Теоретические домашние задания

Теория типов, ИТМО, М3235-М3239, 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 $0, 1, 2, \ldots, n-1$;

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

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

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 ($\&$, $\vee$, $\bot$) и соответствующие им расширенные лямбда-выражения:

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

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

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

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 ($\alpha \to \bot$ 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 $|\cdot| : \Lambda \to \Lambda_{\text{ч}}$:

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

   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: $\sigma \equiv \exists\alpha.\alpha\ \&\ (\alpha \to \alpha)\ \&\ (\alpha \to \alpha)\ \&\ (\alpha \to \texttt{Bool})$. Определите его на Хаскеле аналогично предыдущему пункту. Далее, определите значение («конструктор») $\texttt{createZero} : \sigma$ и функции («методы») $\texttt{inc} : \sigma \to \sigma$, $\texttt{dec} : \sigma \to \sigma$, $\texttt{isZero} : \sigma \to \texttt{Bool}$. 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).

Домашнее задание №7: «типовая система Хиндли-Милнера»

1. *О выразительной силе HM.* Заметим, что список — это «параметризованные» числа в аксиоматике Пеано. Число — это длина списка, а к каждому штриху мы присоединяем какое-то значение. Операции добавления и удаления элемента из списка — это операции прибавления и вычитания единицы к числу.

   Рассмотрим тип «бинарного списка» (расширение Окамля):

   ```
   type 'a bin_list = Nil | Zero of (('a*'a) bin_list) | One of 'a * (('a*'a) bin_list);;
   ```

   и операцию добавления элемента к списку:

   ```
   let rec add elem lst = match lst with
       Nil -> One (elem,Nil)
     | Zero tl -> One (elem,tl)
     | One (hd,tl) -> Zero (add (elem,hd) tl)
   ```

   (a) Какой тип имеет `add` (обратите внимание на ключевое слово `rec`: для точного указания соответствующего лямбда-выражения и вывода типа необходимо использовать Y-комбинатор)? Считайте, что семейство типов `bin_list 'a` предопределено и обозначается как τ_α. Выразим ли этот тип в системе Хиндли-Милнера?

   (b) Реализуйте предложенный тип и функцию `add` на Хаскеле (используйте опцию `RankNTypes`). Также реализуйте функцию для удаления элемента списка (головы).

   (c) Предложите функцию для эффективного соединения двух списков (источник для вдохновения — сложение двух чисел в столбик).

   (d) Предложите функцию для эффективного выделения n-го элемента из списка.

2. На занятии мы рассмотрели функцию `strange_pair x = (x 1, x "a")`. Покажите, что данную функцию невозможно типизировать в типовой системе Хиндли-Милнера. Указания: (а) ограничение мономорфизма отношения к делу не имеет; (б) ограничение на правило введения квантора всеобщности может оказаться существенным.

3. Покажем, что алгоритм W действительно находит корректный тип для лямбда-выражения (доказательство, что он находит наиболее общий тип, мы оставим в стороне). Для этого докажем по индукции, что $W(\Gamma, X)$ действительно находит такие тип τ и подстановку S, что $S(\Gamma) \vdash X : \tau$:

   (a) покажите базу индукции: $W(\Gamma, x)$;

   (b) покажите случай аппликации: $W(\Gamma, P\ Q)$;

   (c) покажите случай лямбда-абстракции: $W(\Gamma, \lambda x.P)$;

   (d) покажите случай `let`-выражения: $W(\Gamma, \texttt{let}\ x = P\ \texttt{in}\ Q)$.

4. Покажите, что в Хаскеле выражается $Y : \forall\alpha.(\alpha \to \alpha) \to \alpha$ и правило исключённого третьего $E : \alpha \vee \neg\alpha$.

5. Возможно ли в C++ построить выражения с типами ранга два и выше (включая конструкции с темплейтами)? Приведите пример, если да.

Домашнее задание №8: «обобщённые типовые системы»

1. Укажите тип (род) в исчислении конструкций для следующих выражений (при необходимости определите типы используемых базовых операций и конструкций самостоятельно) и докажите его:

   (a) Функция возведения целого числа в квадрат: `sq x = x * x`

   (b) `sizeof`

   (c) `std::map`

   (d) Монада `ST` из Хаскеля

   (e) Пусть задано выражение рода **nonzero** : $\star \to \star$, выбрасывающее нулевой элемент из типа. Например, **nonzero unsigned** — тип положительных целых чисел. Определите, каков в коде

```
template<typename T, T x>
struct NonZero { const static std::enable_if_t<x != T(0), T> value = x; };
```

будет тип (род) поля value.

2. Приведём следующее странное рассуждение: если мы рассмотрим правый нижний дальний угол лямбда-куба, соответствующий $S = \{\langle\star,\star\rangle, \langle\star,\Box\rangle, \langle\Box,\star\rangle\}$, то можем заметить, что теоретически возможно существование функций, отображающих тип в значение — а потом значение в тип (например, по типу вернуть его название в строке, изменить его, а потом по изменённому названию построить другой тип).

   Поясните, почему тем не менее необходимо существование случая $\langle\Box,\Box\rangle$ в аксиоматике, почему всё равно мы не сможем формально построить функции рода $\Pi x^\star.F\ x$ в такой теории.

3. Предложите выражение на языке C++ (возможно, использующее шаблоны), имеющее следующий род (тип):

   (a) $\mathbf{int} \to (\star \to \star)$

   (b) $(\star \to \mathbf{int}) \to \star$

   (c) $\Pi x^\star.n^{\mathbf{int}}.F(n,x)$, где
   $$F(n,x) = \begin{cases} \mathbf{int}, & n = 0 \\ x \to F(n,x), & n > 0 \end{cases}$$

4. Аналогично типу Π, мы можем ввести тип Σ, соответствующий квантору существования в смысле изоморфизма Карри-Ховарда.

   (a) Определите правила вывода для Σ в обобщённой типовой системе (воспользуйтесь правилами для экзистенциальных типов в системе F).

   (b) Укажите способ выразить Σ через Π (также воспользуйтесь идеями для системы F).

5. Рассмотрим классы типов в Хаскеле (например, `Num`). Каким образом их можно представить в обобщённой типовой системе? Как формализовать запись типа функции `f :: Num a => a -> a`?

Домашнее задание №9: «непрерывность; связность; равенство в гомотопической теории типов»

1. Непрерывна ли функция $f : (0,1) \cup (2,3) \to [0,3] = \lambda x.x$?

2. Докажите, что если $f : \mathbb{R} \to \mathbb{R}$ и
   $$\forall\varepsilon > 0.\exists\delta > 0.\forall x, x_0 \in \mathbb{R}.|x - x_0| < \delta \to |f(x) - f(x_0)| < \varepsilon$$
   то f — непрерывна.

3. Как связана непрерывность некоторой функции f и обратной к ней (f^{-1}), следует ли одна из другой?

4. Рассмотрим «обратную непрерывность»: f обратно-непрерывна, если образ открытого всегда открыт. Как связаны непрерывность и обратная непрерывность, следует ли одна из другой?

5. Пусть X — множество с антидискретной топологией (открыты X, $\varnothing$ и множество X без конечного количества любых точек: $X \setminus \{a_0, a_1, \ldots, a_{n-1}\}$ открыто при всех n и $a_0, \ldots, a_{n-1}$).

   (a) Любое ли подмножество X связно?

   (b) Любое ли подмножество X линейно-связно?

6. Постройте пример связного, но не линейно-связного множества.

7. Рассмотрим дерево с топологией замкнутости по отношению наследования (топология, предлагаемая моделями Крипке).

   (a) Приведите пример непрерывных функций из такого дерева в $\mathbb{R}$ и обратно (если они есть).

   (b) Приведите пример разрывных функций из такого дерева в $\mathbb{R}$ и обратно (если они есть).

   (c) Верно ли, что в такой топологии линейная связность эквивалентна обычной?

8. Докажите, что любое линейно-связное множество связно.

9. Рассмотрим множество рациональных чисел: это $\mathbb{Z}^2$ с отношением эквивалентности ($\langle p_x, q_x\rangle \approx \langle p_y, q_y\rangle$ если $p_x \cdot q_y = p_y \cdot q_x$). Какую топологию надо задать на $\mathbb{Z}^2$, чтобы $a = b$ в смысле HOTT совпало бы с $a \approx b$? Напомним, в гомотопической теории типов $a = b$ если существует непрерывное отображение $f : [0, 1] \to X$, что $f(0) = a$ и $f(1) = b$.

10. Пусть задан тип X, населённый значениями a, b и c, и пусть существуют пути $a \rightsquigarrow b$ и $b \rightsquigarrow c$ (путь здесь — непрерывное отображение отрезка $[0, 1]$). Покажите, что $b \rightsquigarrow a$, $a \rightsquigarrow c$ и $f(a) \rightsquigarrow f(b)$ для любой непрерывной функции f.

11. Приведите примеры связных и несвязных типов в языках программирования. Как на этих типах задана топология?

Домашнее задание №10: «задачи на простые доказательства в Arend»

1. Докажите `repl (A B : \Type) (eqProof : A = B) : A -> B` (если два типа равны, то существует функция между ними).

2. Докажите коммутативность умножения: то есть обитаемость типа $\Pi x^{\texttt{Nat}}.\Pi y^{\texttt{Nat}}.x \cdot y = y \cdot x$

3. Докажите дистрибутивность: $a \cdot (b + c) = a \cdot b + a \cdot c$.

4. Определим понятие инъективной функции:

   ```
   \func inj (A B : \Type) (f : A -> B) => \Pi (a b : A) (eq : f a = f b) -> a = b
   ```

   (a) Докажите, что `inj Nat Nat suc` обитаемо — т.е. `suc` инъективна.

   (b) Докажите, что если $f : A \to B$ и $g : C \to D$ инъективны, то $\lambda p.\langle f\ \pi_L(p), g\ \pi_R(p)\rangle$ тоже инъективна.

   (c) Докажите, что x^2 инъективна.

   (d) Докажите, что если f инъективна и $f(x) \neq f(y)$, то $x \neq y$.

5. Докажите, что $0 \neq x + 1$.

6. Докажите, что если $a + b = a + c$, то $b = c$. Также докажите, что если $a + b \neq a + c$, то $b \neq c$.

7. Докажите, что если $x = 2 \cdot x$, то $x = 0$.

8. Докажите, что если $x^2 = x^3$, то $x = 0$ или $x = 1$ (логическое или соответствует типу `Or` из модуля `Data.Or`).

9. Докажите, что если $a + p = c$ и $a + q = c$, то $p = q$ (логическое и соответствует упорядоченной паре — типу `\Sigma`).

10. Определите функцию ограниченного вычитания:

$$S(a, b) = \begin{cases} a - b, & a \geq b \\ 0, & a < b \end{cases}$$

    Покажите, что $S(a + b, b) = a$.

11. Определите операцию 2^n и покажите, что $2^{a+b} = 2^a \cdot 2^b$.

Домашнее задание №11: «немного математических доказательств»

1. Рассмотрим класс `DistributiveLattice` из стандартной библиотеки.

   (a) Определите импликативную решётку. Докажите, что в импликативной решётке есть 1. Докажите, что любая импликативная решётка дистрибутивна.

   (b) Определите алгебру Гейтинга и докажите, что $\forall a.\forall b.(a \to b) \to (a \to \neg b) \to \neg a$.

   (c) Определите булеву алгебру и докажите, что $\forall a.\neg\neg a \to a$.

   (d) Докажите, что если в решётке есть диамант (существуют такие вершины a, b, c, d, e, что $a \preceq b$, $a \preceq c$, $a \preceq d$, $b \preceq e$, $c \preceq e$, $d \preceq e$, причём b, c, d несравнимы между собой), то решётка не является импликативной.

2. Рассмотрим класс `TopSpace` из стандартной библиотеки.
   (a) Покажите, что пустое множество — открытое (т.е. принадлежит топологии).
   (b) Покажите, что топологическое пространство является решёткой.
   (c) Покажите, что топологическое пространство является алгеброй Гейтинга.
   (d) Покажите, что дискретная топология является топологией.
   (e) Покажите, что топология стрелки на $\mathbb{N}$ (открыты все лучи вида $[x, \infty)$) является топологией.
3. Рассмотрите класс «группа» из стандартной библиотеки (`Algebra.Group.Group`).
   (a) Докажите, что в нём нейтральный элемент `ide` единственен.
   (b) Определите понятие «гомоморфизм». Определите понятие «ядро гомоморфизма». Докажите, что в тождественном гомоморфизме группы на себя ядро гомоморфизма состоит только из нейтрального элемента.
4. Рассмотрим перестановки натуральных чисел.
   (a) Определите тип «перестановка чисел от 0 до $n-1$». Докажите, что существует всего две перестановки чисел $(0, 1)$.
   (b) Определите композицию перестановок. Покажите, что существует нейтральный элемент для перестановки.
   (c) Покажите, что перестановки образуют группу (`Group`).

Домашнее задание №12: «Универсумы, гомотопические уровни, фактор-множества»

1. Пусть $\langle p_1, q_1 \rangle \approx \langle p_2, q_2 \rangle$, если $p_1 + q_2 = p_2 + q_1$.
   (a) Определите целые числа `AltZ` как $\mathbb{Z}/\approx$, покажите, что они образуют кольцо (`Ring`).
   (b) Покажите, что существует биекция `AltZ` на `Int`.
   (c) Покажите, что равенство на `AltZ` разрешимо.
2. Покажите, что тип с разрешимым равенством — `Set`.
3. Пусть $f : G \to H$ — гомоморфизм групп. Покажите, что H изоморфен $G/\ker f$ (гомоморфный образ группы изоморфен фактор-группе по ядру гомоморфизма).
4. Есть ли какие-то аксиомы теории множеств в аксиоматике Цермело-Френкеля (не рассматривайте аксиомы выбора и подстановки), которые не выполнены для `Set`?
5. Определите понятие мощности множества (`Set`) в Аренде. Покажите, что мощность множества `Bool` равна 2.
6. Рассмотрим следующие определения:

```
\func ChurchT (x : \Type) => (x -> x) -> (x -> x)
\func Church => \Pi (x : \Type) -> ChurchT x
\func Zero : Church => \lam t f x => x

\func inc (n : Church) : Church => \lam t f x => n t f (f x)
\func add (a : Church) (b : Church) : Church => \lam t f x => a t f (b t f x)
```

   И так далее. Возможно ли определить (более-менее следуя реализации операций для бестипового исчисления, развивая определения выше):
   (a) Операцию «деление на три» (естественно, в версии, не использующей Y-комбинатор)?
   (b) Операцию возведения в степень, определявшуюся как $\lambda m.\lambda n.n\ m$?

Домашнее задание №13: «Теорема Диаконеску»

1. Определите сетоид в Аренде и постройте сетоид для кольца вычетов по модулю k (целые числа, факторизованные по отношению сравнимости по модулю).

2. Сформулируйте на языке Аренд и докажите теорему Диаконеску для сетоидов.

3. Покажите, что `Nat` — вполне упорядоченное отношением ($\leq$) множество.

4. Пусть в аксиоме выбора из стандартной библиотеки Аренда все множества вида $B\ x$ суть разрешимые подмножества `Nat` (равенство разрешимо, и предикат, проверяющий принадлежность элемента подмножеству, тоже разрешим). Достаточно ли этого для доказуемости аксиомы выбора в данном случае?

5. Приведите пример семейства множеств, на котором аксиома выбора из стандартной библиотеки не выполнена.

6. Определим аксиому выбора для отношений: для любого отношения $R \subseteq A \times B$, такого, что $\forall x.x \in A \to \exists y.y \in B\ \&\ R(x,y)$, существует подотношение $R' \subseteq R$, что $\forall x.x \in A \to \exists! y.y \in B\ \&\ R'(x,y)$. Здесь $\exists! x.\alpha(x)$ — сокращение для утверждения «существует единственный»: $(\exists x.\alpha(x))\&(\forall x.\forall y.\alpha(x)\& \alpha(y) \to x = y)$

   (a) Покажите, что аксиома выбора для сетоидов влечёт аксиому выбора для отношений.

   (b) Пусть отношение R на `Nat` разрешимо. Покажите, что на нём доказуема аксиома выбора для отношений.

   (c) Доказуема ли в Аренде аксиома выбора для отношений?

7. Верно ли, что аксиома выбора для сетоидов влечёт аксиому выбора из стандартной библиотеки Аренда?

8. Верно ли, что аксиома выбора из стандартной библиотеки Аренда влечёт аксиому выбора для сетоидов?

Домашнее задание №14: «Пропозициональное обрезание, линейные типы»

1. Пропозиционально обрезанные типы (`TruncP`) требуют специальных условий для элиминации `inP`. А именно, тип выражения, внутри которого проводится элиминация, должен иметь гомотопический уровень -1 (то есть, принадлежать универсуму `\Prop`).

   Далеко не для всех типов очевидно, что они имеют гомотопический уровень -1. Например, тип `Dec`. Обратим внимание на два обстоятельства:

   (a) тип `Dec` — это функция из типа в тип, вспомогательная абстрактная конструкция; типом же, интересующим всех, является `Dec P` — результат применения данной функции;

   (b) каково бы ни было утверждение P, в типе `Dec P` всегда единственный представитель: либо `yes P`, если P обитаем, либо `no (P -> Empty)`, если P необитаем. Если бы оба эти варианта имели бы место одновременно, то Аренд был бы противоречив.

   Для таких неочевидных случаев необходимо указывать специальную функцию, которая снабжается префиксом `\use \level`; данная функция указывается как часть определяемого типа данных. Например, в следующем типе «простое число» данная функция должна доказать, что любые два значения типа при данном x равны:

```
\data Div3 (x : Nat) \elim x
  | remainder-zero (Exists (p : Nat) (p Nat.* 3 = x))
  | remainder-one (Exists (p : Nat) (p Nat.* 3 Nat.+ 1 = x))
  | remainder-two (Exists (p : Nat) (p Nat.* 3 Nat.+ 2 = x))
  \where \use \level levelProp {x : Nat} (a b : Div3 x) : a = b => {?}
```

   При необходимости (например, если функция `\use \level` не указана) можно специально указать на доказательство пропозициональности результата, воспользовавшись функцией `TruncP.rec`.

(a) Замените `{?}` в тексте выше на корректное доказательство.

(b) Определите функцию, которая бы по x и значению $\exists pq.3 \cdot p + q = x \,\&\, 0 \leq q < 3$ возвращала бы `Div3 x` (понятно, можно разделить x на 3, но нам уже результат деления дали вторым аргументом — задача в том, чтобы им воспользоваться).

(c) Постройте аналогичный тип `Prime x` для простых чисел — с тремя вариантами `less-than-two`, `is-prime`, `is-composite` — и определите функцию, строящую по числу данный тип.

(d) Покажите, что в типе `Prime (x*x + 2*x + 1)` всегда (кроме граничных случаев) обитает вариант `is-composite`.

(e) Покажите, что в типе

```
\data SuperDec (P : \Prop)
| sure P
| nope (P -> Empty)
| neither ((P || (P -> Empty)) -> Empty)
\where \use \level superDecProp {P : \Prop} (a b : SuperDec P): a = b => {?}
```

вариант `neither` невозможен (также, заполните пропущенное доказательство `superDecProp`).

2. Покажите, что можно построить функцию, которая из значений `x : Nat` и `p : Exists (p : Nat) (p * p = x)` получает `\Sigma (p : Nat) (p * p = x)` (мы здесь должны существенно использовать единственность натурального корня числа).

3. Покажите, что вложения интуиционистской логики в линейную соответствуют правилам вывода для интуиционистской логики.

4. Приведите пример структур данных для C++, соответствующих связкам линейной логики (для линейных значений используйте уникальные указатели или аналогичные уникальные объекты). Покажите, что правила вывода выполнены.

5. Формализуйте функции доступа к массиву целых чисел: укажите их тип в линейной логике. Функции должны быть следующими: прочесть элемент массива, изменить элемент массива, узнать длину массива, применить `const` функцию к каждому элементу массива, обновить при помощи функции каждый элемент массива, не меняющая массив свёртка (fold).

6. В статье Вадлера структурное правило перестановки гипотез указано так:

$$\frac{\Gamma, \Delta \vdash \alpha}{\Delta, \Gamma \vdash \alpha}$$

В ходе лекции было предложено вместо него рассмотреть правило с боковыми формулами:

$$\frac{\Psi, \Gamma, \Delta, \Xi \vdash \alpha}{\Psi, \Delta, \Gamma, \Xi \vdash \alpha}$$

Является ли правило с боковыми формулами необходимым? Достаточно ли формулировки Вадлера для получения любой перестановки гипотез?