Теоретические (“малые”) домашние задания

Теория типов, ИТМО, M3334-M3339, осень 2019 года

Домашнее задание №1: «знакомство с лямбда-исчислением»

1. Расставьте скобки:
   (a) $\lambda x.x\ x\ \lambda x.x\ x$
   (b) $(\lambda x.x\ x)\ \lambda x.x\ x$
   (c) $\lambda x.(x\ x)\ \lambda x.x\ x$
   (d) $\lambda f.\lambda x.f f f x$

2. Проведите бета-редукции и приведите выражения к нормальной форме:
   (a) $(\lambda a.\lambda b.a)\ (\lambda a.\lambda b.a)\ (\lambda a.\lambda b.b)$
   (b) $(\lambda a.\lambda b.a)\ b$
   (c) $(\lambda f.\lambda x.f\ (f\ x))\ (\lambda f.\lambda x.f\ (f\ x))$

3. Выразите следующие функции в лямбда-исчислении:
   (a) Or, Xor;
   (b) тернарная операция в Си (`?:`);
   (c) isZero (T, если аргумент равен 0, иначе F);
   (d) isEven (T, если аргумент чётный);
   (e) умножение на 2;
   (f) умножение;
   (g) возведение в степень;
   (h) упорядоченная пара. К паре должны прилагаться три лямбда-выражения (M, P_l, P_r): выражение M по двум значениям строит упорядоченную пару, а выражения P_l и P_r возвращают первый и второй элемент упорядоченной пары соответственно.
   Убедитесь, что для ваших выражений выполнено
   $$P_l\ (M\ A\ B) \to_\beta \cdots \to_\beta A$$
   и
   $$P_r\ (M\ A\ B) \to_\beta \cdots \to_\beta B$$
   (i) вычитание 1;
   (j) вычитание;
   (k) сравнение («меньше»);
   (l) деление.

4. Назовём бета-эквивалентностью транзитивное, рефлексивное и симметричное замыкание отношения бета-редукции, будем записывать его как $(=_\beta)$. В частности, бета-эквивалентны те термы, которые имеют одинаковую нормальную форму. Также, нетрудно заметить следующее:
   (a) $And\ T\ F =_\beta F$;
   (b) $\Omega =_\beta \Omega$;
   (c) $(\lambda n.\lambda f.\lambda x.n\ f\ (f\ x))\ \overline{n} =_\beta \overline{n+1}$;
   (d) $a \neq_\beta b$.

   Мы будем говорить, что лямбда-выражение E выражает функцию $f(x_1, \ldots, x_k) : \mathbb{N}_0^k \to \mathbb{N}_0$, если при любых $x_1, \ldots, x_k \in \mathbb{N}_0$ выполнено
   $$E\ \overline{x_1}\ \ldots\ \overline{x_k} =_\beta \overline{f(x_1, \ldots, x_k)}$$
   Какие функции выражают следующие выражения? Ответ обоснуйте.

(a) $\lambda m.\lambda n.n\ m$;
(b) $\lambda m.\lambda n.\lambda x.n\ (m\ x)$.

5. *Ненормализуемым* назовём лямбда-выражение, не имеющее нормальной формы, то есть выражение, для которого нет конечной последовательности бета-редукций, приводящей к нормальной форме. *Сильно нормализуемым* назовём лямбда-выражение, для которого не существует бесконечной последовательности бета-редукций (любая последовательность бета-редукций неизбежно заканчивается нормальной формой, если её продолжать достаточно долго). *Слабо нормализуемым* назовём лямбда-выражение, которое имеет нормальную форму, но существует бесконечная последовательность бета-редукций, которая не приводит его в нормальную форму. Приведите примеры сильно нормализуемого, слабо нормализуемого и ненормализуемого лямбда-выражения.

Домашнее задание №2: «теорема Чёрча-Россера, Y-комбинатор»

1. *Полное доказательство теоремы Чёрча-Россера.* На лекции был представлен план доказательства теоремы, в котором необходимо заполнить пустоты.

   (a) Покажите, что отношение бета-редукции — подотношение отношения параллельной бета-редукции. В символической записи, $(\to_\beta) \subseteq (\rightrightarrows_\beta)$. То есть, если $A \to_\beta B$, то $A \rightrightarrows_\beta B$.

   (b) Покажите, что каковы бы ни были термы A, P, Q и переменная x, если выполнено $P \rightrightarrows_\beta Q$, то $A[x := P] \rightrightarrows_\beta A[x := Q]$. Убедитесь, что это справедливо и если x не входит свободно в A.

   (c) Покажите, что каковы бы ни были термы A, B, P, Q и переменная x, если $A \rightrightarrows_\beta B$ и $P \rightrightarrows_\beta Q$, то $A[x := P] \rightrightarrows_\beta B[x := Q]$.

   (d) Покажите, что $(\rightrightarrows_\beta)$ обладает ромбовидным свойством.

   (e) *Транзитивным и рефлексивным замыканием* отношения $R \subseteq U^2$ назовём такое отношение $R^* \subseteq U^2$, что $(x, y) \in R^*$ тогда и только тогда, когда существует $n \in \mathbb{N}$ и последовательность $a_1, \ldots, a_n \in U$, что: $a_1 = x$, $a_n = y$ и $(a_i, a_{i+1}) \in R$.

   Покажите, что если R — некоторое отношение, обладающее ромбовидным свойством, то R^* тоже обладает ромбовидным свойством.

   (f) Покажите, что каковы бы ни были отношения R и S, если $R \subseteq S$, то $R^* \subseteq S^*$. В частности, покажите, что $(\twoheadrightarrow_\beta) \subseteq (\rightrightarrows_\beta)^*$.

   (g) Покажите, что $(\rightrightarrows_\beta)^* \subseteq (\twoheadrightarrow_\beta)$

   На основании доказанных лемм несложно показать утверждение теоремы Чёрча-Россера: из последних пунктов следует, что $(\rightrightarrows_\beta)^* = (\twoheadrightarrow_\beta)$, а из пункта (d) — что это отношение обладает ромбовидным свойством.

2. Реализуйте следующие функции с помощью Y-комбинатора, вычисляющие:

   (a) факториал числа;
   (b) n-е простое число;
   (c) функцию Аккермана;
   (d) частичный логарифм.

3. *Отмеченным объединением* множеств $L \uplus R$ назовём множество пар

$$U = \{\langle 1, x\rangle \mid x \in L\} \cup \{\langle 2, y\rangle \mid y \in R\}$$

   Соответственно, для данного множества мы можем определить три функции

название	обозначение	определение
левая инъекция	$in_L : L \to U$	$in_L(x) = \langle 1, x\rangle$
правая инъекция	$in_R : R \to U$	$in_R(x) = \langle 2, x\rangle$
выбор	$Case : U \times (L \to X) \times (R \to X) \to X$	$Case(u, f, g) = \begin{cases} f(x), & \text{если } u = \langle 1, x\rangle \\ g(x), & \text{если } u = \langle 2, x\rangle \end{cases}$

   Говоря простыми словами, инъекции приписывают к значению цифру 1 или 2, получая значение из множества U, а выбор, основываясь на приписанной цифре, применяет к значению первую или вторую фукнцию.

   Построим аналогичную конструкцию для типов. *Типом-суммой* типов L и R (или, иначе, *алгебраическим типом*) назовём тип данных U, хранящий значения либо типа L, либо типа R, причём всегда точно известно, какого именно (сравните с определением дизъюнкции в интуиционистской логике). С точки зрения теории множеств, множество значений типа U — отмеченное объединение множеств значений типов L и R. Для этого типа существует три базовых операции: две инъекции и выбор. Данный тип данных довольно широко распространён, и присутствует в ограниченном объёме даже в языках Си и Паскаль.

Например, в языке Паскаль с возможно следующее определение (там данная конструкция называется «записью с вариантами»):

```
type value: record
    is_real: boolean;
    case is_real of
        false: (real_value: real);
        true:  (int_value: integer);
    end;
```

Данная запись если `is_real = true` содержит поле `int_value`, а если `false` — поле `real_value`. Реализация данной структуры предполагает, что оба эти поля расположены в одной памяти.

В языке Си аналогом этой структуры является объединение (`union`), однако, явного поля для выбора одного из вариантов там не предусмотрено. В языке Си++ довольно близким аналогом является класс `std::variant` — «безопасное» объединение.

В лямбда-исчислении оказывается возможно реализовать эту конструкцию в чистом математическом виде:

$$\begin{array}{l} In_L = \lambda x.\lambda f.\lambda g.f\ x \\ In_R = \lambda x.\lambda f.\lambda g.g\ x \\ Case = \lambda u.\lambda f.\lambda g.u\ f\ g \end{array}$$

Также ещё заметим, что список можно представить, как алгебраический тип с двумя вариантами:

- `Nil` (соответствует пустому списку)
- `Cons(h,t)` (соединение головы списка `h` и хвоста `t`)

В частности, можно записать список `[1,2,3]` как `Cons(1,Cons(2,Cons(3,Nil)))`.

В лямбда-исчислении мы можем представить `Cons(h,t)` как правую инъекцию упорядоченной пары $\langle h, t\rangle$ (так будем обозначать выражение $\lambda a.a\ h\ t$), а `Nil` — как левую инъекцию любого значения, например, F. Тогда список `[1,2,3]` может быть представлен следующим лямбда-выражением:

$$In_R\ \langle\overline{1}, In_R\ \langle\overline{2}, In_R\ \langle\overline{3}, In_L\ F\rangle\rangle\rangle$$

(a) Реализуйте конструкции In_L, In_R, $Case$ на языках Си, Паскаль и Си++ как можно ближе к формальному определению.

(b) Покажите, что $Case\ (In_L\ \overline{0})\ (\lambda x.p)\ (\lambda x.q) =_\beta p$ и $Case\ (In_R\ q)\ (\lambda x.p)\ (\lambda x.x) =_\beta q$.

(c) Постройте лямбда-выражение, по чёрчевскому нумералу $\overline{n}$ возвращающее список `[1,2,3,...,n]`.

(d) Постройте лямбда-выражение, по списку возвращающее его длину.

(e) Постройте лямбда-выражение, суммирующее список чёрчевских нумералов.

(f) Покажите, как реализовать алгебраический тип на n вариантов.

(g) Покажите, как реализовать обращение списка (функция должна вернуть список в обратном порядке).

4. Чёрчевские нумералы соответствуют аксиоматике Пеано (числа записываются путём приписываний штрихов — прибавлений единиц). В частности поэтому вся арифметика с ними крайне медленная. А можно ли реализовать их с использованием двоичной записи?

    (a) Предложите, как можно реализовать «логарифмические» нумералы — значения, которые соответствовали бы двоичной записи чисел.

    (b) Определите операцию преобразования чёрчевского нумерала в логарифмический.

    (c) Определите операцию преобразования логарифмического нумерала в чёрчевский.

    (d) Определите операцию суммы логарифмических нумералов.

    (e) Определите операцию ограниченного вычитания единицы из логарифмического нумерала (напомним, ограниченное вычитание возвращает 0, если вычитаемое больше уменьшаемого).

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

1. На прошлой лекции определение параллельной бета-редукции было сформулировано неточно, отчего следующее утверждение не могло быть доказано:

   Покажите, что каковы бы ни были термы A, B, P, Q и переменная x, если $A \rightrightarrows_\beta B$ и $P \rightrightarrows_\beta Q$, то $A[x := P] \rightrightarrows_\beta B[x := Q]$.

   Однако, данное утверждение можно доказать, если переформулировать параллельную бета-редукцию так. $A \rightrightarrows_\beta B$, если:

   (a) $A = x$, $B = y$ и $x = y$
   (b) $A = P\ Q$, $B = R\ S$ и $P \rightrightarrows_\beta R$, $Q \rightrightarrows_\beta S$
   (c) $A = \lambda x.P$, $B = \lambda x.Q$ и $P \rightrightarrows_\beta Q$
   (d) $A = (\lambda x.P)\ Q$, $B = R[x := S]$ и $P \rightrightarrows_\beta R$, $Q \rightrightarrows_\beta S$

   В связи с этим:

   (a) Докажите утверждение из прошлого домашнего задания при заданном определении.
   (b) Обладает ли исходное отношение параллельной бета-редукции (заданное на прошлой лекции) ромбовидным свойством? Возможно, вы можете привести для него контрпример?

2. Покажите, что если $A \twoheadrightarrow_\beta B$ и $\vdash A : \alpha$, то $\vdash B : \alpha$.

3. Верно ли, что если $A \twoheadrightarrow_\beta B$ и $\vdash B : \alpha$, то $\vdash A : \alpha$? Верно ли это свойство для исчисления по Чёрчу?

4. Покажите, что комбинатор $\Omega = (\lambda x.x\ x)\ (\lambda x.x\ x)$ не имеет типа.

5. Покажите, что никакое лямбда-выражение не имеет типа $((\alpha \to \beta) \to \alpha) \to \alpha$.

6. Докажите следующие утверждения в ИИВ и постройте соответствующие лямбда-выражения согласно изоморфизму Карри-Ховарда:

   (a) $\vdash (\alpha \to \alpha) \to \alpha \to \alpha$
   (b) $\vdash \alpha \to \beta \to \beta$
   (c) $\vdash (\alpha \to \beta) \to \alpha \to \beta$
   (d) $\vdash (\alpha \to \alpha \to \beta) \to (\alpha \to \beta)$
   (e) $\vdash (\alpha \to \beta \to \gamma) \to (\beta \to \alpha \to \gamma)$
   (f) $\vdash (\alpha \to \beta) \to (\beta \to \gamma) \to (\alpha \to \gamma)$ (аналог контрапозиции)

7. Каков тип лямбда-выражения для суммы двух чёрчевских нумералов? Ответ поясните.

8. Заметим, что:

   $$\begin{array}{l} \vdash S : (\alpha \to \beta \to \gamma) \to (\alpha \to \beta) \to (\alpha \to \gamma) \\ \vdash K : \alpha \to \beta \to \alpha \end{array}$$

   Как несложно заметить, данные утверждения соответствуют (в смысле изоморфизма Карри-Ховарда) схемам аксиом для импликационного фрагмента интуиционистского исчисления высказываний в гильбертовском стиле. Значит, и доказательство утверждений может быть (согласно изоморфизму) перенесено в лямбда-исчисление.

   Гильбертовский стиль, который мы использовали в курсе матлогики, предполагал плоский список высказываний и номера утверждений для подсказок. Однако, мы можем изображать эти доказательства и в виде дерева:

   $$\dfrac{\alpha \to \alpha \to \alpha \qquad \dfrac{\alpha \to (\alpha \to \alpha) \to \alpha \quad (\alpha \to (\alpha \to \alpha) \to \alpha) \to (\alpha \to \alpha \to \alpha) \to (\alpha \to \alpha)}{(\alpha \to \alpha \to \alpha) \to (\alpha \to \alpha)}}{\alpha \to \alpha}$$

   Давайте теперь изобразим вывод типа (для экономии места мы не указываем вывод типов для комбинаторов S и K).

$$\dfrac{\vdash K : \alpha \to \alpha \to \alpha \qquad \dfrac{\vdash K : \alpha \to (\alpha \to \alpha) \to \alpha \quad \vdash S : (\alpha \to (\alpha \to \alpha) \to \alpha) \to (\alpha \to \alpha \to \alpha) \to (\alpha \to \alpha)}{\vdash S\ K : (\alpha \to \alpha \to \alpha) \to (\alpha \to \alpha)}}{\vdash S\ K\ K : \alpha \to \alpha}$$

Осталось заметить, что действительно $I =_\beta S\ K\ K$.

На основе изложенного, постройте доказательства следующих утверждений в гильбертовском стиле и выразите соответствующие выражения с помощью комбинаторов S и K:

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

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

(c) $(\alpha \to \alpha \to \beta) \to (\alpha \to \beta)$

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

9. В дополнение к базису SK рассмотрим базис $BCKW$:

$$\begin{aligned}
&B = \lambda x.\lambda y.\lambda z.x\ (y\ z)\\
&C = \lambda x.\lambda y.\lambda z.x\ z\ y\\
&K = \lambda x.\lambda y.x\\
&W = \lambda x.\lambda y.x\ y\ y
\end{aligned}$$

Выведите типы для данных комбинаторов, постройте схемы аксиом для соответствующего гильбертовского исчисления высказываний и покажите, что данное исчисление также позволяет доказать любое утверждение из импликационного фрагмента ИИВ.

Домашнее задание №4: «выразительная сила $\lambda_\to$; три задачи»

1. *(Теорема о замкнутости импликационного фрагмента интуиционистского исчисления высказываний)* Пусть формулы $\Gamma = \{\gamma_1, \gamma_2, \dots, \gamma_n\}$ и α взяты из импликационного фрагмента ИИВ. Покажем, что если $\Vdash_C \Gamma$ влечёт $\Vdash_C \alpha$ в любой модели Крипке C, то тогда $\Gamma \vdash_{\text{иф}} \alpha$.

Возьмём следующее множество миров: $W = \{\Delta \mid \Gamma \subseteq \Delta\}$. Пусть заданы два мира $w_1, w_2 \in W$. Договоримся, что $w_1 \preceq w_2$, если $w_1 \subseteq w_2$. Также договоримся, что если $w_i \in W$ и P — некоторая пропозициональная переменная, что $w_i \vdash P$, то $w_i \Vdash P$. Напомним, что данной тройки $\langle W, (\preceq), (\Vdash)\rangle$ достаточно для задания модели Крипке. Тогда рассмотрим следующие задачи:

(a) Покажите корректность определения модели Крипке: покажите, что если $w_i \preceq w_j$ и $w_i \Vdash P$, то $w_j \Vdash P$.

(b) Покажите, что $w_i \Vdash \varphi$ тогда и только тогда, когда $w_i \vdash \varphi$. *Указание:* Из всего определения моделей Крипке в импликационном фрагменте имеют смысл только определения для переменной и импликации. Поэтому главное содержательное утверждение — показать, что $w_i \Vdash \psi_1 \to \psi_2$ тогда и только тогда, когда $w_i \vdash \psi_1 \to \psi_2$. Используйте структурную индукцию и определение оценки импликации в моделях Крипке.

(c) Пусть $W = \{\Gamma\}$. Предъявите пример таких Γ и α, что $\Vdash \Gamma$, $\Vdash \alpha$, но $\Gamma \not\vdash \alpha$.

(d) К сожалению, подобным путём доказать полноту моделей Крипке для ИИВ со всеми связками невозможно. Пусть мы построили аналогичную конструкцию для полного ИИВ. Тогда предложите такие Γ и α, что при выполненном $\Vdash \Gamma$ выполнение $\Vdash \varphi$ не будет влечь $\Gamma \vdash \varphi$.

Теперь завершим доказательство: в самом деле, если $\Vdash_C \Gamma$ влечёт $\Vdash_C \alpha$ в любой модели Крипке C, то оно будет выполнено и в построенной выше модели $\langle W, (\preceq), (\Vdash)\rangle$. То есть, если $\Vdash \gamma_1$, ..., $\Vdash \gamma_n$, то $\Vdash \alpha$. Значит, по определению импликации в моделях Крипке имеем

$$\Vdash \gamma_1 \to \dots \to \gamma_n \to \alpha$$

Значит, по свойству (b):

$$\vdash \gamma_1 \to \dots \to \gamma_n \to \alpha$$

И по теореме о дедукции получаем искомое $\Gamma \vdash \alpha$.

2. Покажите, что функция возведения в степень не является расширенным полиномом.

3. Пусть тип $\nu = (\alpha \to \alpha) \to (\alpha \to \alpha)$, где α — это некоторый заранее зафиксированный атомарный тип. Предложите такие лямбда-выражения $F : \nu \to \nu \to \nu$, что:
   (a) Если $m, n \in \mathbb{N}_0$, то $F_a\ \overline{m}\ \overline{n} =_\beta \overline{m+n}$
   (b) Если $m, n \in \mathbb{N}_0$, то $F_b\ \overline{m}\ \overline{n} =_\beta \overline{m \cdot n}$

Домашнее задание №5: «унификация алгебраических термов»

1. Покажите, что несовместная система не имеет решений. А именно, решений нет, если:
   (a) Если в системе есть уравнение вида $x = \Theta(x)$, где $\Theta(x)$ — некоторый нетривиальный алгебраический терм со свободной переменной x.
   (b) Если в системе есть уравнение вида
   $$f_n\ \theta_1\ \theta_2\ \dots\ \theta_n = g_m\ \sigma_1\ \sigma_2\ \dots\ \sigma_m$$
   где $f_n \not\equiv g_m$.

2. Покажите, что следующие операции строят эквивалентную систему уравнений:
   (a) Исключение переменной: из системы
   $$\begin{cases} x = \xi \\ \sigma_1 = \theta_1 \\ \dots \\ \sigma_n = \theta_n \end{cases}$$
   операция строит систему
   $$\begin{cases} x = \xi \\ \sigma_1[x := \xi] = \theta_1[x := \xi] \\ \dots \\ \sigma_n[x := \xi] = \theta_n[x := \xi] \end{cases}$$
   (b) Редукция терма: из системы
   $$\begin{cases} f_l\ \zeta_1\ \dots\ \zeta_l = f_l\ \eta_1\ \dots\ \eta_l \\ \sigma_1 = \theta_1 \\ \dots \\ \sigma_n = \theta_n \end{cases}$$
   операция строит систему
   $$\begin{cases} \zeta_1 = \eta_1 \\ \dots \\ \zeta_l = \eta_l \\ \sigma_1 = \theta_1 \\ \dots \\ \sigma_n = \theta_n \end{cases}$$

3. В доказательстве завершаемости алгоритма унификации использовалась лексикографически упорядоченная монотонно убывающая последовательность троек чисел. Однако, точного доказательства конечности этой последовательности не было дано. Покажите, что:
   (a) Не существует бесконечной строго убывающей последовательности упорядоченных троек: таких $\langle x_i, y_i, z_i \rangle$, $x_i, y_i, z_i \in \mathbb{N}_0$, что $\langle x_n, y_n, z_n \rangle > \langle x_{n+1}, y_{n+1}, z_{n+1} \rangle$ при любом n.
   (b) Покажите, что любая строго убывающая последовательность ординалов имеет конечную длину.
   (c) Поясните, почему первый пункт данной задачи является частным случаем второго.

4. При помощи рассказанного на лекции алгоритма найдите типы для следующих лямбда-выражений, или покажите, что у них нет типа:
   (a) $\lambda x.\lambda y.x$
   (b) $\lambda x.\lambda y.\lambda z.x\ z\ (y\ z)$
   (c) $(\lambda x.x\ x)\ (\lambda x.x\ x)$
   (d) $\lambda f.\lambda x.f\ (f\ x)$
   (e) $\lambda m.\lambda n.\lambda f.\lambda x.m\ f\ (n\ f\ x)$

(f) $\lambda m.\lambda n.n\ m$
(g) $(\lambda m.\lambda n.n\ m)\ \overline{3}\ \overline{3}$
(h) $(\lambda s.(\lambda m.\lambda n.n\ m)\ s\ s)\ \overline{3}$
(i) In_L, In_R, $Case$
(j) Pr_L, Pr_R, $MkPair$
(k) Лямбда-выражение, возвращающее T, если чёрчевский нумерал равен нулю — иначе F.
(l) Лямбда-выражение, проверяющее чётность чёрчевского нумерала.

5. Покажите, что если алгоритм нашёл тип для выражения, то можно построить доказательство, выводящее этот тип в просто типизированном лямбда-исчислении.

6. Покажите, что если для некоторое лямбда-выражение имеет тип, то у уравнения в алгебраических термах, строящегося в алгоримте по выражению, найдётся решение.

7. Назовём *наиболее общей парой* для лямбда-выражения M такую пару $\langle\Gamma,\sigma\rangle$ ($\Gamma\vdash M:\sigma$), что любая другая пара $\langle\Delta,\tau\rangle$ ($\Delta\vdash M:\tau$) является её частным случаем: существует подстановка S, что $\Delta = S(\Gamma)$ и $\sigma = S(\tau)$.

   (a) дайте корректное определение подстановкам на типах и контекстах ($S(\Gamma)$ и $S(\tau)$), «рукомашески» использованным выше;
   (b) покажите, что алгоритм типизации находит наиболее общую пару.

Домашнее задание №6: «движемся вперёд: полное исчисление высказываний, логика второго порядка»

1. Найдите термы, населяющие указанные ниже типы, постройте доказательство (вывод соответствующего типа), поясните смысл соответствующих следующим лямбда-выражениям программ:

   (a) $\alpha\to\neg\neg\alpha$
   (b) $(\alpha\to\beta)\to(\neg\beta\to\neg\alpha)$
   (c) *Даёшь теорему Гливенко!* $\neg\neg(\alpha\vee\neg\alpha)$
   (d) $\neg\neg(((\alpha\to\beta)\to\alpha)\to\alpha)$
   (e) *Как вы догадываетесь, обитаем только один из двух вариантов законов Де Моргана, укажите этот вариант и решите задачу для него:* $\alpha\vee\beta\to\neg(\neg\alpha\ \&\ \neg\beta)$ или $\neg(\neg\alpha\ \&\ \neg\beta)\to\alpha\vee\beta$.

2. Постройте доказательства в импликационном фрагменте исчисления второго порядка для следующих аксиом полного исчисления:

   (a) введение конъюнкции;
   (b) исключение конъюнкции;
   (c) введение дизъюнкции;
   (d) исключение дизъюнкции;
   (e) исключение лжи;
   (f) введение квантора существования;
   (g) исключение квантора существования;
   (h) введение отрицания;
   (i) исключение отрицания.

3. Существует два различных варианта аксиом для конъюнкции в исчислении высказываний. Один был на лекции, второй приведён ниже:

$$\frac{\Gamma\vdash P\qquad\Gamma\vdash Q}{\Gamma\vdash P\ \&\ Q}\qquad\frac{\Gamma\vdash P\ \&\ Q\qquad\Gamma,P,Q\vdash R}{\Gamma\vdash R}$$

   Покажите, что аксиомы конъюнкции в каждом из вариантов исчисления могут быть доказаны как теоремы в другом варианте.

4. Предложите лямбда-выражения в системе F и выведите типы для следующих конструкций нетипизированного лямбда-исчисления:
   (a) T, F, исключающее или;
   (b) чёрчевский нумерал, сложение;
   (c) возведение в степень чёрчевских нумералов;
   (d) разность чёрчевских нумералов.
5. Докажите дистрибутивность в логике второго порядка, покажите обитаемость типа и поясните смысл получившейся программы:
$$\forall\alpha.\forall\beta.\forall\gamma.\alpha \vee (\beta \,\&\, \gamma) \to (\alpha \vee \beta) \,\&\, (\alpha \vee \gamma)$$
6. Выразимы ли Ω или Y комбинаторы в системе F?
7. Сформулируйте теорему Чёрча-Россера для исчисления второго порядка. Предложите схему её доказательства.

Домашнее задание №7: «экзистенциальные типы»

1. На лекции были выписаны следующие лямбда-выражения для конструкций с кванторами:
$$\begin{array}{ll} (a) & \textbf{pack}\ M, \tau\ \textbf{to}\ \exists\alpha.\sigma = \Lambda\beta.\lambda x^{\forall\alpha.\sigma\to\beta}.x\ \tau\ M \\ (b) & \textbf{abstype}\ \alpha\ \textbf{with}\ x:\sigma\ \textbf{is}\ M\ \textbf{in}\ N^{\rho} = M\ \rho\ (\lambda\alpha.\lambda x^{\sigma}.N) \end{array}$$
   Возможно, в конструкциях есть ошибки — исправьте их и докажите, что данные конструкции удовлетворяют аксиомам, если выразить $\exists\alpha.P$ как $\forall\phi.(\forall\alpha.P \to \phi) \to \phi$.
2. Переформулируйте систему F в исчислении по Карри: укажите новые схемы аксиом для квантора всеобщности.
3. Переформулируйте операции abstype и pack для исчисления по Карри, укажите соответствующие им лямбда-выражения и покажите, что эти выражения соответствуют аксиомам.

Домашнее задание №8: «типовая система Хиндли-Милнера»

1. Определим отрицание двумя способами: $\neg\phi = \phi \to \forall p.p$ и $\sim\phi = \forall p.\phi \to p$.
   (a) Покажите, что оба отрицания эквивалентны в логике 2 порядка, то есть, что выполены следующие правила:
$$\frac{\Gamma \vdash \sim\phi}{\Gamma \vdash \neg\phi} \qquad \frac{\Gamma \vdash \neg\phi}{\Gamma \vdash \sim\phi}$$
   (b) Покажите, что если $\Gamma \vdash M : \neg\tau$, то найдётся такое выражение M', что $\Gamma \vdash M' : \sim\tau$; и наоборот, если $\Gamma \vdash N : \sim\tau$, то для какого-то N' выполнено $\Gamma \vdash N' : \neg\tau$
2. Обозначим минимальный ранг типа τ за $rk(\tau)$. Приведите пример типа τ, что $rk(\tau) = 3$.
3. Докажите, что $rk(\exists x.\phi) > 1$.
4. Придумайте семейство типов τ_n, такое, что $rk(\tau_n) = n$.
5. Определим *арифметическую иерархию* на *классическом* исчислении предикатов *второго* порядка (сразу упомянем, что данное определение не классическое — традиционно его вводят для исчисления предикатов первого порядка):
   - $\Pi_0 = \Sigma_0$ — все выражения, логически эквивалентные бескванторным выражениям;
   - $\Pi_n, n > 0$ — все выражения, логически эквивалентные выражениям вида $\forall x.S$, где $S \in \Sigma_{n-1}$;
   - $\Sigma_n, n > 0$ — все выражения, логически эквивалентные выражениям вида $\exists x.P$, где $P \in \Pi_{n-1}$.

Заметим, что, например, Π_3 состоит из выражений вида $\forall x.\exists y.\forall z.R$. Также заметим, что, поскольку к любой формуле можно приписать любые кванторы по свежим переменным и получить формулу, логически эквивалентную исходной, то если $m < n$, то $\Pi_m \subset \Pi_n$ и $\Pi_m \subset \Sigma_n$.

На лекции была высказана гипотеза, что ранг типов для лямбда-выражений в системе F связан с чередованием кванторов. В данной задаче мы предлагаем вам разобраться в этом вопросе:

(a) Существует ли такая константа k и такое семейство типов τ_n, что $rk(\tau_n) = n$, но $\tau_n \in \Pi_k$

(b) Существует ли такая константа k и такое семейство типов τ_n, что $\tau_n \in \Pi_n \setminus \Pi_{n-1}$, но $rk(\tau_n) \le k$.

6. Пусть даны две типовых схемы σ_1 и σ_2. Придумайте алгоритм проверки того, что $\sigma_1 \sqsubseteq \sigma_2$.

7. Напомним, что, по определению,

$$(\textbf{let } x = E_0 \textbf{ in } E_1) =_\beta (\lambda x.E_1)\ E_0$$

На основании этой эквивалентности мы можем для каждого лямбда-выражения E' в системе HM сопоставить ему β-эквивалентное выражение E в системе F.

Пользуясь этой эквивалентностью, найдите выражения в системе HM для следующих конструкций в системе F, и докажите, что их типы могут быть выведены в системе HM:

(a) Булевские значения, функция XOR.

(b) Чёрчевские нумералы, функция «сумма».

(c) Возведение в степень (для чёрчевских нумералов).

(d) Вычитание.

(e) Деление.

Домашнее задание №9: «алгоритм W; типовая система HM»

1. С помощью алгоритма W выведите типы для следующих выражений (или укажите, что выражения не имеют типа):

(a) a

(b) $x : \alpha \vdash x\ x$

(c) $\vdash \lambda x.x\ x$

(d) $\vdash \textbf{let } \overline{1} = \lambda f.\lambda x.f\ x \textbf{ in } \overline{1}\ \overline{1}$

(e) $\vdash \textbf{let } s = \lambda f.\lambda x.f\ (f\ x) \textbf{ in } sq\ (sq\ \overline{1})$

(f) $\vdash \textbf{let } s = \lambda f.\lambda x.f\ (f\ x) \textbf{ in } (sq\ sq)\ \overline{1}$

(g) $\vdash \textbf{let } s = \lambda f.\lambda x.f\ (f\ x) \textbf{ in } sq\ sq\ sq\ sq\ sq\ \overline{1}$; чему равен результат бета-редукции указанного терма?

2. Рассмотрим типовую систему HM+Y: система Хиндли-Милнера в которой, по определению, $\vdash (\lambda f.(\lambda x.f\ (x\ x))\ (\lambda x.$ $\forall \alpha.(\alpha \to \alpha) \to \alpha$.

(a) Типизируем ли в этой типовой системе $\Omega = \omega\ \omega$, где $\omega = \lambda x.x\ x$?

(b) Найдите такой терм F, что $\vdash F : \perp$.

(c) Найдите такой терм E, что $\vdash E : \alpha \vee \neg\alpha$.

3. Как нетрудно заметить, 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»:

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

Если бы такое можно было выразить в типовой системе Хиндли-Милнера, то операция добавления элемента к списку записалась бы на языке Окамль вот так (сравните с прибавлением 1 к числу в двоичной системе счисления):

```
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` (рекурсивная функция должна уже включать в себя Y-комбинатор и не требовать никаких дополнительных усилий для вызова)? Считайте, что тип `bin_list 'a` уже как-то задан, и обозначается как $\tau(\alpha)$.

(b) Какой ранг имеет тип этой функции, почему её не скомпилировать в Окамле?

(c) Предложите функцию для удаления элемента списка (головы).

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

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

4. Задан тип «дерево»:

```
type 'a tree = Leaf of 'a | Node of (tree 'a) * (tree 'a);;
```

Задайте тип $\tau(\alpha)$:

(a) Как эквирекурсивный тип (задайте через μ-оператор).

(b) Как изорекурсивный тип (определите функции `roll` и `unroll`, укажите их тип).

Домашнее задание №10: «Язык Идрис»

1. Воспользовавшись функцией `sprintf` как образцом, добавьте следующие шаблоны:

   (a) строка (шаблон `%s`);

   (b) десятичное число заданной длины с ведущими нулями (шаблон `%05d` и подобные);

   (c) строка заданной параметрами длины (шаблон `%*.*s`).

2. Определите в языке Идрис конъюнкцию и дизъюнкцию с помощью квантора всеобщности и импликации (аналогично интуиционистскому исчислению предикатов второго порядка). Определите все стандартные операции для них (инъекции, проекции и т.п.): эти операции, очевидно, будут доказательством некоторых утверждений в интуиционистской логике. Какие это утверждения, приведите их.

3. Аналогично предыдущему заданию, определите в языке Идрис чёрчевские нумералы и арифметические операции на них.

4. Рассмотрим три алгебраических типа из языка Идрис:

   (a) натуральное число `Nat`:
   `https://github.com/idris-lang/Idris-dev/blob/master/libs/prelude/Prelude/Nat.idr`;

   (b) ограниченное целое число `Fin`:
   `https://github.com/idris-lang/Idris-dev/blob/master/libs/base/Data/Fin.idr`;

   (c) ограниченный вектор `Vect`:
   `https://www.idris-lang.org/docs/current/base_doc/docs/Data.Vect.html` и

   Определите функцию `swap: Vect n a -> (Fin n) -> (Fin n) -> Vect n a`, строящую новый вектор, в котором два элемента вектора поменяны местами.

5. Определите функции арифметики для Fin:

   (a) `plus_fin: Fin a -> Fib b -> Fin (a+b)`

   (b) `mul_fin: Fin a -> Fin b -> Fin (a*b)`

   (c) `dec_fin: Fin (S a) -> Fin a`

6. Определите функции минимума для натуральных (Пеано) и конечных чисел:
   `nat_min: Nat -> Nat -> Nat`
   `min_fin: {a:Nat} -> {b:Nat} -> Fin (S a) -> Fin (S b) -> Fin (S (nat_min a b))`

   Также определите функции:

```
map2: {X:Type} -> {Y:Type} -> {Z:Type} -> (a:Nat) -> (b:Nat) ->
                        (X->Y->Z) -> Vect a X -> Vect b Y -> Vect (nat_min a b) Z

index2: {X:Type} -> {Y:Type} -> (a:Nat) -> (b:Nat) -> Fin (nat_min a b) ->
                        Vect a X -> Vect b Y -> (X,Y)
```

Домашнее задание №11: «Идрис, простые доказательства»

Ещё раз напомним конструкцию `replace`, использованную на лекции:
`replace: (x=y) -> P x -> P y`.

Функция replace берёт два явных параметра, и один неявный (P). P — это некоторый тип, зависящий от x. Функция имеет естественный смысл: если два значения равны, и свойство выполнено для одного из них, то оно выполнено и для другого.

Неявность P предполагает, что компилятор может догадаться до того, что это за значение, но на практике он обычно не справляется с этой задачей. Поэтому обычно P нужно указывать.

Давайте поймём, что это должен быть за P, и для этого рассмотрим пример:
`plus_zero_commutative: (a:Nat) -> 0 + a = a + 0`

Здесь мы, имея предположением индукции `0+a=a+0`, должны доказать `0 + (S a) = (S a) + 0`. Логично взять предположение за равенство $x = y$, при этом `x` будет соответствовать `0 + a`, а `y` будет соответствовать `a + 0`. Теперь подберём такое P, чтобы `P (0 + a)` унифицировалось с `0 + (S a)`, а `P (a + 0)` унифицировалось с `(S a) + 0`.

Давайте возьмём `P = \w => S (0+a)=S w`. Тогда `P (0 + a)` — это `S (0+a) = S (0+a)` (что очевидно доказывается при помощи `Refl`), а `P (a+0)` — это `S (0+a) = S (a+0)` (что является требуемым утверждением, так как `S (0+a)` = `S(a)` = `0+S(a)`; компилятор, как мы обсуждали, способен короткие цепочки подобных преобразований производить самостоятельно).

Итак, мы получили следующий код:

```
plus_zero_commutative: (a:Nat) -> 0 + a = a + 0
plus_zero_commutative Z = Refl
plus_zero_commutative (S a) =
      replace {P = \w => S (0+a)=S w} (plus_zero_commutative a) Refl
```

В отличие от `replace`, конструкция `rewrite` имеет дополнительный эвристический алгоритм для подбора соответствующего P, поэтому в части случаев мы можем довериться её интеллекту.

1. Свойства равенства. Докажите, что:
   (a) `x = y -> y = x`
   (b) `x = y -> y = z -> x = z`
   (c) (конгруэнтность) `(P: A->B) -> x = y -> P x = P y`
2. Простая арифметика — сложение. Докажите, что:
   (a) `x = x + 0`
   (b) `S x = 1 + x`
   (c) `S x = x + 1`
   (d) `S x + S x = S (S (x + x))`
   (e) `S x + S y = S (S (x + y))`
   (f) `S (x + y) = x + (S y)`
   (g) `x + y = y + x`
   (h) `x + (y + z) = (x + y) + z`
3. Простая арифметика — умножение. Докажите, что:
   (a) `0 = x * 0`

(b) `0 = 0 * x`

(c) `x = 1 * x`

(d) `x = x * 1`

(e) `x * y = y * x`

(f) `x * (y * z) = (x * y) * z`

(g) `x * (y + z) = x * y + x * z`

(h) `(y + z) * t = y * t + z * t`

(i) `(x + y) * 2 = x + x + y + y`

4. Отношение «меньше или равно» определено в библиотеке Идрис так:

```
data LTE : (n : Nat) -> (m : Nat) -> Type where
    LTEZero : LTE 0 right  -- Zero is the smallest Nat
    LTESucc : LTE left right -> LTE (S left) (S right)
                           -- If n <= m, then n + 1 <= m + 1
```

(a) Докажите, что `LTE 3 5`

(b) Докажите, что `LTE x y -> LTE x (S y)`

(c) Докажите, что `LTE x y -> LTE (x+n) (y+n)`

(d) `LTE x y -> LTE y z -> LTE x z`

(e) `LTE x y -> LTE y x -> x = y`

(f) `LTE x x`

5. Определите отношение «строго больше», GT. Докажите, что

(a) `GT 5 3`

(b) `GT x y -> GT y z -> GT x z`

(c) `GT (x+1) x`

(d) `GT x y -> GT (x+n) (y+n)`

(e) `GT (x*x) x`

(f) `GT x 2 -> GT (x*x) (x+x)`

(g) `Either (LTE x y) (GT x y)`

6. Определите ограниченное вычитание `sub` (`sub x y` равно 0, если $x < y$), докажите:

(a) `LTE x y -> sub x y = 0`

(b) `sub x y = 0 -> LTE x y`

(c) `LTE y x -> y + (sub x y) = x`