Прикладная теория типов

Лекция 3: Просто типизированное лямбда-исчисление $\lambda_{\to}$

Воронов Михаил

ВМК МГУ, Осень 2024

План лекции

1. Нормализация, нормальная форма

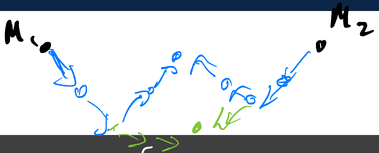
2. Типы и системы типов (теории типов)

3. Просто типизированное λ-исчисление $\lambda_{\rightarrow}$

План лекции

1. Нормализация, нормальная форма
2. Типы и системы типов (теории типов)
3. Просто типизированное λ-исчисление $\lambda_{\to}$

Существование общего редукта



Лемма о существовании общего редукта

Если $\forall M_1, M_2 \in \Lambda$, $M_1 =_\beta M_2$, то тогда существует общий редукт L, такой что $M_1 \twoheadrightarrow_\beta L$, $M_2 \twoheadrightarrow_\beta L$.

Доказательство

Доказательство индукцией по способам генерации, ромбы спрямляются с помощью теоремы Чёрча-Россера.

Редуцируемость к нормальной форме

Лемма о редуцируемости к $\beta - NF$

Если терм $M \in \Lambda$ имеет N в качестве $\beta - NF$, то его можно свести к ней, т.е. $M \twoheadrightarrow_\beta N$.

Доказательство

Пусть $M =_\beta N$, где N находится в $\beta - NF$, тогда по лемме о существовании общего редукта, существует терм L, такой что $M \twoheadrightarrow_\beta L, N \twoheadrightarrow_\beta L$. Но тогда $N \equiv L$, т.к. в N отсутствуют редексы, отсюда $M \twoheadrightarrow_\beta N$.

Данная лемма может быть использована для доказательства отсутствия нормальной формы у каких-то термов. Например, можно заметить, что Ω редуцируется только к себе и при этом не находится в нормальной форме.

Единственность нормальной формы

Лемма о единственности $\beta - NF$

Любой терм $M \in \Lambda$ имеет не более одной $\beta - NF$

Доказательство

От противного: пусть M имеет N_1 и N_2 в качестве $\beta - NF$, тогда $N_1 =_\beta M =_\beta N_2$. Тогда по теореме Чёрча-Россера $\exists L \in \Lambda : N_1 \twoheadrightarrow_\beta L, N_2 \twoheadrightarrow_\beta L$. Но тогда по лемме редукции, $N_1 \equiv L \equiv N_2$.

Данная лемма позволяет ввести отношение эквивалентности на множестве термов путём сравнения нормальных форм, к которым они редуцируются. Например, комбинаторы K и K_* относятся к разным классам эквивалентности, а Ω и $(\lambda x.x)\Omega$ к одному. Это отношение с точностью до α-преобразования обычно используется, как более широкое понятие равенства термов.

Головные нормальные формы

- **Головной редекс** - это редекс $(\lambda z.M)N_1$ в терме вида $\lambda x_1 \dots x_n.(\lambda z.M)N_1 \dots N_k$
- **Головная нормальная форма** (HNF) - это терм, в котором отсутствует головной редекс, т.е. это терм вида $\lambda x_1 \dots x_n.zN_1 ... N_k$, где z - это переменная (возможно, связанная).
- **Слабая головная нормальная форма**(WHNF) – HNF, λ-абстракция (т.е. не редекс на верхнем уровне).

Сильная и слабая нормализуемость

- Разделают сильно и слабо нормализуемые термы. В первых любой путь редукции приводит к нормальной форме, а у вторых существует бесконечный путь редукции.
- Примером слабо нормализуемого терма является $KI\Omega$.
- Поэтому, несмотря на свойство Чёрча-Россера, порядок применения редукций очень важен.

Слабая головная нормальная форма

- Ещё один недостаток нормализации заключается в том, что она неустойчива относительно расширения терма.
- Например, терм Ω не нормализуем, однако больший терм, включающий в себя заданный, уже нормализуется ($KI\Omega$).
- Данная проблема решается отказом от применения некоторых редукций в терме, т.е. ослабление требований к нормальной форме и использование слабой головной нормальной формы.
- Всякая HNF является $WHNF$, но не наоборот.
- Часто не требуется доводить редукцию до нормальной формы, ограничиваются $WHNF$, это позволяет избежать захвата переменной при редуцировании замкнутого терма

Типы вызовов

- **Вызов по значению** (call-by-value) - сначала редуцируются аргументы, а потом применяется функция. Соответствует аппликативной стратегии редукции.
  $square(1+1) \to square(2) \to 2*2 \to 4$
- **Вызов по имени** (call-by-name) - сначала редуцируется функция, а только потом редуцируется аргументы. Соответствует нормальной стратегии редукции.
  $square(1+1) \to (1+1)*(1+1) \to 2*(1+1) \to 2*2 \to 4$
- **Вызов по необходимости** (call-by-need) - порядок как при вызове по имено, но с мемоизацией результатов.
  $square(1+1) \to 1+1\ in\ x*x \to 2\ in\ x*x \to 2*2 \to 4$

$f(x_1, x_2, \ldots, x) \to$

let let x = 1+1 in x*x

План лекции

1. Нормализация, нормальная форма
2. Типы и системы типов (теории типов)
3. Просто типизированное λ-исчисление $\lambda_{\to}$

Особенности(недостатки) λ исчисления

ф-я

аргументы

$(\lambda x. N)(F)$

1. Возможно самоприменение "функции"к себе: FF
2. У любого терма есть неподвижная точка
3. Не у всех термов есть нормальная форма, возможна бесконечная редукция
4. Возможно выразить некоторые парадоксы

Типы в computer science

Существует несколько распространённых определений понятия типа (Parnas, Shore, Weiss, 1976):

$Int \Rightarrow Int$

- **Тип** - это синтаксическая метка, ассоциированная с переменной (синтаксический).
- **Тип** - это некоторая композиция более примитивных типов (обычно, машинных типов) (репрезентативный).
- **Тип** - это некоторая композиция более примитивных типов и набор операторов для манипулирования этими представлениями (репрезентативный и поведенческий).
- **Тип** - это множество всех возможных значений, которая может принимать переменная (классы эквивалентности) (ориентированный на значения).
- **Тип** - это множество всех возможных значений, которая может принимать переменная и множество функций, которые могут быть применены к этим значениям (ориентированный на значения и поведенческий).

Системы типов

TAPL

Существует несколько распространённых определений понятия типа (Parnas, Shore, Weiss, 1976):

- В более общем смысле, системы типов относятся и к математике, и к философии. В этом смысле ситемы типов были впервые использованы в 1900-х как средство избежать логических парадоксов (например, парадокса Рассела, 1902).
- В течение 20-го века типы превратились в стандартный инструмент логики, в частности, в теории доказательств:
    - теория типов Рассела (ramified theory of types, 1910)
    - простая теория типов Рамсея (simple theory of types, 1925)
    - конструктивная теория типов Мартина-Лёфа (constructive theory of types, 1973,1984)
    - чистые системы типов Берарди (1988), Терлоу (1989) и Берендрегта (1992)

Системы типов

- Система типов (или теория типов) (по *TAPL*) - гибко управляемый синтаксический метод доказательства отсутствия в программе определённых видов поведения при помощи классификации выражений языка по разновидностям вычисляемых ими значений.
- В более общем смысле, системы типов относятся и к математике, и к философии. В этом смысле ситемы типов были впервые использованы в 1900-х как средство избежать логических парадоксов (например, парадокса Рассела, 1902).
- В течение 20-го века типы превратились в стандартный инструмент логики, в частности, в теории доказательств:
  - теория типов Рассела (ramified theory of types, 1910)
  - простая теория типов Рамсея (simple theory of types, 1925)
  - конструктивная теория типов Мартина-Лёфа (constructive theory of types, 1973,1984)
  - чистые системы типов Берарди (1988), Терлоу (1989) и Берендрегта (1992)

- Внутри CS обычно разделяют две ветви: прикладная и более абстрактная (соответствие Карри-Ховарда-Ламбека)
- более прикладные обычно отличаются тем, что допускают отсутствие свойства тотальности, потому что большинство язынок программирования не обладают таким свойством
- систему типов можно рассматривать как статическую аппроксимацию поведения программы во время выполнения
- будучи такими, системы типов обязательно : они способны однозначно доказать отсутствие нежелательных видов поведения, но не могут доказать их наличие. Это приводит к тому, что отвергаются некоторые, на самом деле, корректные программы.

if ( ) {

< 0

3

План лекции

1. Нормализация, нормальная форма
2. Типы и системы типов (теории типов)
3. Просто типизированное λ-исчисление $\lambda_{\to}$

Просто типизированное λ-исчисление $\lambda_{\to}$

Λ

Будем считать, что тип - это метка, которая присваивается терму и которая обладает определёнными свойствами.

Множество типов Θ из $\lambda_{\to}$ определяется индуктивно:

$(\alpha \to \beta \to \gamma) \to \gamma$

1. Переменные типа: $\alpha, \beta, \dots \in \Theta$
2. Типы функций (стрелочные типы): $\forall \sigma, \tau \in \Theta \Rightarrow (\sigma \to \tau) \in \Theta$

Соглашение: начальные буквы греческого алфавита используются для обозначения конкретных переменных, а конечные (σ, τ, γ) для обозначения произвольных типов (в том числе и стрелочных).

Примеры типов в $\lambda_{\to}$

Тип функции правоассоциативен, это выражает порядок принимаемых аргументов:
$\sigma_1 \to \sigma_2 \to \ldots \to \sigma_n \equiv (\sigma_1 \to (\sigma_2 \to \ldots \to \sigma_n) \ldots)$
Примеры:

$$(\alpha \to \beta) \equiv \alpha \to \beta$$

$$(\alpha \to (\beta \to \gamma)) \equiv \alpha \to \beta \to \gamma$$

$$((\alpha \to \beta) \to \gamma) \equiv (\alpha \to \beta) \to \gamma$$

$$((\alpha \to \beta) \to ((\gamma \to \gamma) \to (\alpha \to \beta))) \equiv (\alpha \to \beta) \to (\gamma \to \gamma) \to (\alpha \to \beta)$$

Как типизировать термы?

- **Переменная** — $\forall$ тип

$\underbrace{x : \alpha}_{\text{утверждение о типизации}}, \quad y : \alpha \to \beta$

MN

- **Аппликация** MN, то
  - M — ф-я, стрелочный тип $(M : \sigma \to \tau)$
  - N — аргумент, „подходящего" типа $N : \sigma$
  - $(MN) : \tau$

$$\begin{array}{l} x : \gamma \\ y : \gamma \to \beta \qquad yx : \beta \\ z : \beta \to \alpha \\ z(yx) : \alpha \\ \hline x(yx) : \alpha \\ x : \beta \to \alpha \\ y : (\beta \to \alpha) \to \beta \end{array}$$

- **Абстракция** $\lambda x.\, M$
  - Тип абстракции — стрелочный $(\lambda x. M) : \sigma \to \tau$
  - $(\lambda x. M) : \sigma \to \tau \Rightarrow x : \sigma$, $M : \tau$

с-ма Карри

$(\lambda x. M) : \sigma \to \tau$

с-ма Чёрч

$(\lambda x : \alpha.\, M) : \sigma \to \tau \qquad (\lambda x^{\alpha}.\, M) : \sigma \to \tau$

$\lambda x\, y.\ x(\underbrace{y\,x}_{\beta}) : (\beta \to \alpha) \to ((\beta \to \alpha) \to \beta) \to \alpha$ ← а-я Карри

$x : \beta \to \alpha$

$y : (\beta \to \alpha) \to \beta$

$\lambda x^{\beta \to \alpha}\, y^{(\beta \to \alpha) \to \beta}.\ x(y\,x) : (\beta \to \alpha) \to ((\beta \to \alpha) \to \beta) \to \alpha$ ← а-ия Чёрч

Пред терм (псевдо терм) $\lambda_{\to}$ – терм из бестипового λ.

$\lambda x^{\alpha}. y^{\beta}. x(y\,x)$ – предтерм $\lambda_{\to}$ а-ия Чёрч

Объявление – это утв о типизации для переменной
$x : \alpha \to \beta$

Контекст – это мн-во объявлений для различных термовых переменных

$$\Gamma = \{ x_1 : \sigma_1, x_2 : \sigma_2, \ldots, x_n : \sigma_n \}$$

Выводимость —

$$\Gamma \vdash M : \tau$$

Правила вывода в контексте:

1. $(x:\sigma) \in \Gamma \Rightarrow \Gamma \vdash x:\sigma$ $\qquad x:\sigma \vdash \;\Rightarrow\; \overline{x:\sigma} \vdash x:\sigma$

2. $\Gamma \vdash M : \sigma \to \tau \quad \Gamma \vdash N:\sigma \Rightarrow \Gamma \vdash (MN) : \tau$

3. $\Gamma, x:\sigma \vdash M:\tau \Rightarrow \Gamma \vdash (\lambda x. M) : \sigma \to \tau$

$$\frac{\text{посылка}}{\text{следствие}} \quad \tau$$

1. $\Gamma \vdash x:\sigma$ если $(x:\sigma) \in \Gamma$ (аксиома)

2. $$\frac{\Gamma \vdash M : \sigma \to \tau \qquad \Gamma \vdash N : \sigma}{\Gamma \vdash (MN) : \tau}$$ (elim $\to$) ($\to E$)

3. $$\frac{\Gamma, x:\sigma \vdash M : \tau}{\Gamma \vdash (\lambda x^{\sigma}. M) : \sigma \to \tau}$$ (intro $\to$) ($\to I$)

а-ля Чёрч

Задачи в теории типов

① Well-typedness

$? \vdash M : ?$

1a. $\Gamma \vdash M : ?$ (Type assignment)

② Type-checking

$\Gamma \overset{?}{\vdash} M : \sigma$

③ Type inhabitation

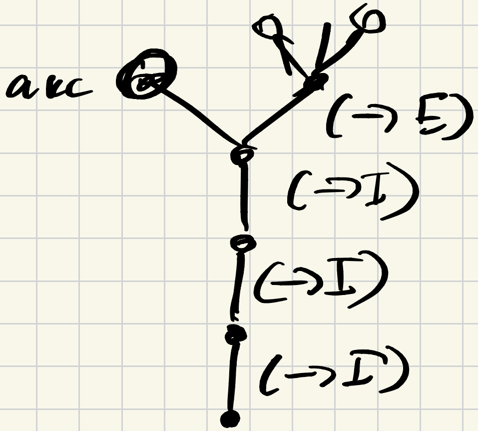
$\Gamma \vdash ? : \sigma$

$$\dfrac{\dfrac{x:\alpha \quad y:\beta \vdash x:\alpha}{x:\alpha \vdash (\lambda y^{\beta}.\, x) : \beta \to \alpha}}{\vdash (\lambda x^{\alpha} y^{\beta}.\, x) : \alpha \to \beta \to \alpha}$$

аксиома

$$\frac{(2)}{(1)} \to E$$

$\circlearrowleft \to I$



$$\underline{\lambda f g x.\ f(g x)}$$

$$\dfrac{\Gamma \vdash f : (\beta \to \gamma) \qquad \dfrac{\Gamma \vdash g : (\alpha \to \beta) \quad \Gamma \vdash x : \alpha}{\Gamma \vdash g x : \beta}}{f : (\beta \to \gamma),\, g : (\alpha \to \beta),\, x : \alpha \vdash f(g x) : \gamma}$$

Γ

$$\dfrac{f : (\beta \to \gamma),\, g : (\alpha \to \beta),\, x : \alpha \vdash f(g x) : \gamma}{f(\beta \to \gamma),\, g : (\alpha \to \beta) \vdash \lambda x.\, f(g x) : \alpha \to \gamma}\ (\to I)$$

$$\dfrac{f(\beta \to \gamma),\, g : (\alpha \to \beta) \vdash \lambda x.\, f(g x) : \alpha \to \gamma}{f : (\beta \to \gamma) \vdash \lambda g x.\, f(g x) : (\alpha \to \beta) \to \alpha \to \gamma}\ (\to I)$$

$$\dfrac{f : (\beta \to \gamma) \vdash \lambda g x.\, f(g x) : (\alpha \to \beta) \to \alpha \to \gamma}{\vdash \lambda f g x.\, f(g x) : (\beta \to \gamma) \to (\alpha \to \beta) \to \alpha \to \gamma}\ (\to I)$$

ass

$(\to E)$

$(\to I)$

(a) 1 $\boxed{f : \beta \to \alpha}$

(b) 2 $\boxed{g : \alpha \to \beta}$

(c) 3 $\boxed{x : \alpha}$

4 $g\,x : \beta$ $(\to E\ 2, 3)$

5 $f(g\,x) : \gamma$ $(\to E\ 1, 4)$

6 $\lambda x.\ f(g\,x) : \alpha \to \gamma$ $(\to I\ 5)$

7 $\lambda g\,x.\ f(g\,x) : (\alpha \to \beta) \to \alpha \to \gamma$ $(\to I\ 6)$

8 $\lambda f\,g\,x.\ f(g\,x) : (\beta \to \gamma) \to (\alpha \to \beta) \to \alpha \to \gamma$ $(\to I\ 7)$

акс
(→E)
(→I)

$\boxed{f : \beta \to \gamma}$

$\boxed{g : \alpha \to \beta}$

$\boxed{x : \alpha}$

$f : \beta \to \gamma$

$g : \alpha \to \beta$

$x : \alpha$

$g x : \beta \quad (\to E)$

$f(g x) : \gamma \quad (\to E)$

$\lambda x.\ f(g x) : \alpha \to \gamma \quad (\to I)$

$\lambda g x.\ f(g x) : (\alpha \to \beta) \to \alpha \to \gamma \quad (\to I)$

$\lambda f g x.\ f(g x) : (\beta \to \gamma) \to (\alpha \to \beta) \to \alpha \to \gamma \quad (\to I)$

$S = \lambda x y z.\ x z (y z) : \ ?$

$\lambda x y z.\ x z (y z) : (\sigma \to \beta \to \tau) \to (\sigma \to \beta) \to \sigma \to \tau$

β

$\sigma \to \beta \to \tau$ $\qquad \underbrace{(y z)}_{\beta}$

акс

(⇒ E)

(⇒ I)

$z : \alpha \to \beta \to \alpha$

$x : \alpha$

$y : \beta$

$x : \alpha$

$\lambda y.\ x : \beta \to \alpha$ (⇒ I)

$\lambda x \cancel{y} y.\ x : \alpha \to \beta \to \alpha$ (⇒ I)