Теоретические («малые») домашние задания

Теория типов, ИТМО, M3235-M3239, весна 2020 года

Домашнее задание №1: «вводная лекция для ТТ и ФП»

1. 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) $And\ F\ T$
   (d) $And\ T\ T$

2. Постройте лямбда-выражения для следующих булевских выражений:

   (a) Дизъюнкция
   (b) Штрих Шеффера («и-не»)
   (c) Исключающее или

3. 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
$IsZero$	$\lambda n.n\ (\lambda x.F)\ T$	проверка на 0

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

   (a) Сложение
   (b) Умножение на 2 ($Mul2$)
   (c) Умножение
   (d) Возведение в степень
   (e) Проверка на чётность
   (f) Деление на 3 (могут потребоваться пары и/или вычитания)
   (g) Сравнение двух чисел ($IsLess$) — истина, если первый аргумент меньше второго (могут потребоваться пары и/или вычитания)

4. Проредуцируйте выражение и найдите его нормальную форму:

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

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
$Case$	$\lambda l.\lambda r.\lambda c.c\ l\ r$	case для алгебраического типа
InL	$\lambda l.(\lambda x.\lambda y.x\ l)$	левая инъекция
InR	$\lambda r.(\lambda x.\lambda y.y\ r)$	правая инъекция

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

(b) Убедитесь, что $Case\ (\lambda x.T)\ (\lambda y.y)\ (InR\ p) \twoheadrightarrow_\beta p$.

(c) Постройте операцию вычитания 1 из числа

(d) Постройте операцию вычитания чисел

(e) Постройте опреацию деления чисел

6. Напомним определение Y-комбинатора: $\lambda f.(\lambda x.f\ (x\ x))\ (\lambda x.f\ (x\ x))$.

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

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

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

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

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

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

7. Пусть $\eta = (\alpha \rightarrow \alpha) \rightarrow (\alpha \rightarrow \alpha)$. Покажите (т.е. постройте соответствующее доказательство в исчислении по Карри), что:

   (a) $\vdash \overline{2} : \eta$.

   (b) $\vdash (+1) : \eta \rightarrow \eta$.

   (c) $\vdash Plus : \eta \rightarrow \eta$.

   (d) $\vdash Mul : \eta \rightarrow \eta$ (не каждая реализация умножения будет удовлетворять этому свойству; вам требуется найти нужную)

8. 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 вычитание 1 и вычитание? Что получается, а что — нет?

9. Типы для конъюнкции и дизъюнкции на Хаскеле. Списки.

   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. Тогда, используя данную идею, реализуйте в Хаскеле:

   (a) определите конструкции mkpair, prl, prr на Хаскеле — какой тип у данных конструкций? Сравните его с типом конъюнкции с лекции.

   (b) определите конструкции case, inl, inr — какой тип у данных конструкций? Сравните его с типом дизъюнкции с лекции.

   (c) постройте список целых чисел из данных конструкий.

   (d) определите функцию вычисления длины списка целых чисел с помощью данных конструкций (к сожалению, скомпилировать это выражение на Хаскеле не получится — поэтому достаточно написать исходный код).

Домашнее задание №2: «формализация лямбда-исчисления»

1. На лекции было использовано понятие свободы для подстановки.

   (a) Найдите лямбда-выражение, которое при однократной редукции требует переименования связанных переменных (редукция невозможна без переименования).

   (b) Заметим, что даже если мы запретим использовать одни и те же переменные в разных лямбда-абстракциях, это не будет решением проблемы переименований. Предложите лямбда-выражение, в котором (а) все лямбда-абстракции указаны по разным переменным; но (б) через некоторое количество редукций потребуется переименование связанных переменных.

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

   Также напомним определение:

$$\begin{aligned}&S := \lambda x.\lambda y.\lambda z.x\ z\ (y\ z)\\&K := \lambda x.\lambda y.x\\&I := \lambda x.x\end{aligned}$$

   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) $\overline{n}$

3. Бесконечное количество комбинаторов неподвижной точки. Дадим следующие определения

$$\begin{aligned}&L := \lambda abcdefghijklmnopqstuvwxyzr.r(thisisafixedpointcombinator)\\&R := LLLLLLLLLLLLLLLLLLLLLLLLL\end{aligned}$$

   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 33 параметрами и осмысленной русской фразой в терме L; покажите, что оно является комбинатором неподвижной точки.

4. Пусть задано $n \in \mathbb{N}$. Постройте лямбда-выражение, которое преобразуется в нормальную форму в n раз медленнее с помощью нормального порядка редукции, чем с помощью какого-то другого (самого быстрого) порядка редукции.

5. 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.

   (d) Предложите реализацию функции сложения в данном представлении.

   (e) Предложите реализацию функции вычитания в данном представлении.

   (f) Какова вычислительная сложность арифметопераций с двоичными нумералами?

6. Предложим альтернативные аксиомы для конъюнкции:

$$\frac{\Gamma \vdash \alpha \quad \Gamma \vdash \beta}{\Gamma \vdash \alpha \,\&\, \beta}\ \text{Введ. \&} \qquad \frac{\Gamma \vdash \alpha \,\&\, \beta \quad \Gamma, \alpha, \beta \vdash \gamma}{\Gamma \vdash \gamma}\ \text{Удал. \&}$$

(a) Предложите лямбда-выражения, соответствующие данным аксиомам; поясните, как данные выражения абстрагируют понятие «упорядоченной пары».

(b) Выразите изложенные в лекции аксиомы конъюнкции через приведённые в условии.

(c) Выразите приведённые в условии аксиомы конъюнкции через изложенные в лекции.

7. Как мы уже разбирали, $\nvdash x\ x : \tau$ в силу дополнительных ограничений аксиомы

$$\frac{}{\Gamma, x : \tau \vdash x : \tau}\ x \notin FV(\Gamma)$$

Найдите лямбда-выражение N, что $\nvdash N : \tau$ в силу ограничения аксиомы

$$\frac{\Gamma, x : \sigma \vdash N : \tau}{\Gamma \vdash \lambda x.N : \sigma \to \tau}\ x \notin FV(\Gamma)$$

Домашнее задание №3: «вывод типов; алгоритм унификации»

1. *Вполне упорядоченным* множеством назовём такое линейно-упорядоченное отношением ($\prec$) множество S (и такой порядок назовём *полным*), что какое бы ни было множество $U \subseteq S$, в U найдётся наименьший элемент.

   (a) Покажите, что неотрицательные вещественные числа $[0, +\inf)$ — не вполне упорядоченное множество. Существуют ли конечные и счётные не вполне упорядоченные множества?

   (b) Определим лексикографический порядок на $\mathbb{N}^n$: положим, что $\langle a_1, a_2, \dots a_n\rangle \prec \langle b_1, b_2, \dots b_n$, если найдётся такой k, что $a_1 = b_1$, ..., $a_{k-1} = b_{k-1}$, но $a_k < b_k$. Покажите, что такой порядок — полный.

   (c) Пусть S вполне упорядочено отношением ($\prec$), определим $a \succ b := b \prec a$. Пусть $a_1 \succ a_2 \succ a_3 \succ \dots$ — строго монотонно убывающая последовательность значений из S. Покажите, что данная последовательность всегда имеет конечную длину.

2. Рассмотрим полное интуиционистское исчисление высказываний. Дополните алгоритм вывода типов дополнительными функциональными символами для связок &, $\vee$ и $\perp$ (а также сделайте дополнительные необходимые исправления в нём) и продемонстрируйте вывод типов для выражения, использующего хотя бы две из данных трёх конструкций.

3. Поясним название «алгебраические типы» — это семейство составных типов, позволяющих строить «алгебраические» выражения на типах:

название	обозначение	алгебраический смысл
тип-сумма, «алгебраический»	$\alpha \vee \beta$	$\alpha + \beta$
тип-произведение, пара	$\alpha \,\&\, \beta$	$\alpha \times \beta$
тип-степень, функция	$\alpha \to \beta$	β^α

   Название «алгебраический» закрепилось в первую очередь за типом-суммой (видимо потому, что остальные типы имеют устоявшиеся названия), однако, может быть отнесено и к другим типам.

   Поясните «типовый» (программистский) смысл следующих алгебраических тождеств — и постройте программы, их доказывающие:

   (a) $\gamma \times (\alpha + \beta) = \gamma \times \alpha + \gamma \times \beta$.

   (b) $\gamma^{\alpha \times \beta} = (\gamma^\alpha)^\beta$. Как называется данное тождество?

   (c) $\gamma^{\alpha + \beta} = \gamma^\alpha \times \gamma^\beta$.

4. Найдите лямбда-выражения, доказывающие:

   (a) Формулу де-Моргана $\neg(\alpha \vee \beta) \to \neg\alpha \,\&\, \neg\beta$.

   (b) Контрапозицию $(\alpha \to \beta) \to (\neg\beta \to \neg\alpha)$.

   (c) Закон исключённого третьего после применения теоремы Гливенко $\neg\neg(\alpha \vee \neg\alpha)$.

Домашнее задание №4: «логика второго порядка; система F»

1. Докажите в исчислени предикатов второго порядка теоремы–аналоги правил вывода для следующих связок:

   (a) конъюнкция;

   (b) дизъюнкция;

   (c) отрицание и ложь;

   (d) квантор существования.

   Напомним: в задании требуется по заданным аксиоме и аргументам связки найти такое дерево применений правил И.П. 2 порядка, которым можно было бы заменить применение исходной аксиомы.

2. Напомним, что в системе F естественно предложить выражение $\Lambda\alpha.\lambda f^{\alpha\to\alpha}.\lambda x^{\alpha}.f^{n}\ x$ как аналог для чёрчевского нумерала. Постройте в F выражения для следующих функций и выведите их тип:

   (a) сложение;

   (b) умножение;

   (c) возведение в степень

   (d) вычитание 1

3. Перенесите в систему F выражения для булевских значений и операции And и выведите их тип.

4. Выразите в языке Хаскель конструкции системы F через импликацию и квантор всеобщности.

   Точнее: запишите выражение, напишите программы, доказывающие соответствующие аксиомы, укажите, как преобразовать данное выражение в нативный хаскелевский тип и наоборот. При реализации вам потребуется использовать квантор всеобщности в типах (`type T = forall x. ...`) и включить опцию RankNTypes.

   Список конструкций:

   (a) конъюнкция;

   (b) дизъюнкция;

   (c) отрицание и ложь (соответствует `undefined`).

5. *Экзистенциальные типы.* Укажем правила вывода для квантора существования в системе F:

$$\frac{\Gamma \vdash M : \sigma[\alpha := \tau]}{\Gamma \vdash \textbf{pack}\ \tau, M\ \textbf{to}\ \exists\alpha.\sigma : \exists\alpha.\sigma}\ \text{Введ. }\exists$$

$$\frac{\Gamma \vdash M : \exists\alpha.\sigma \quad \Gamma, x : \sigma \vdash N : \rho}{\Gamma \vdash \textbf{abstype}\ \alpha\ \textbf{with}\ x : \sigma\ \textbf{is}\ M\ \textbf{in}\ N : \rho}\ \text{Удал. }\exists; \alpha \notin FV(\Gamma, \rho)$$

   Постараемся прояснить смысл данных конструкций. В объектно-ориентированных языках программирования обычно хранение данных объединено с библиотечной структурой: структура данных всегда неявно присутствует как параметр методов класса. Аналогия с квантором существования подсказывает, что эта связь необязательна: библиотека характеризуется сигнатурой её методов (σ) и типом, хранящим значения (α).

   Если мы хотим задать некоторый абстрактный тип данных, мы используем pack. Например, давайте построим АТД «список» в некотором обобщении лямбда-исчисления:

$$\begin{array}{l} let\ intstack = \textbf{pack}\ list, \langle Nil, \langle (\lambda l^{list}.\lambda x^{int}.Cons\ x\ l), (\lambda l^{list}.\langle Head\ l, Tail\ l\rangle)\rangle\rangle \\ \textbf{to}\ \exists\alpha.\alpha \,\&\, ((\alpha \to int \to \alpha) \,\&\, (\alpha \to int \,\&\, \alpha)) \end{array}$$

   Если же мы желаем использовать такой АТД, мы используем abstype; вот такой код вернёт число 12:

$$\begin{array}{l} \textbf{abstype}\ \alpha\ \textbf{with}\ x : \alpha \,\&\, ((\alpha \to int \to \alpha) \,\&\, (\alpha \to int \,\&\, \alpha))\ \textbf{is}\ intstack\ \textbf{in} \\ let\ st^{\alpha} = Pr_L\ x\ in \\ let\ st^{\alpha} = (Pr_L\ (Pr_R\ x))\ st\ 12\ in \\ Pr_L\ ((Pr_R\ (Pr_R\ x))\ st) \end{array}$$

   В завершение определим правило бета-редукции для данных конструкций.

$$\textbf{abstype}\ \alpha\ \textbf{with}\ x : \sigma\ \textbf{is}\ \textbf{pack}\ M, \tau\ \textbf{to}\ \exists\alpha.\sigma\ \textbf{in}\ N \to_{\beta} N[\alpha := \tau][x := M]$$

(a) Предъявите лямбда-выражение, соответствующее pack в системе F (без использования сокращений записи — в частности, без $(\exists)$). Покажите, что оно действительно имеет приписываемый ему тип.

(b) Предъявите лямбда-выражение, соответствующее abstype в системе F (без $(\exists)$). Покажите, что оно действительно имеет приписываемый ему тип.

(c) Рассмотрим вариант системы F, типизированный по Карри (отсутстуют типовые абстракции и применения, а также указания типов в аргументах). Предложите реализацию pack и abstype в этом варианте исчисления.

(d) Предложите реализацию на Хаскеле и пример использования для массива фиксированного размера (размер указывается при инициализации), соответствующего интерфейсу, заданному следующим экзистенциальным типом (для компиляции требуется включить опцию RankNTypes):

```
data AbstractArray x = AA (forall b . (forall a .
     (Integer -> a, a -> (Integer, x) -> a, a -> Integer -> (a, x)) -> b) -> b)
```

(e) Предположим, что в Джаве мы не используем наследования, а только разрешаем реализовывать интерфейсы. Предложите формализацию для классов и интерфейсов с использованием экзистенциальных типов. Как реализовать публичные и приватные поля?