Теория типов

Михайлов Максим

23 октября 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:

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

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

- And $\coloneqq \lambda a.\lambda b.a\ b\ F$

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 $Y f \approx f(Y f)$. 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. $(\lambda x.A)\ B \to A[x \coloneqq B]$

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

$\square$

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

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.

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

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

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

Лекция 4

28 сентября

3.3 Противоречивость нетипизированного λ-исчисления

???

1. Логические выражения
2. Запрещенные выражения

Y явно нехорошее выражение. $\Phi_A =_\beta \Phi_A \supset A$

Добавим очевидные аксиомы:

1. $A =_\beta B$, то $\vdash A \supset B, \vdash B \supset A$. Почему? Потому что мы хотим, чтобы $\sin 0 = 0$, а не только $\sin 0 \to 0$
2. $(A \supset A \supset B) \supset (A \supset B)$
3. $A, A \supset B$, тогда B

Тогда заметим, что при любом $A, \vdash A$:

$$
\begin{aligned}
&\Phi_A \supset \Phi_A \\
&\Phi_A \supset (\Phi_A \supset A) \\
(\Phi_A \supset (\Phi_A \supset A)) \supset{} &(\Phi_A \supset \Phi_A) \\
&\Phi_A \supset A \\
&\Phi_A \\
&A
\end{aligned}
$$

4 Изоморфизм Карри-Ховарда

$$\frac{}{\Gamma, x : \tau \vdash x : \tau}\ x \notin \Gamma \qquad \frac{}{\Gamma, \tau \vdash \tau}$$

$$\frac{\Gamma \vdash A : \sigma \to \tau \quad \Gamma \vdash B : \sigma}{\Gamma \vdash A\, B : \tau} \qquad \frac{\Gamma \vdash \sigma \to \tau \quad \Gamma \vdash \sigma}{\Gamma \vdash \tau}$$

$$\frac{\Gamma, x : \sigma \vdash A : \tau}{\Gamma \vdash \lambda x.A : \sigma \to \tau}\ x \notin \Gamma \qquad \frac{\Gamma, \sigma \vdash \tau}{\Gamma \vdash \sigma \to \tau}$$

Теорема 7 (об изоморфизме Карри-Ховарда).

1. $\Gamma \vdash_{\lambda\to} A : \tau$, то $|\Gamma| \vdash_{\to} \tau$
2. Если $\Delta \vdash_{\to} \tau$, то найдутся $\Gamma, A : |\Gamma| = \Delta, \Gamma \vdash A : \tau$

Определение.

$$|\Gamma| \coloneqq \{\tau \mid x : \tau \in \Gamma\}$$

4.1 Импликационный фрагмент ИИВ

У нас есть только импликация.

Есть три правила: $I_{\to}, E_{\to}$, Ax. Будет ли у нас всё работать? Утверждается, что да.

Обозначение. Доказуемость в И.Ф. ИИВ будем обозначать $\Gamma \vdash_{\to} \tau$

Определение (язык И.Ф. ИИВ).

$$\tau ::= \alpha \mid (\tau \to \tau)$$

Теорема 8. Импликационный фрагмент ИИВ замкнут относительно доказуемости: Если $\Gamma \vdash \tau$ и τ содержит только пропозиционные переменные и импликацию, то $\Gamma \vdash_{\to} \tau$. Обратное очевидно верно.

Доказательство. Рассмотрим Γ^* — множество формул, замкнутых по доказуемости.

$$\tau \in \Gamma^* \Leftrightarrow \Gamma^* \vdash \tau$$

Обозначение. Γ — множество формул, тогда Γ^* — замыкание этого множества по доказуемости, а $\Gamma^{\overset{*}{\to}}$ — замыкание по доказуемости в ИФИИВ.

Рассмотрим множество миров: $\Gamma^{\overset{*}{\to}} \preceq \Delta^{\overset{*}{\to}}$, если $\Gamma^{\overset{*}{\to}} \subseteq \Delta^{\overset{*}{\to}}, \Delta^{\overset{*}{\to}}$ — замкн., $\Gamma^* \Vdash \tau$, если $\tau \in \Gamma^*$

Утверждение. Γ^* образует модель Крипке.

Определение (модель Крипке).

1. Множество миров, упорядоченных отношением $\preceq$
2. $\Vdash$ такое, что если $\Gamma \Vdash \alpha$, то $\Gamma \preceq \Delta$, то $\Delta \Vdash \alpha$.

Тогда $\Gamma \Vdash \tau \to \sigma$ тогда и только тогда, когда в любом $\Gamma \preceq \Delta$ из $\Delta \Vdash \tau$, следует $\Delta \Vdash \sigma$.

Утверждение. $\tau \in \Gamma^{\overset{*}{\to}}$ тогда и только тогда, когда $\Gamma^{\overset{*}{\to}} \vdash_{\text{и}} \tau$

Доказательство. Индукция по структуре τ.

База. $\tau \equiv \alpha$

$\Rightarrow$ $\alpha \in \Gamma^{\overset{*}{\to}}$, то $\alpha \vdash_{\text{и}} \alpha$

$\Leftarrow$ $\alpha \vdash_{\text{и}} \alpha$, тогда очевидно $\alpha \in \Gamma^{\overset{*}{\to}}$

Переход. $\tau \equiv \delta \to \pi$

$\Rightarrow$ $\sigma \to \pi \in \Gamma^{\overset{*}{\to}}$, то $\Gamma^{\overset{*}{\to}} \vdash_{\to} \sigma \to \pi$

$\Leftarrow$ $\Gamma^{\overset{*}{\to}} \vdash_{\text{и}} (\sigma \to \pi)$. Значит, $\Gamma^{\overset{*}{\to}} \Vdash \sigma \to \pi$.

$\square$

Рассмотрим $\Gamma^{\overset{*}{\to}} \preceq \Delta : \Delta \Vdash \sigma$, то $\Delta \Vdash \pi$. Значит, $\Delta \vdash \sigma$. Значит, $\sigma \in \Delta$, т.е. $\Delta \vdash_{\to} \sigma$. Значит, $\Delta \vdash_{\to} \pi$ по М.Р., т.к. $\Gamma^{\overset{*}{\to}} \Vdash \sigma \to \pi \Rightarrow \Delta \Vdash \sigma \to \pi \Rightarrow \Delta \vdash_{\to} \sigma \to \pi$

Утверждение. $\Gamma \Vdash \tau \Leftrightarrow \Gamma|_{\to}\tau$

Доказательство.

$\Rightarrow$ $\Gamma \Vdash \tau$.

1. $\Gamma \Vdash \alpha$, т.е. $\alpha \in \Gamma$, т.е. $\alpha \vdash_{\to} \alpha$
2. $\Gamma \Vdash \sigma \to \pi$.

   Рассмотрим $\Gamma \preceq \Delta$, причём $\Delta \Vdash \sigma$, тогда $\Delta \Vdash \pi$. Т.е. по индукционному предположению $\Delta \vdash_{\to} \sigma$. Пусть $\Delta = \{\Gamma, \sigma\}^*$. Тогда $\Gamma, \sigma \vdash_{\to} \sigma$.

   Тогда $\Gamma, \sigma \Vdash \pi$ по индукционному предложил и определению $\Vdash$. Тогда $\Gamma, \sigma \vdash_{\to} \pi$, т.е. $\Gamma \vdash_{\to} \sigma \to \pi$

$\Leftarrow$ $\Gamma \vdash_{\to} \tau$, тогда $\Gamma \Vdash \tau$

1. $\tau \equiv \alpha$ — очевидно.

2. $\tau \equiv \sigma \to \pi$. Дано, что $\Gamma_{\to} \vdash \sigma \to \pi$.

   Пусть $\Delta \Vdash \sigma$. $\Gamma \preceq \Delta$. Тогда $\Delta \vdash_{\to} \sigma$ по индукционному предположению. $\Gamma \vdash_{\to} \sigma \to \pi$, т.е. $\Delta \vdash_{\to} \sigma \to \pi$. По М.Р. $\Delta \vdash_{\to} \pi$. По индукционному предположению $\Delta \Vdash \pi$. Т.е. $\Gamma \Vdash \sigma \to \pi$. В лекции было $\models$.

□

Схема доказательства:

1. $\tau \in \Gamma^*$, если $\Gamma^* \vdash_{и} \tau$
2. $\Gamma^* \Vdash \tau$
3. $\Gamma^* \Vdash \tau$ тогда и только тогда, когда $\Gamma^* \vdash_{\to} \tau$

□

Обозначение. $\lambda_{\to}$ — типизированное λ-исчисление.

1. Обитаемость: $\overset{?}{\Gamma} \vdash ? : \tau$ — по изоморфизму Карри-Ховарда и теореме об эквивалентности $\Gamma \vdash \tau$
2. Вывод (реконструкция): $\Gamma \vdash A :?$
3. Проверка: $\Gamma \vdash A : \tau$

Пункты 2 и 3 это одно и то же.

Лекция 5

5 октября

5 Алгебраические термы

Определение (алгебраические термы).

$$T ::= \underbrace{a}_{\text{переменная}} \mid \underbrace{f}_{\substack{\text{функциональный}\\\text{символ}}} \quad T_1 \dots T_n$$

Функциональные символы $\in F$, переменные $\in T$

Пример. $f\ (f_2\ a\ b)\ c$

Определение. **Подстановка переменных** — отображение $S_0 : V \to T$, являющееся тождественным почти всюду[1], то есть $\exists$ фиксированные $a_1 \dots a_n$, для которых S_0 не тождественна: $S_0(a_i) = T_i$, а для $b \notin \{a_i\}$ $S_0(b) = b$.

Тогда можно определить определить подстановку $S : T \to T$:

$$S(f\ T_1 \dots T_n) = f\ (S(T_1))\ (S(T_2)) \dots (S(T_n))$$

$$S(a) = S_0(a)$$

Определение. Рассмотрим уравнение $T_1 = T_2$. Его **решение** — такая подстановка S, что $S(T_1) \equiv S(T_2)$, где $\equiv$ обозначается равенство строк.

Пример.

$$f\ a\ (g\ b) = f\ (g\ c)\ d$$

$$S_0(a) = g\ c \quad S_0(d) = g\ b$$

$$S(f\ a\ (g\ b)) = f\ (g\ c)\ (g\ b)$$

[1] Кроме конечного количества.

Определение (система уравнений).

$$\begin{cases} T_1 = P_1 \\ T_2 = P_2 \\ \vdots \\ T_n = P_n \end{cases}$$

5.1 Эквивалентность уравнений и систем

Определение. Две системы $E_1 : \begin{cases} T_1 = P_1 \\ \vdots \\ T_n = P_n \end{cases}$ и $E_2 : \begin{cases} T_1' = P_1' \\ \vdots \\ T_n' = P_n' \end{cases}$ называются **эквивалентными**, если любое решение системы E_1 подходит к E_2 и наоборот.

Утверждение. Для любой системы существует эквивалентное уравнение.

Доказательство. Выберем новый n-местный функциональный символ h, построим уравнение $h\ T_1 \dots T_n = h\ P_1 \dots P_n$. $\square$

Определение.

$$(U \circ T)(P) = U(T(P))$$

Определение. Определим порядок на подстановках. $S \preceq T$, если S — частный случай T, т.е. $\exists U \colon S = U \circ T$

Определение. **Наиболее общим решением** уравнения $T = P$ назовём подстановку S, такую что $S(T) = S(P)$ и для любой $S_1 \colon S_1(T) \equiv S_1(P)$ выполнено $S_1 \preceq S$

Теорема 9. У уравнения в алгебраических термах $T = P$ всегда есть наиболее общее решение, если есть хоть какое-то.

Определение. **Несовместная система** — система с уравнениями вида $f\ T_1 \dots T_n = g\ P_1 \dots P_k$, где $f \not\equiv g$, либо $x = \dots x \dots$.

В `OCaml` и `Haskell` это называется “occurs check”.

Определение. Система в **разрешённой форме** — система вида $\begin{cases} a_1 = T_1 \\ \vdots \\ a_n = T_n \end{cases}$, где:

1. Все a_i различны
2. T_i не содержит a_j для $i \neq j$

Альтернативное определение — каждый a_i входит по одному разу.

5.2 Алгоритм унификации

Рассмотрим систему $\begin{cases} T_1 = P_1 \\ \vdots \\ T_n = P_n \end{cases}$

Применяем одно из следующих:

1. $x = x$ — отбрасываем.
2. $T = x$, где $T \not\equiv x$, тогда заменяем на $x = T$
3. $\begin{cases} x = P \\ \vdots \\ T_2 = P_2 \\ \vdots \\ T_n = P_n \end{cases} \Rightarrow \begin{cases} T_2[x := P] = P_2[x := P] \\ \vdots \\ T_n[x := P] = P_n[x := P] \\ x = P \end{cases}$
4. $f\ T_1 \dots T_n = f\ P_1 \dots P_n \Rightarrow \begin{cases} T_1 = P_1 \\ \vdots \\ T_n = P_n \end{cases}$

Теорема 10. Применяя шаги алгоритма унификации, за конечное время можно получить систему либо в разрешенной форме, либо несовместную.

Доказательство. Рассмотрим тройку $\left\langle \begin{smallmatrix}\text{количество}\\\text{неразрешенных}\\\text{переменных}\end{smallmatrix}, \begin{smallmatrix}\text{максимальная}\\\text{сложность}\\\text{слева}\end{smallmatrix}, \begin{smallmatrix}\text{количество}\\\text{уравнений}\\\text{максимальной}\\\text{сложности}\\\text{слева}\end{smallmatrix} \right\rangle$. Сложность — вложенность.

1. Применения правил уменьшают тройку.
2. $\langle 0, 0, t \rangle$ — система в разрешенной форме.
3. Количество применений правил конечно, т.к. каждая из троек $\in \omega^3$ и любая последовательность убывающих ординалов конечна.

$\square$

5.3 Вывод типов в $\lambda_\to$

$(\to)$ — 2-местный функциональный символ.

Проведём индукцию по структуре λ-выражения. Результатом разбора будет пара $\langle$система, тип$\rangle$

5.3.1 Построение уравнений

Структура	Комментарий	Система	Тип
x	Введём тип α_x	$\varnothing$	α_x
$A\ B$	Рекурсивный вызов на A и B даст $\langle E_A, \tau_A\rangle, \langle E_B, \tau_B\rangle$. Вводим β — новый тип.	$E_A, E_B, \tau_B \to \beta = \tau_A$	β
$\lambda x.A$	Рекурсивный вызов на A даст $\langle E_A, \tau_A\rangle$. Берём тип для $x : \alpha_x$.	E_A	$\alpha_x \to \tau_A$

Несложно заметить, что эти правила соответствуют правилом вывода в типизированном λ-исчислении.

5.3.2 Разрешение системы

Это унификация.

Пример. Разберём $B = \lambda x.\overbrace{x}^{A}$.

- $E_A = \varnothing, \tau_A = \alpha_x$
- $E_B = \varnothing, \tau_B = \alpha_x \to \alpha_x$

Разрешим систему уравнений τ_A, τ_B. Оказывается, эта система уже в разрешенной форме. Таким образом, $\vdash \lambda x.x : \alpha \to \alpha$. Контекст пустой, т.к. свободных переменных нет ($E_A, E_B = \varnothing$).

Определение. Терм называется **слабо-нормализуемым**, если существует последовательность β-редукций, приводящих его в нормальную форму.

Определение. Терм называется **сильно-нормализуемым**, если не существует бесконечной последовательности β-редукций, нигде не приводящая его в нормальную форму.

Пример. $K\ I\ \Omega$ — слабо нормализуемый, но не сильно нормализуемый

Теорема 11. $\lambda_\to$ сильно нормализуемо.

Примечание. Это сильно ограничивает выразительность $\lambda_\to$.

6 Исчисление предикатов второго порядка

Второй порядок — это когда переменные есть предикаты.

Определение (предикат).

$$\Phi_\Pi ::= p \mid \Phi_\Pi \cup \Phi_\Pi \mid \Phi_\Pi \ \&\ \Phi_\Pi \mid \Phi_\Pi \to \Phi_\Pi \mid \forall p.\Phi_\Pi \mid \exists p.\Phi_\Pi$$

Утверждение. Можно выразить:

$$
\begin{aligned}
a \mathbin{\&} b &::= \forall p.(a \to b \to p) \to p \\
a \vee b &::= \forall p.(a \to p) \to (b \to p) \to p \\
\perp &::= \forall p.p \\
\exists p.A &::= \forall x.(\forall p.p \to x) \to x
\end{aligned}
$$

Это исчисление также называется “Система F”, оно же L_2.

$$L_2 ::= x \mid \lambda x^{\alpha}.A \mid P\ Q \mid P\tau \mid \lambda\alpha.A$$