Теория типов

Михайлов Максим

22 сентября 2021 г.

Оглавление

Лекция 1

7 сентября

1 Лямбда-исчисление

То, чем мы будем заниматься, можно назвать прикладной матлогикой.

В рамках курса матлогики мы рукомахательно рассмотрели изоморфизм Карри-Ховарда, в этом курсе мы его формализуем. Мы затронем систему типов Хиндли-Милнера *(Haskell)* и язык Arend, основанный на гомотопической теории типов.

1.1 Определение

В 20-30х годах XX века Алонзо Чёрчем была создана альтернатива теории множеств как основе математики — лямбда-исчисление. Основная идея — выбросить из языка все, кроме вызова функций.

В лямбда исчислении есть три конструкции:

- Функция *(абстракция)*: $(\lambda x.A)$
- Применение функции *(аппликация)*: $(A\ B)$
- Переменная *(атом)*: x

Большими буквами начала латинского алфавита мы будем обозначать термы, малыми буквам конца — переменные. λ жадная, как $\forall$ и $\exists$ в исчислении предикатов. Аппликация идёт слева направо, т.е. $\lambda p.p\ F\ T = \lambda p.((p\ F)\ T)$

Вычисление происходит с помощью β-редукции, его мы определим позже, общее понимание у нас есть из вводной лекции функционального программирования.

1.2 Булево исчисление

Определим булево исчисление в λ-исчислении:

- $T := \lambda x.\lambda y.x$ — истина
- $F := \lambda x.\lambda y.y$ — ложь
- Not $:= \lambda p.p\ F\ T$

$$\begin{aligned} \text{Not}\ F &\to_\beta \\ ((\lambda x.\lambda y.y)\ F)\ T &\to_\beta \\ (\lambda y.y)\ T &\to_\beta T \end{aligned}$$

- And $:= \lambda a.\lambda b.a\ b\ F$

And берёт свой второй аргумент, если первый аргумент истина и ложь иначе.

And использует идею **карринга** — функция от 2 аргументов есть функция от первого аргумента, возвращающая другую функцию от второго аргумента.[1] Например, в выражении "$((+)\ 2)\ 3$" $((+)\ 2)$ это функция, которая прибавляет к своему аргументу 2.

1.3 Числа

Числа в лямбда-исчислении кодируются **нумералами Чёрча**. Это только один из способов кодировки, есть и другие. Общая идея — число n применяет данную функцию к данному аргументу n раз.

- $0 = \lambda f.\lambda x.x$
- $1 = \lambda f.\lambda x.f\ x$
- $3 = \lambda f.\lambda x.f\ (f\ (f\ x))$
- $\overline{n+1} = \lambda f.\lambda x.f(\overline{n}\ f\ x)$
- $(+1) = \lambda n.\lambda f.\lambda x.n\ f\ (f\ x)$ — функция инкремента.
- $(+) = \lambda a.\lambda b.b\ ((+)\ \overline{1})\ a$: b раз прибавляет единицу к a.
- $(\cdot) = \lambda a.\lambda b.a\ ((+)\ b)\ \overline{0}$: a раз прибавляет b к 0.

Ходят легенды, что Клини изобрел декремент у зубного врача под действием наркоза. Существует много способов определить декремент различных степеней упоротости.

Рассмотрим декремент, основанный на следующей идее: пусть есть упорядоченная пара $\langle a, b\rangle$ и функция $(*) : \langle a, b\rangle \mapsto \langle b, b+1\rangle$. Тогда применив $(*)$ n раз к $\langle 0, 0\rangle$ и взяв первый элемент, возьмём первый элемент пары.

[1] Аналогично для n аргументов.

Упорядоченная пара определяется следующим способом:

$$\text{MkPair} = \lambda a.\lambda b.(\lambda p.p\ a\ b)$$

Можно потрогать эмулятор лямбда-исчисления `lci`, будет полезно для домашних заданий.

1.4 Типизированное лямбда-исчисление

Лямбда-исчисление для нас будет просто языком программирования. Для начала мы его типизируем, потому что нетипизированное лямбда-исчисление противоречиво.

Пусть у каждого выражения A есть тип τ, что обозначается $A : \tau$. Также используется некоторый контекст с переменными и их типами, обозначаемый M. Все вместе это записывается как $M \vdash A : \tau$, что напоминает исчисление предикатов.

1.5 Y-комбинатор и противоречивость нетипизированного λ-исчисления

Мы хотим, чтобы $\to_\beta$ сохраняло значения, т.к. иначе мы вообще не можем говорить о равенстве термов.

Определение. $Y \coloneqq \lambda f.(\lambda x.f\ (x\ x))(\lambda x.f\ (x\ x))$ — Y-комбинатор, для него верно $Yf \approx f(Yf)$. Такое свойство называется "быть **комбинатором неподвижной точки**", т.е. он находит неподвижную точку функции: A такое, что $f(A) = A$.

Пусть мы добавили бинарную операцию $(\supset)$ — импликацию с некоторыми аксиомами. Оказывается, что доказуемо любое A. Мы это докажем на последующих лекциях.

Y-комбинатор полезен тем, что позволяет реализовывать рекурсию.

Пример. Запишем факториал в неформальном виде:

$$\text{Fact} = \lambda n.\text{If}\ (\text{IsZero}\ n)\ \overline{1}\ (\text{Fact}\ (n-1)\cdot n)$$

На самом деле Fact есть неподвижная точка функции

$$\lambda f.\lambda n.\text{If}\ (\text{IsZero}\ n)\ \overline{1}\ (f\ (n-1)\cdot n)$$

по определению неподвижной точки функции. Тогда Fact это

$$Y(\lambda f.\lambda n.\text{If}\ (\text{IsZero}\ n)\ \overline{1}\ (f\ (n-1)\cdot n))$$

У нас появляется проблема: есть выражения, которым мы не можем приписать значение, например

$$Y(\lambda f.\lambda x.f\ (\text{Not}\ x))$$

Эта проблема происходит из-за того, что наш язык слишком мощный — мы написали решатель любых уравнений, даже тех, у которых нет решения. Логичный выход из этой ситуации — запретить то, из-за чего у нас возникают проблемы. Как запретить Y? Оказывается, это позволяют сделать типы — они будут делить выражения на добропорядочные и недобропорядочные.

Лекция 2

14 сентября

2 Формализация λ-исчисления

Определение. **Пред-λ-терм** определяется индуктивно как одно из:

1. x — переменная
2. $(L\ L)$ — применение
3. $(\lambda x.L)$ — абстракция

Почему пред-λ-терм? Мы не хотим различать $\lambda x.x$ и $\lambda y.y$.

Определение. **α-эквивалентность** — обозначается $A =_\alpha B$ и выполняется, если[1]:

1. $A \equiv x, B \equiv x$ — одна и та же переменная
2. $A \equiv P\ Q, B \equiv R\ S, P =_\alpha R, Q =_\alpha S$
3. $A \equiv \lambda x.P, B \equiv \lambda y.Q$ и существует t — новая переменная, такая что $P[x := t] =_\alpha Q[y := t]$

Определение. Свобода для подстановки: $A[x := B]$, никакое свободное вхождение переменной в B не станет связанным.

Определение (λ-терм). Множество всех λ-термов это $\Lambda/_{=_\alpha}$

Определение (β-редекс). Выражение вида $(\lambda x.A)\ B$

Определение (β-редукция). Обозначается $A \to_\beta B$ и выполняется, если выполняется одно из:

1. $A \equiv P\ Q, B \equiv R\ S$ и либо $P \to_\beta R$ и $Q =_\alpha S$, либо $P =_\alpha R$ и $Q =_\alpha S$.

[1] И только если.

2. $A \equiv \lambda x.P, B \equiv \lambda x.Q$ и $P \to_\beta Q$
3. $A \equiv (\lambda x.P)\ Q, B \equiv P[x \coloneqq Q]$ и Q свободно для подстановки.

Определение. Придуман Моисеем Шейнфинкелем.

$I \coloneqq \lambda x.x$ — Identität[2]

Определение.

- $K = \lambda x.\lambda y.x$
- $\Omega = \omega\ \omega$
- $\omega = \lambda x.x\ x$

Пример.

$$\begin{tikzcd}
KI\Omega \arrow[loop above] \arrow[r, "\beta"] \arrow[d, "\beta"] & ((\lambda x.\lambda y.x)I)\Omega \arrow[loop above] \\
(\lambda y.I)\Omega \arrow[loop left] \arrow[d, "\beta"] & \\
I &
\end{tikzcd}$$

Определение. R обладает **ромбовидным свойством** *(diamond)*, если для любых a, b, c, таких что:

1. aRb, aRc
2. $b \neq c$

существует d: bRd и cRd.

$$\begin{tikzcd}
& A \arrow[dl] \arrow[dr] & \\
B \arrow[dr] & & C \arrow[dl] \\
& D &
\end{tikzcd}$$

Пример. $>$ на $\mathbb{Z}$ не ромбовидно: для $a = 3, b = 2, c = 1$ выполнено условие, но $\nexists d$.

$>$ на $\mathbb{R}$ ромбовидно.

[2] Тождество (с немецкого)

Определение (β-редуцируемость)**.** Рефлексивное, транзитивное замыкание отношения $\to_\beta$, обозначается $\twoheadrightarrow_\beta$.

Теорема 1 (Чёрча-Россера)**.** β-редуцируемость обладает ромбовидным свойством.

Определение. $\rightrightarrows_\beta$ — параллельная β-редукция, выполняется если:

0. $A =_\alpha B$
1. $A \equiv P\ Q, B \equiv R\ S$ и $P \rightrightarrows_\beta R$ и $Q \rightrightarrows_\beta S$.
2. Аналогично β-редукции.
3. Аналогично β-редукции.

Лемма 1. $(\rightrightarrows_\beta)$ обладает ромбовидным свойством.

Лемма 2. Если R обладает ромбовидным свойством, то R^* обладает ромбовидным свойством.

Доказательство. Две индукции. □

Лемма 3. $(\rightrightarrows_\beta) \subseteq (\twoheadrightarrow_\beta)$

Доказательство теоремы Чёрча-Россера. Заметим, что:

1. $(\rightrightarrows_\beta)^* \subseteq (\twoheadrightarrow_\beta)$ — из леммы
2. $(\twoheadrightarrow_\beta) \subseteq (\rightrightarrows_\beta)^*$ — из определения
3. Т.к. $(\rightrightarrows_\beta)^*$ обладает р.с., то и $(\twoheadrightarrow_\beta)$ обладает р.с.

□

Следствие 1.1. У λ-выражения существует не более одной нормальной формы.

Доказательство. Пусть A имеет две нормальные формы: $A \twoheadrightarrow_\beta B, A \twoheadrightarrow_\beta C$ и $B \neq_\alpha C$. Тогда есть D: $B \twoheadrightarrow_\beta D$ и $C \twoheadrightarrow_\beta D$. Противоречие. □

Определение. Нормальный порядок редукции — редуцируем самый левый редекс.

Теорема 2. Если нормальная форма существует, она может быть получена нормальным порядком редукции.

Примечание. Нижеследующее объяснение — с практики.

Рассмотрим $Y\ f\ =_\beta\ f\ (Y\ f)\ =_\beta\ f\ (f\ (Y\ f))\ =_\beta\ \dots$. Можно считать, что у f сколько угодно аргументов, первый аргумент можно считать указателем на свой рекурсивный вызов.

Пример. Числа Фибоначчи:

```
fib a b n =
    if n = 0 then a
    else fib b (a + b) (n - 1)
```

Здесь решение уравнения заметано под ковер, в λ-исчислении оно видно:

$$\text{Fib} = \lambda f.\lambda a.\lambda b.\lambda n.(\text{IsZero}\ n)\ a\ (f\ f\ a\ (a+b)\ (n-1))$$

Здесь f передается само себе, чтобы иметь ссылку на себя для рекурсивного вызова.

Для работы Fib нужно дать его самому себе: Fib Fib $1\ 1\ 10$.

Лекция 3

21 сентября

В λ-исчислении можно сделать:

1. Целые числа, где $\langle a, b\rangle \leftrightarrow a - b$
2. Рациональные числа в виде дробей
3. Матлогику?

Попытки сделать матлогику всегда приводили к парадоксам.

Оказывается, нельзя относиться к любому выражению как к логическому.

Обозначение. $\supset$ — импликация

Пример. Рассмотрим комбинатор $\Phi_A =_\beta A \supset \Phi_A$. Это $Y\ (\lambda f.\lambda a.a \supset f\ a)$.

Добавим аксиому $(A \supset (A \supset B)) \supset (A \supset B)$. Если такой аксиомы нет, то теория грустная.

Мы также хотим, чтобы если $X =_\beta Y$, то $X \supset Y$.

Каким-то образом мы получим парадокс.

3 Просто-типизированное λ-исчисление

Определение (типовые переменные).

- α, β, γ — атомарные
- τ, σ — составные

2 традиции:

1. Исчисление по Чёрчу

2. Исчисление по Карри

Мы сначала рассмотрим исчисление по Карри.

3.1 Исчисление по Карри

Типизация: $\Gamma \vdash A : \tau, \Gamma = \{x_1 : \tau_1, x_2 : \tau_2 \dots\}$

Правила:

1. $\dfrac{}{\Gamma, x_1 : \tau_1 \vdash x_1 : \tau_1}$ Ax.
2. $\dfrac{\Gamma \vdash A : \sigma \to \tau \qquad \Gamma \vdash B : \sigma}{\Gamma \vdash A\, B : \tau}$
3. $\dfrac{\Gamma, x : \tau \vdash A : \sigma}{\Gamma \vdash \lambda x.A : \tau \to \sigma}$

Пример.

$$\lambda f^{\alpha \to \alpha}.\lambda x^{\alpha}.f\ (f\ x) : (\alpha \to \alpha) \to \alpha \to \alpha$$

Подгоним доказательство под результат:

$$\dfrac{\dfrac{\dfrac{\dfrac{}{f : \alpha \to \alpha \vdash f : \alpha \to \alpha} \qquad \dfrac{\dfrac{}{\Gamma \vdash f : \alpha \to \alpha} \qquad \dfrac{}{\Gamma \vdash x : \alpha}}{\Gamma \vdash f\ x : \alpha}}{f : \alpha \to \alpha, x : \alpha \vdash f\ (f\ x) : \alpha}}{f : \alpha \to \alpha \vdash \lambda x.f\ (f\ x) : \alpha \to \alpha}}{\lambda f.\lambda x.f\ (f\ x) : (\alpha \to \alpha) \to (\alpha \to \alpha)}$$

Теорема 3. Если $\Gamma \vdash A : \tau$, то любое подвыражение имеет тип.

Доказательство. По индукции по длине.

База. Это правило 1.

Переход. Пусть любое выражение длиной $< n$ символов обладает искомым свойством. Покажем искомое для $A : |A| = n$. Рассмотрим варианты того, по какому правилу доказана типизируемость A:

1. Второе правило: B и C короче A, следовательно для них искомое верно.
2. Третье правило: аналогично для x, B

□

Теорема 4 (Subject reduction, о редукции). Если $\Gamma \vdash A : \sigma$ и $A \twoheadrightarrow_\beta B$, то $\Gamma \vdash B : \sigma$

Доказательство. Скучно.

Самая интересная часть: рассмотрим $A \to_\beta B$. Случаи:

1. $\lambda x.A \to \lambda x.B$ — индукция
2. $A\ B$ — индукция
3. $(\lambda x.A)\ B \to A[x := B]$

   По теореме о типизации подвыражений, $(\lambda x^{\tau\to\sigma}.A^\sigma)\ B^\tau\ :\ \sigma$. Кроме того, доказывается $(A[x := B]) : \sigma$.

$\square$

Лемма 4. Если $\Gamma, x : \tau \vdash A : \sigma$,$\Gamma \vdash B : \tau$, то $\Gamma \vdash A[x := B] : \sigma$

Теорема 5 (Чёрча-Россера). Если $\Gamma \vdash M : \sigma$ и существуют $N, P : M \twoheadrightarrow_\beta N, M \twoheadrightarrow_\beta P$, то найдется такой S, что $\Gamma \vdash S : \sigma$ и $N \twoheadrightarrow_\beta S$ и $P \twoheadrightarrow_\beta S$

3.2 Исчисление по Чёрчу

Язык:

- x — переменная
- $A\ B$ — аппликация
- $\lambda x^\tau.P$ — абстракция

Обозначение. Когда нужно различить исчисления, будем писать $\vdash_{\text{Ч}}$ или $\vdash_{\text{К}}$

Теорема 6. Если контекст Γ и выражение P типизируется, то $\Gamma \vdash_{\text{Ч}} P : \sigma$

Пример.

$$\vdash_{\text{К}} \lambda x.x : \alpha \to \alpha$$

$$\vdash_{\text{К}} \lambda x.x : \beta \to \beta$$

$$\vdash_{\text{Ч}} \lambda x^\sigma.x : \sigma \to \sigma$$