 δ-规约的情况下的语义正规化实现技巧, 在元语言支持惰性求值时尤其优雅, 但没有惰性求值时也不是不能用. 大体思路是将 δ-可化简式表示为它的化简结果和未化简形式两者的叠加, 这样就不会对表达式树进行多于一层的展开, 从而尽可能地实现了结构的共享.