Теоретические домашние задания

Теория типов, ИТМО, M3235-M3239, осень 2021 года

Домашнее задание №1: «вводная лекция»

1. Напомним правила расстановки скобок в лямбда-выражениях. Лямбда-абстракция ведёт себя жадно: включает всё, что может. Пример: $\lambda z.(\lambda x.a\ b\ c\ \lambda y.d\ e)\ f$ эквивалентно $\lambda z.((\lambda x.(a\ b\ c\ (\lambda y.(d\ e))))\ f)$. В аппликациях скобки расставляются слева направо: $\lambda z.(\lambda x.(a\ b\ c\ (\lambda y.(d\ e))))\ f$ можно преобразовать в $(\lambda z.((\lambda x.((((a\ b)\ c)\ (\lambda y.(d\ e)))))\ f))$.

   (a) Расставьте скобки в выражении: $\lambda z.\lambda x.a\ b\ c\ \lambda y.d\ e\ f$

   (b) Уберите все «лишние» скобки из выражения: $(\lambda f.((\lambda x.(f\ (f\ (x\ (\lambda z.(z\ x))))))\ z))$

   (c) Всегда ли лишние скобки можно убрать единственным образом (когда из результирующего выражения нельзя больше убрать ни одной пары скобок)? Докажите или опровергните.

2. Напомним определения с лекций:

Обозначение	лямбда-терм	название
T	$\lambda a.\lambda b.a$	истина
F	$\lambda a.\lambda b.b$	ложь
Not	$\lambda x.x\ F\ T$	отрицание
And	$\lambda x.\lambda y.x\ y\ F$	конъюнкция

   Проредуцируйте следующие выражения и найдите нормальную форму:

   (a) $T\ F$

   (b) $(T\ Not\ (\lambda t.t))\ F$

   (c) $And\ (And\ F\ F)\ T$

3. Постройте лямбда-выражения для следующих булевских выражений:

   (a) Дизъюнкция

   (b) Штрих Шеффера («и-не»)

   (c) Исключающее или

4. Напомним определения с лекций:

$$f^{(n)}\ X ::= \begin{cases} X, & n = 0 \\ f^{(n-1)}\ (f\ X), & n > 0 \end{cases}$$

Обозначение	лямбда-терм	название
$\overline{n}$	$\lambda f.\lambda x.f^{(n)}\ x$	чёрчевский нумерал
$(+1)$	$\lambda n.\lambda f.\lambda x.n\ f\ (f\ x)$	прибавление 1
$(+)$	$\lambda a.\lambda b.a\ (+1)\ b$	сложение
$(\cdot)$	$\lambda a.\lambda b.a\ ((+)\ b)\ \overline{0}$	умножение

   Используя данные определения, постройте выражения для следующих операций над числами:

   (a) Умножение на 2 ($Mul2$)

   (b) Возведение в степень

   (c) Проверка на чётность

   (d) IsZero: возвращает T, если аргумент равен нулю, иначе F

5. Напомним определения с лекций:

Обозначение	лямбда-терм	название
$MkPair$	$\lambda a.\lambda b.(\lambda x.x\ a\ b)$	создание пары
PrL	$\lambda p.p\ T$	левая проекция
PrR	$\lambda p.p\ F$	правая проекция

   (a) Убедитесь, что $PrL\ (MkPair\ a\ b) \twoheadrightarrow_\beta a$.

(b) Постройте операцию вычитания 1 из числа

(c) Постройте операцию вычитания чисел

(d) Постройте операцию деления на 3 (могут потребоваться пары и/или вычитания)

(e) Постройте операцию деления чисел

(f) Сравнение двух чисел ($IsLess$) — истина, если первый аргумент меньше второго.

6. Существует ли выражение A, что существуют такие выражения B и C, что $A \to_\beta B$ и $A \to_\beta C$, но B и C различны?

7. Будем говорить, что лямбда-выражение находится в нормальной форме, если в нём невозможно провести бета-редукции. Нормальной формой выражения называется результат (возможно, многократной) его бета-редукции, находящийся в нормальной форме. Проредуцируйте выражение и найдите его нормальную форму:

   (a) $\overline{2}\ \overline{2}$

   (b) $\overline{2}\ \overline{2}\ \overline{2}$

   (c) $\overline{2}\ \overline{2}\ \overline{2}\ \overline{2}\ \overline{2}\ \overline{2}\ \overline{2}$

8. Напомним определение Y-комбинатора: $\lambda f.(\lambda x.f\ (x\ x))\ (\lambda x.f\ (x\ x))$. Напомним, что отношение бета-эквивалентности ($=_\beta$) есть транзитивное, рефлексивное и симметричное замыкание отношения бета-редукции. Будем говорить, что выражение не имеет нормальной формы, если никакая конечная последовательность его бета-редукций не приводит к выражению в нормальной форме.

   (a) Покажите, что $Y\ f =_\beta f\ (Y\ f)$.

   (b) Покажите, что выражение $Y\ f$ не имеет нормальной формы;

   (c) Покажите, что выражение $Y\ (\lambda f.\overline{0})$ имеет нормальную форму.

   (d) Покажите, что выражение $Y\ (\lambda f.\lambda x.(IsZero\ x)\ \overline{0}\ (f\ Minus1\ x))\ 2$ имеет нормальную форму.

   (e) Какова нормальная форма выражения $Y\ (\lambda f.\lambda x.(IsZero\ x)\ \overline{0}\ ((+1)\ (f\ (Minus1\ x))))\ \overline{n}$?

   (f) Какова нормальная форма выражения $Y\ (\lambda f.\lambda x.(IsZero\ x)\ \overline{1}\ (Mul2\ (f\ (Minus1\ x))))\ \overline{n}$?

   (g) Определите с помощью Y-комбинатора функцию для вычисления n-го числа Фибоначчи.

9. Определим на языке Хаскель следующую функцию: `show_church n = show (n (+1) 0)` Убедитесь, что `show_church (\f -> \x -> f (f x))` вернёт 2. Пользуясь данным определением и его идеей, реализуйте следующие функции:

   (a) `int_to_church` — возвращает чёрчевский нумерал (т.е. функцию от двух аргументов) по целому числу. Каков точный тип результата этой функции?

   (b) сложение двух чёрчевских нумералов.

   (c) умножение двух чёрчевских нумералов.

   (d) можно ли определить функцию вычитания 1 и вычитания двух чисел? Что получается, а что — нет?

10. Бесконечное количество комбинаторов неподвижной точки. Дадим следующие определения

$$L := \lambda abcdefghijklmnopqstuvwxyzr.r(thisisafixedpointcombinator)$$
$$R := LLLLLLLLLLLLLLLLLLLLLLLLLL$$

   В данном определении терм R является комбинатором неподвижной точки: каков бы ни был терм F, выполнено $R\ F =_\beta F\ (R\ F)$.

   (a) Докажите, что данный комбинатор — действительно комбинатор неподвижной точки.

   (b) Пусть в качестве имён переменных разрешены русские буквы. Постройте аналогичное выражение по-русски: с 32 параметрами (без ё) и осмысленной русской фразой в терме L; покажите, что оно является комбинатором неподвижной точки.

11. Дадим определение: комбинатор — лямбда-выражение без свободных переменных.

    Также напомним определение:

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

    Известна теорема о том, что для любого комбинатора X можно найти выражение P (состоящее только из скобок, пробелов и комбинаторов S и K), что $X =_\beta P$. Будем говорить, что комбинатор P *выражает* комбинатор X в базисе SK.

    Выразите в базисе SK:

    (a) $F = \lambda x.\lambda y.y$
    (b) $\overline{1}$
    (c) Not
    (d) Xor
    (e) InL
    (f) $\overline{n}$

Домашнее задание №2: «формализация лямбда-исчисления»

1. Придумайте грамматику для лямбда-выражений, однозначно разбирающую любое выражение (в частности, учитывающую все сокращения скобок в записи).

2. Приведите пример лямбда-выражения, корректная бета-редукция которого невозможна без переименования связанных переменных. Возможно ли, чтобы в этом выражении все переменные в лямбда-абстракциях были различными?

3. Два выражения A и B назовём родственными, если существует C, что $A \twoheadrightarrow_\beta C$ и $B \twoheadrightarrow_\beta C$. Как соотносится родственность и бета-эквивалентность?

4. Рассмотрим представление лямбда-выражений де Брауна (de Bruijn): вместо имени связанной переменной будем указывать число промежуточных лямбда-абстракций между связывающей абстрацией и переменной. Например, $\lambda x.\lambda y.y\ x$ превратится в $\lambda.\lambda.0\ 1$.

   Докажите, что $A =_\alpha B$ тогда и только тогда, когда представления де Брауна для A и B совпадают. Сформулируйте правила (алгоритмы) для подстановки термов и бета-редукции для этого представления.

5. Как мы знаем, $\Omega \to_\beta \Omega$. А существуют ли такие лямбда-выражения A и B ($A \neq_\alpha B$), что $A \to_\beta B$ и $B \to_\beta A$?

6. Рассмотрим следующие лямбда-выражения для задания алгебраических типов:

Обозначение	лямбда-терм	название
$Case$	$\lambda l.\lambda r.\lambda c.c\ l\ r$	сопоставление с образцом
InL	$\lambda l.(\lambda x.\lambda y.x\ l)$	левая инъекция
InR	$\lambda r.(\lambda x.\lambda y.y\ r)$	правая инъекция

   Сопоставление с образцом — это функция от значения алгебраического типа и двух действий l и r, которая выполняет действие l, если значение создано «левым» конструктором, и r в случае «правого» конструктора. Иными словами, $Case\ l\ r\ c$ — это аналог `case c { InL x -> l x; InR x -> r x }`.

   Заметим, что список (например, целых чисел) — это алгебраический тип:

   `List = Nil | Cons Integer List`.

   Можно сконструировать значение данного типа: `Cons 3 (Cons 5 Nil)`. Можно, например, вычислить его длину:

```
length Nil = 0
length (Cons _ tail) = length tail + 1
```

   Определим $Nil = InL\ 0$, а $Cons\ a\ b = InR\ (MkPair\ a\ b)$. Заметим, что теперь списки могут быть впрямую перенесены в лямбда выражения.

   Определите следующие функции в лямбда-исчислении для списков:

(a) вычисление длины списка;

(b) построение списка длины n из элементов $0, 1, 2, \dots, n-1$;

(c) разворот списка: из списка $a_1, a_2, \dots, a_n$ сделать список $a_n, a_{n-1}, \dots, a_1$;

(d) функцию высшего порядка *map*, которая по функции f и списку $a_1, a_2, \dots, a_n$ строит список $f(a_1), f(a_2), \dots, f(a_n)$.

Решением задачи является полный текст соответствующего лямбда-выражения с объяснениями механизма его работы. Используйте интерпретатор лямбда-выражений *lci* или аналогичный для демонстрации результата.

7. Чёрчевские нумералы соответствуют натуральным числам в аксиоматике Пеано.

   (a) Предложите «двоичные нумералы» — способ кодирования чисел, аналогичный двоичной системе (такой, при котором длина записи числа соответствует логарифму числового значения).

   (b) Предложите реализацию функции $(+1)$ в данном представлении.

   (c) Предложите реализацию лямбда-выражения преобразования числа из двоичного нумерала в чёрчевский.

   Аналогично прошлому заданию, решение должно содержать полный код лямбда-выражения вместе с объяснением механизма его работы.

Домашнее задание №3: «просто-типизированное лямбда исчисление»

1. Оцените временну́ю сложность сложения, вычитания, умножения, деления, вычисления факториала (реализация с помощью Y-комбинатора) для чёрчевских нумералов в лямбда-исчислении.

2. Докажите обитаемость следующих типов с помощью вывода в просто-типизированном лямбда-исчислении и с помощью программы на Хаскеле.

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

   (b) $(\alpha \to \alpha \to \beta) \to (\alpha \to \beta)$

3. Лемма о расширении типа: докажите, что если $\Gamma \vdash A : \tau$, $\Gamma \vdash B : \sigma$, $A \to_\beta B$, то $\Gamma \vdash A : \sigma$.

4. Верно ли, что если $A =_\beta B$, $\vdash A : \tau$ и $\vdash B : \sigma$, то $\vdash B : \tau$? Докажите или приведите опровергающий контрпример.

5. Рассмотрим остальные базовые логические связки (&, $\vee$, $\perp$) и соответствующие им расширенные лямбда-выражения:

$$\frac{\Gamma \vdash A : \varphi \quad \Gamma \vdash B : \psi}{\Gamma \vdash \langle A, B \rangle : \varphi \,\&\, \psi} \qquad \frac{\Gamma \vdash \langle A, B \rangle : \varphi \,\&\, \psi}{\Gamma \vdash \pi_l \langle A, B \rangle : \varphi} \qquad \frac{\Gamma \vdash \langle A, B \rangle : \varphi \,\&\, \psi}{\Gamma \vdash \pi_r \langle A, B \rangle : \psi}$$

$$\frac{\Gamma \vdash A : \varphi}{\Gamma \vdash \mathbf{in}_l\ A : \varphi \vee \psi} \qquad \frac{\Gamma \vdash B : \psi}{\Gamma \vdash \mathbf{in}_r\ B : \varphi \vee \psi} \qquad \frac{\Gamma \vdash L : \varphi \vee \psi \quad \Gamma \vdash f : \varphi \to \tau \quad \Gamma \vdash g : \psi \to \tau}{\mathbf{case}\ L\ f\ g : \tau}$$

$$\frac{\Gamma \vdash A : \perp}{\Gamma \vdash A : \tau}$$

   Конструкции $\langle A, B \rangle$, **case** и им подобные — это новые конструкции для записи выражений: запись $\lambda x.\langle x, \lambda y.\mathbf{in}_l\ x \rangle$ вполне возможна. Конечно, в бестиповом исчислении $\langle A, B \rangle$ может быть задано как терм $\lambda p.p\ A\ B$, но в просто-типизированном исчислении выразительной силы языка для его типизации не хватит, потому мы ввели специальные конструкции со своими специальными правилами типизации. Заметьте, что правила введения лжи не существует (иначе теория станет противоречивой).

   В каждом из подзаданий напишите: (i) лямбда-выражения, доказывающие утверждения, (ii) соответствующие им программы на Хаскеле, (iii) а также покажите, что эти выражения действительно имеют соответствующий тип ($\alpha \to \perp$ обозначим как $\neg\alpha$).

   (a) $\alpha \vee \beta \to \neg(\neg\alpha \,\&\, \neg\beta)$

(b) $\alpha \& \beta \to \neg(\neg\alpha \vee \neg\beta)$

(c) $\alpha \& (\beta \vee \gamma) \to (\alpha \& \beta) \vee (\alpha \& \gamma)$

(d) $(\alpha \to \beta) \to (\neg\beta \to \neg\alpha)$

(e) $\alpha \to \neg\alpha \to \beta$

6. Зафиксируем атомарный тип α и определим η как $(\alpha \to \alpha) \to (\alpha \to \alpha)$. Пусть $\vdash A : \eta \to \eta \to \eta$, $M \equiv \overline{m}$, $N \equiv \overline{n}$, и f — некоторая свободная переменная типа $\alpha \to \alpha$. Покажите, что:

   (a) Сложение и умножение имеют тип $\eta \to \eta \to \eta$ (указание: не каждая реализация этих операций имеет такой тип).

   (b) Любое подвыражение внутри нормальной формы $A\ M\ N\ f$ имеет тип либо α, либо $\alpha \to \alpha$, либо η.

   (c) Тип η внутри нормальной формы $A\ M\ N\ f$ могут иметь только термы M и N.

   (d) Тип $\alpha \to \alpha$ внутри нормальной формы $A\ M\ N\ f$ могут иметь только термы

      i. f;

      ii. $M\ X$ и $N\ X$ при некотором X;

      iii. $\lambda x.S_1\ (S_2\ \ldots (S_k\ y)\ ))$, где $S_i \equiv h$, либо $S_i \equiv M\ X$, либо $S_i \equiv N\ X$ при некотором X.

   (e) Обозначим за $X^n(x)$ n-кратное применение X к аргументу x, а за $P(m,n)$ — полином от m и g. Тогда любое выражение $S : \alpha \to \alpha$, являющееся подвыражением нормальной формы $A\ M\ N\ f$, есть либо полином применений f к аргументу ($S =_\beta \lambda x.f^{P(m,n)}(x)$), либо константная функция ($S =_\beta \lambda x.f^{P(m,n)}(y)$, $x \not\equiv y$).

   (f) Пусть $E(m,n)$ — *расширенный полином* (P_i — натуральные полиномы, c — некоторая натуральная константа):

$$E(m,n) = \begin{cases} P_1(m,n) & m > 0,\ n > 0 \\ P_2(m) & m > 0,\ n = 0 \\ P_3(n) & m = 0,\ n > 0 \\ c & m = 0,\ n = 0 \end{cases}$$

   Покажите, что любая функция, имеющая тип η, вычисляет некоторый расширенный полином. То есть, если $A : \eta \to \eta \to \eta$, то найдётся такой $E(m,n)$, что при всех $m, n \in \mathbb{N}_0$ выполнено $A\ M\ N =_\beta \overline{E(m,n)}$.

Домашнее задание №4: «просто-типизированное лямбда исчисление»

1. Сформулируйте аксиомы для просто типизированного исчисления по Чёрчу. *Указание:* аксиомы должны быть согласованы с типами аргументов лямбда-абстракций.

2. Рассмотрим типизацию по Чёрчу. Определим стирающее преобразование $|\cdot| : \Lambda \to \Lambda_ч$:

$$|A| = \begin{cases} \alpha, & A = \alpha \\ |P||Q|, & A = PQ \\ \lambda x.|P|, & A = \lambda x^\tau.P \end{cases}$$

   Верно ли следующее: если $P \to_\beta Q$ и $|P'| = P$, $|Q'| = Q$, то $P' \to_\beta Q'$.

3. Покажите, что если $A =_\alpha B$ и $\Gamma \vdash A : \tau$, то $\Gamma \vdash B : \tau$ (или, иными словами, доказательство не зависит от выбора пред-лямбда-терма).

Домашнее задание №5: «логика второго порядка, система F»

1. Будем говорить, что связка $\alpha \star \beta$ *выражается* через формулу $F(\alpha,\beta)$, если в правила вывода для связки $\star$ после замены связок на формулы превращаются в теоремы. Как уже было упомянуто на лекции, логические операции могут быть выражены через следующие выражения в логике второго порядка:

$$\alpha \& \beta \equiv \forall\xi.(\alpha \to \beta \to \xi) \to \xi$$

$$\alpha \vee \beta \equiv \forall\xi.(\alpha \to \xi) \to (\beta \to \xi) \to \xi$$

$$\bot \equiv \forall\xi.\xi$$
$$\exists\alpha.\varphi \equiv \forall\xi.(\forall\alpha.\varphi \to \xi) \to \xi$$

Покажите, что правила вывода (аналогичны правилам исчисления высказываний) превратятся в теоремы для:

(a) конъюнкции (3 правила);

(b) дизъюнкции (3 правила);

(c) квантора существования. Данные правила ранее не приводились, поэтому укажем их:

$$\frac{\Gamma \vdash \varphi[p := \theta]}{\Gamma \vdash \exists p.\varphi} \qquad \frac{\Gamma \vdash \exists p.\varphi \quad \Gamma, \varphi \vdash \psi}{\Gamma \vdash \psi}\ p \notin FV(\Gamma)$$

(d) Покажите, что ограничение $p \notin FV(\Gamma)$ существенно в правиле удаления квантора существования.

2. Покажите, что если принять $\langle A, B\rangle^{\alpha\&\beta} \equiv \Lambda\xi.\lambda p^{\alpha\to\beta\to\xi}.p\ A\ B$, то правила для $\alpha \,\&\, \beta$ соответствуют правилам вывода для упорядоченной пары в системе F (правила достаются по наследству из $\lambda_{\to}$).

3. Покажите, что если принять $\mathbf{in}_L A^{\alpha\vee\beta} \equiv \Lambda\xi.\lambda p^{\alpha\to\xi}.\lambda q^{\beta\to\xi}.p\ A$ (аналогично, $\mathbf{in}_R B^{\alpha\vee\beta} \equiv \Lambda\xi.\lambda p^{\alpha\to\xi}.\lambda q^{\beta\to\xi}.q\ B$), то правила для $\alpha \vee \beta$ соответствуют правилам вывода для алгебраического типа в системе F.

4. Пусть чёрчевский нумерал задаётся как $\Lambda\alpha.\lambda f^{\alpha\to\alpha}.\lambda x^{\alpha}.f^n\ x$. Соответственно, целочисленный тип будет $\eta = \forall\alpha.(\alpha \to \alpha) \to (\alpha \to \alpha)$. Покажите, что следующие операции выразимы в системе F (то есть, существует лямбда-выражение $f_\star : \eta \to \eta \to \eta$, что $f_\star\ \overline{m}\ \overline{n} =_\beta \overline{m \star n}$):

   (a) умножение;

   (b) вычитание 1;

   (c) вычитание;

   (d) возведение в степень.

5. Определите (аналогично предыдущему пункту) полиморфный тип для булевского выражения и покажите выразимость в нём:

   (a) отрицания;

   (b) операции `Xor`.

6. Покажите, что Y-комбинатор не типизируется по Карри в системе F.

7. Покажите, что $(\lambda x.x\ x)\ (\lambda z.z\ y\ z)$ типизируется по Карри в системе F.

Домашнее задание №6: «экзистенциальные типы»

1. Покажите, что указанная на лекции реализация `abstype` действительно имеет указанный тип. Покажите, что она действительно соответствует аксиоме для квантора существования в смысле изоморфизма Карри-Ховарда (какой?).

2. Покажите, что указанная на лекции реализации `pack` действительно имеет указанный тип. Покажите, что она действительно соответствует аксиоме для квантора существования в смысле изоморфизма Карри-Ховарда (какой?).

3. Рассмотрите реализацию экзистенциальных типов на Хаскеле (файл `existential.hs` в текущем репозитории). Модифицируйте её для реализации *очереди с приоритетами*: должны быть предусмотрены функции для создания пустой очереди, добавления целого числа с приоритетом (также целым числом), взятия самого первого целого числа с минимальным приоритетом (предусмотрите случай пустой очереди), проверки пустоты очереди. Дайте две реализации данного АТД (простую и эффективную), и также приведите пример использования типа.

4. Рассмотрим АТД `Counter`, имеющий текущее состояние, функции увеличения, уменьшения и проверки на ноль: $\sigma \equiv \exists\alpha.\alpha \,\&\, (\alpha \to \alpha) \,\&\, (\alpha \to \alpha) \,\&\, (\alpha \to \mathtt{Bool})$. Определите значение («конструктор») `createZero` $: \sigma$ и функции («методы») `inc` $: \sigma \to \sigma$, `dec` $: \sigma \to \sigma$, `isZero` $: \sigma \to$ `Bool`. Реализуйте данный АТД на Хаскеле двумя разными способами, а также создайте список из разнородных (по разному реализованных) счётчиков и продемонстрируйте, что он ведёт себя ожидаемым образом. Главное отличие от предыдущего задания: мы не пытаемся вызывать код пользователя из `abstype` (мы не работаем с модулем, реализующим АТД), вместо этого мы открываем и обратно упаковываем экзистенциальный тип при каждом вызове функции («посылаем сообщения» АТД).