Формальные модели программ

Соответствие Карри-Говарда: программа = доказательство утверждения (интуиционистская логика)

Основные свойства типизированного λ-исчисления

Параллельная редукция

Двусторонняя проверка типов

λ-куб Барендрегта: расширения типизированного λ-исчисления. Полиморфное λ-исчисление, λ-исчисление с конструкторами типов, λ-исчисление с зависимыми типами

maxim.krivchikov@gmail.com

Материалы курса: https://maxxk.github.io/formal-models-2015/

Свойство Чёрча-Россера

Takahashi M. Parallel Reductions in λ-Calculus // Information and computation. 1995.

Нормализация

Редекс — терм удаления, для которого есть правило редукции.
Нормальная форма терма относительно редукции — это такой вид, при котором к нему неприменимы правила редукции.
Головная нормальная форма терма — если в головной позиции (корне дерева) не стоит редекс.

Нормализация — свойство формальной системы: если у терма есть нормальная форма, то она единственная.
Сильная нормализация — у всех термов есть единственная нормальная форма (= нет термов, редукция которых не завершается).

Сильная нормализация для просто типизированного λ-исчисления

Просто типизированное λ-исчисление обладает свойством сильной типизации.
Normalization by Evaluation (A. Abel, Habilitation thesis, 2013)

Эквивалентность термов

Обычно определяется следующим образом: A эквивалентен B, если A и B приводятся β-редукцией к идентичному виду, с точностью до корректных (не меняющих) переименований переменных. На индексах де Брёйна последнее замечание неактуально.

η-эквивалентность

Пусть $f : \alpha \to \beta$. Тогда $\lambda(x : \alpha).\, f \cdot x$ интуитивно эквивалентен исходной f, но по указанному выше определению формально это разные термы.

η-эквивалентность включает такое понятие и, в случае просто типизированного λ-исчисления, не нарушает разрешимости эквивалентности типов.

Проверка типов разрешима — если есть алгоритм, который для любого терма определяет, корректно ли он типизирован.
Эквивалентность термов разрешима — если есть алгоритм, который для любой пары термов определяет, эквивалентны ли они при заданных правилах.

Соответствие Карри-Говарда для просто типизированного λ-исчисления

Типы — импликативные суждения. Доказательства — термы, имеющие этот вид. Если ввести тип ложных высказываний как базовый тип False без правил введения и с правилом удаления ex falso, можно ввести и отрицание — «не α» ≡ α → False.

Двусторонняя проверка типов

Алгоритм проверки типов.
Достаточны аннотации типов у констант и переменных абстракции, остальные — можно вывести.
Две взаимно-рекурсивные функции — `check` и `infer`.
`infer` выводит тип терма, `check` проверяет, что терм имеет заданный тип.
Опциональные аннотации.

Не все утверждения удобно представимы в λ-исчислении с простыми типами

В частности, для нумералов Чёрча представимый класс называется «расширенные полиномы» над $\mathbb{N}$:

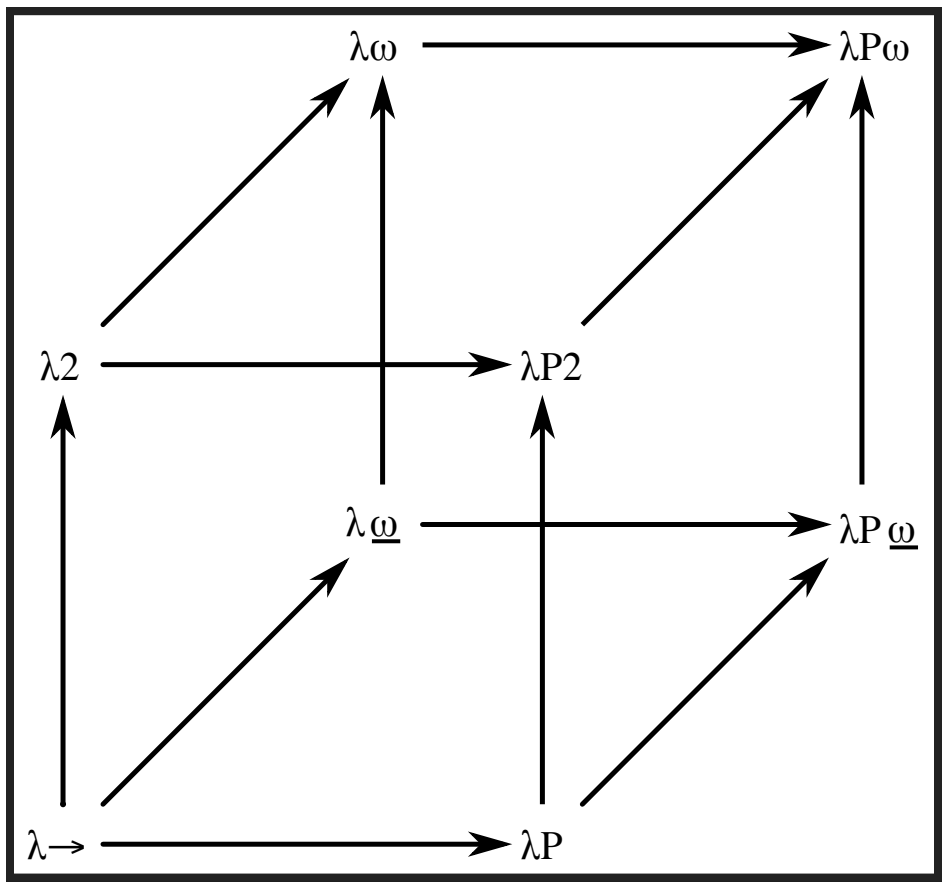
- 0, 1, проекции
- сложение, умножение
- функция `ifzero`(n, m, p) = `if` n = 0 `then` m `else` p

В следующий раз мы рассмотрим различные способы расширения набора типов и постараемся убрать разделение между типами и термами.

Расширения типизированного λ-исчисления

λ-куб Барендрегта

Barendregt H.P. Introduction to generalized type systems // J. Funct. Program. 1991. Vol. 1, № 2. P. 125–154.



Расширения типизированного λ-исчисления

полиморфное λ-исчисление

λ-исчисление с конструкторами типов

λ-исчисление с зависимыми типами

Б. Пирс. Типы в языках программирования. М.: Лямбда-прес, 2011. Часть V (глава 23), VI.

Могу порекомендовать дополнительно введение, главы 5-6, 8-12, 20-24, 26 как литературу по значительной части нашего курса.

Задачи со звёздочкой

Задача 5.1/6.1** Реализовать алгоритм двусторонней проверки типов для λ-исчисления с простыми типами (синтаксис входных данных — на ваше усмотрение).

- бонусная * — добавить редукцию
- бонусная * — визуализация редукции

Задача 5.2/6.2* Реализовать нумералы Чёрча с операцией сложения.

- бонусная * — умножение
- бонусная * — нетривиальный пример расширенного полинома

Задача 6.3*** Реализовать алгоритм проверки типов для одного из расширений λ-исчисления (полиморфное, с конструкторами типов, с зависимыми типами)