Введение в Теорию Типов
Конспект лекций

Штукенберг Д. Г.
Университет ИТМО

28 января 2019 г.

1 Введение

Эти лекции были рассказаны студентам групп M3334–M3337, M3339 в 2018 году в Университете ИТМО, на Кафедре компьютерных технологий Факультета информационных технологий и программирования.

Конспект подготовили студенты Кафедры: Егор Галкин (лекции 1 и 2), Илья Кокорин (лекции 3 и 4), Никита Дугинец (лекции 5 и 6), Степан Прудников (лекции 7 и 8).

(возможно, история сложнее)

2 Лекция 1

2.1 λ-исчисление

Определение 2.1 (λ-выражение)**.** λ-выражение — выражение, удовлетворяющее грамматике:

$$\Phi ::= x \,|\, (\Phi) \,|\, \lambda x.\Phi \,|\, \Phi\ \Phi$$

Иногда для упращения записи мы будем опускать скобки. В этом случае, перед разбором выражения, следует расставить все опущенные скобки. При их рассатвлении будем придерживаться правил:

1. В аппликации расставляем скобки слева направо: $A\ B\ C \implies (A\ B)\ C$.

2. Абстракции жадные — поглащают скобками все что могут до конца строки: $\lambda a.\lambda b.a\ b \implies \lambda a.(\lambda b.(a\ b))$.

   Пример. $\lambda x.(\lambda f.((fx)(fx)\lambda y.(yf)))$

Договоримся, что:

- Переменные — x, a, b, c.
- Термы (части λ-выражения) — X, A, B, C.
- Фиксированные переменные обозначаются буквами из начала алфавита, метапеременные — из конца.

Есть понятия связанного и свободного вхождения переменной (аналогично исчислению предикатов).

Определение 2.2. Если вхождение x находится в области действия абстракции по x, то такое вхождение называется связанным, иначе вхождение называется свободным.

Определение 2.3. Терм Q называется свободным для подстановски в Φ вместо x, если после подстановки Q ни одно вхождение не станет связанным.

Пример. $\lambda x.A$ связывает все свободные вхождения x в A.

Определение 2.4. Функция $V(A)$ — множество переменных, входящих в A.

Определение 2.5. Функция $\mathrm{FV}(A)$ — множество свободных переменных, входящих в A:

$$\mathrm{FV}(A) = \begin{cases} \{x\} & \text{если } A \equiv x \\ \mathrm{FV}(P) \cup \mathrm{FV}(Q) & \text{если } A \equiv PQ \\ \mathrm{FV}(P) \setminus \{x\} & \text{если } A \equiv \lambda x.P \end{cases}$$

λ-выражение можно понимать как функцию. Абстракция — это функция с аргументом, аппликация — это передача аргумента.

Определение 2.6 (α-эквивалентность)**.** $A =_\alpha B$, если имеет место одно из следующих условий:

1. $A \equiv x$, $B \equiv y$ и $x \equiv y$.
2. $A \equiv P_1Q_1$, $B \equiv P_2Q_2$ и $P_1 =_\alpha P_2$, $Q_1 =_\alpha Q_2$.
3. $A \equiv \lambda x.P_1$, $B \equiv \lambda y.P_2$ и $P_1[x := t] =_\alpha P_2[y := t]$, где t — новая переменная.

Пример. $\lambda x.\lambda y.xy =_\alpha \lambda y.\lambda x.yx$.

Доказательство.

1. $tz =_\alpha tz$ верно по второму условию.
2. Тогда получаем, что $\lambda y.ty =_\alpha \lambda x.tx$ по третьему условию, так как из предыдущего пункта следует $ty[y := z] =_\alpha tx[x := z]$.
3. Из второго пункта пункта получаем что $\lambda x.\lambda y.xy =_\alpha \lambda y.\lambda x.yx$ по третьему условию, так как $\lambda y.xy[x := t] =_\alpha \lambda x.yx[y := t]$.

□

Определение 2.7 (β-редекс)**.** β-редекс — выражение вида: $(\lambda x.A)\,B$

Определение 2.8 (β-редукция)**.** $A \to_\beta B$, если имеет место одно из следующих условий:

1. $A \equiv P_1Q_1$, $B \equiv P_2Q_2$ и либо $P_1 =_\alpha P_2$, $Q_1 \to_\beta Q_2$, либо $P_1 \to_\beta P_2$, $Q_1 =_\alpha Q_2$
2. $A \equiv (\lambda x.P)\,Q$, $B \equiv P[x := Q]$ причем Q свободна для подстановки вместо x в P
3. $A \equiv \lambda x.P$, $B \equiv \lambda x.Q$ и $P \to_\beta Q$

Пример. $(\lambda x.x)\,y \to_\beta y$

Пример. $a\,(\lambda x.x)\,y \to_\beta ay$

2.2 Представление некоторых функций в лямбда исчислении

Логические значения легко представить в терминах λ-исчисления. В самом деле, положим:

- $\text{True} \equiv \lambda a \lambda b.a$
- $\text{False} \equiv \lambda a \lambda b.b$

Также мы можем выражать и более сложные функции

Определение 2.9. $\text{If} \equiv \lambda c.\lambda t.\lambda e.(ct)e$

Пример. $\text{If}\ \ \text{T}\ \ a\ b \to_\beta\ \ a$

Доказательство.

$$((\lambda c.\lambda t.\lambda e.(ct)e)\ \lambda a\lambda b.a)\ a\ b \to_\beta (\lambda t.\lambda e.(\lambda a\lambda b.a)\ t\ e)\ a\ b \to_\beta$$
$$(\lambda t.\lambda e.(\lambda b.t)\ e)\ a\ b \to_\beta\ \ (\lambda t.\lambda e.t)\ a\ b \to_\beta\ \ (\lambda e.a)\ b \to_\beta\ \ a$$

□

Как мы видим If T действительно возвращает результат первой ветки.
Другие логические операции:

$$\text{Not} = \lambda a.a\ \ \text{F}\ \ \text{T} \qquad \text{Add} = \lambda a.\lambda b.a\ b\ \ \text{F} \qquad \text{Or} = \lambda a.\lambda b.a\ \ \text{T}\ \ b$$

2.3 Черчевские нумералы

Определение 2.10 (черчевский нумерал)**.**

$$\overline{n} = \lambda f.\lambda x.f^n x, \quad \text{где} \quad f^n x = \begin{cases} f\,(f^{n-1}x) & \text{при } n > 0 \\ x & \text{при } n = 0 \end{cases}$$

Пример.

$$\overline{3} = \lambda f.\lambda x.f\,(f\,(fx))$$

Несложно определить прибавление единицы к такому нумералу:

$$(+1) = \lambda n.\lambda f.\lambda x.f(nfx)$$

Арифметические операции:

1. $\text{IsZero} = \lambda n.n\,(\lambda x.\,\text{F})\,\text{T}$
2. $\text{Add} = \lambda a.\lambda b.\lambda f.\lambda x.a\,f\,(b\,f\,x)$
3. $\text{Pow} = \lambda a.\lambda b.b\,(\text{Mul}\ \ a)\,\overline{1}$
4. $\text{IsEven} = \lambda n.n\ \ \text{Not}\ \ \text{T}$
5. $\text{Mul} = \lambda a.\lambda b.a\,(\text{Add}\ \ b)\,\overline{0}$

Для того, чтобы определить (-1), сначала определим пару:

$$\langle a, b\rangle = \lambda f.f\,a\,b \qquad \text{First} = \lambda p.p\,\text{T} \qquad \text{Second} = \lambda p.p\,\text{F}$$

Затем n раз применим функцию $f\,(\langle a, b\rangle) = \langle b, b+1\rangle$ и возьмём первый элемент пары:

$$(-1) = \lambda n.\,\text{First}(n\,(\lambda p.\,\langle(\text{Second}\,p)\,, (+1)\,(\text{Second}\,p)\rangle)\,\langle\overline{0}, \overline{0}\rangle)$$

3 Лекция 2

3.1 Формализация λ-термов, классы α-эквивалентности термов

Определение 3.1 (λ-терм)**.** Рассмотрим классы эквивалентности $[A]_{=_\alpha}$
Будем говорить, что $[A] \to_\beta [B]$, если существуют $A' \in [A]$ и $B' \in [B]$, что $A' \to_\beta B'$.

Лемма 3.1. $(=_\alpha)$ — отношение эквивалентности.

Пусть в А есть β-редекс $(\lambda x.P)Q$, но Q не свободен для подстановски вместо x в P, тогда найдем $y \notin V[P]$, $y \notin V[Q]$. Сделаем замену $P[x := y]$. Тогда замена $P[x := y][y := Q]$ допустима. То есть, можно сказать, что мы просто переименовали переменную x в P и получили свободу для подстановки, тем самым получив возможность редукции.

Лемма 3.2. $P[x := Q] =_\alpha P[x := y][y := Q]$, если замена допустима.

3.2 Нормальная форма, λ-выражения без нормальной формы, комбинаторы K, I, Ω

Определение 3.2. λ-выражение A находится в нормальноф форме, если оно не содержит β-редексов.

Определение 3.3. A — нормальная форма B, если существует последовательность термов $A_1 ... A_n$ такая, что $B =_\alpha A_1 \to_\beta A_2 \to_\beta ... \to_\beta A_n =_\alpha A$.

Определение 3.4. Комбинатор — λ-выражение без свободных переменных.

Определение 3.5.

- $I \equiv \lambda x.x$ (Identitant)
- $K \equiv \lambda a.\lambda b.a$ (Konstanz)
- $\Omega \equiv (\lambda x.xx)(\lambda x.xx)$

Лемма 3.3. Ω — не имеет нормальной формы.

Доказательство. Ω Имеет единтсвенный β-редекс, где $A \equiv xx$, $B \equiv (\lambda x.xx)$. Тогда единственный возможный путь редукции — подставить B вместо x в A. Но тогда мы получим Ω. Следовательно у Ω нет нормальной формы, так как в полученном выражении у нас всегда будет β-редекс. $\square$

3.3 β-редуцируемость

Определение 3.6. Будем говорить, что $A \twoheadrightarrow_\beta B$, если $\exists$ такие $X_1 .. X_n$, что $A =_\alpha X_1 \to_\beta X_2 \to_\beta ... \to_\beta X_{n-1} \to_\beta X_n =_\alpha B$.

$(\twoheadrightarrow_\beta)$ — рефлексивное и транзитивное замыкание $(\to_\beta)$. $(\twoheadrightarrow_\beta)$ не обязательно приводит к нормальной форме

Пример. $\Omega \twoheadrightarrow_\beta \Omega$

3.4 Ромбовидное свойство

Определение 3.7 (Ромбовидное свойство)**.** Отношение R обладает ромбовидным свойством, если $\forall a, b, c$, таких, что aRb, aRc, $b \neq c$, $\exists d$, что bRd и cRd.

Пример. $(\leqslant)$ на множестве натуральных чисел обладает ромбовидным свойством, $(>)$ на множестве натуральных чисел не обладает ромбовидным свойством.

3.5 Теорема Чёрча-Россер, следствие о единственности нормальной формы

Теорема 3.1 (Черча-Россера)**.** $(\twoheadrightarrow_\beta)$ обладает ромбовидным свойством.

Следствие 3.1. Если у A есть нормальная форма, то она единтсвенная с точностью до $(=_\alpha)$ (переименования переменных).

Доказательство. Пусть $A \twoheadrightarrow_\beta B$ и $A \twoheadrightarrow_\beta C$. B, C — нормальные формы и $B \neq_\alpha C$. Тогда по теореме Черча-Россера $\exists D$: $B \twoheadrightarrow_\beta D$ и $C \twoheadrightarrow_\beta D$. Тогда $B =_\alpha D$ и $C =_\alpha D \Rightarrow B =_\alpha C$. Противоречие. $\square$

Лемма 3.4. Если B — нормальная форма, то не существует Q такой, что $B \to_\beta Q$. Значит если $B \twoheadrightarrow_\beta Q$, то количество шагов редукции равно 0.

Лемма 3.5. Если R — обладает ромбовидным свойством, то и R^* (транзитивное, рефлексивное замыкание R) им обладает.

Доказательство. Пусть $M_1 R^* M_n$ и $M_1 R N_1$. Тогда существуют такие $M_2 \dots M_{n-1}$, что $M_1 R M_2 \dots M_{n-1} R M_n$. Так как R обладает ромбовидным свойством, $M_1 R M_2$ и $M_1 R N_1$, то существует такое N_2, что $N_1 R N_2$ и $M_2 R N_2$. Аналогично, существуют такие $N_3 \dots N_n$, что $N_{i-1} R N_i$ и $M_i R N_i$. Мы получили такое N_n, что $N_1 R^* N_n$ и $M_n R^* N_n$.

Пусть теперь $M_{1,1} R^* M_{1,n}$ и $M_{1,1} R^* M_{m,1}$, то есть имеются $M_{1,2} \dots M_{1,n-1}$ и $M_{2,1} \dots M_{m-1,1}$, что $M_{1,i-1} R M_{1,i}$ и $M_{i-1,1} R M_{i,1}$. Тогда существует такое $M_{2,n}$, что $M_{2,1} R^* M_{2,n}$ и $M_{1,n} R^* M_{2,n}$. Аналогично, существуют такие $M_{3,n} \dots M_{m,n}$, что $M_{i,1} R^* M_{i,n}$ и $M_{1,n} R^* M_{i,n}$. Тогда $M_{1,n} R^* M_{m,n}$ и $M_{m,1} R^* M_{m,n}$. $\square$

Лемма 3.6 (Грустная лемма)**.** $(\to_\beta)$ не обладает ромбовидным свойством.

Доказательство. Пусть $A = (\lambda x.xx)(\mathcal{I}\mathcal{I})$. Покажем что в таком случае не будет выполняться ромбовидное свойство: $\square$

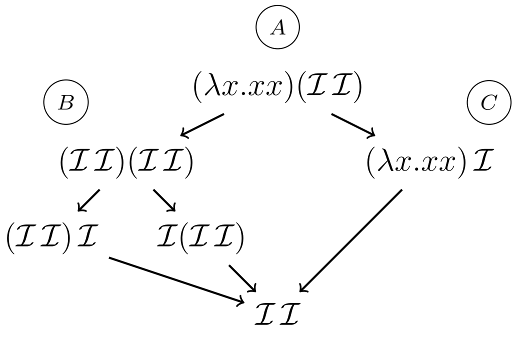

Рис. 1: Нет такого D, что $B \to_\beta D$ и $C \to_\beta D$.

Определение 3.8 (Параллельная β-редукция)**.** $A \rightrightarrows_\beta B$, если

1. $A =_\alpha B$
2. $A \equiv P_1Q_1$, $B \equiv P_2Q_2$ и $P_1 \rightrightarrows_\beta P_2$, $Q_1 \rightrightarrows_\beta Q_2$
3. $A \equiv \lambda x.P_1$, $B \equiv \lambda x.P_2$ и $P_1 \rightrightarrows_\beta P_2$
4. $A =_\alpha (\lambda x.P_1)Q_1$, $B =_\alpha P_2[x := Q_2]$ причем Q_2 свободна для подстановки вместо x в P_2 и $P_1 \rightrightarrows_\beta P_2$, $Q_1 \rightrightarrows_\beta Q_2$

Лемма 3.7. Если $P_1 \rightrightarrows_\beta P_2$ и $Q_1 \rightrightarrows_\beta Q_2$, то $P_1[x := Q_1] \rightrightarrows_\beta P_2[x := Q_2]$

Доказательство. Будем доказывать индукцией по определению $\rightrightarrows_\beta$. Рассмотрим случаи:

- Пусть $P_1 =_\alpha P_2$. Тогда лемма легко доказывается индукцией по структуре выражения.
- Пусть $P_1 \equiv A_1B_1$, $P_2 \equiv A_2B_2$. По определению $(\rightrightarrows_\beta)$ $A_1 \rightrightarrows_\beta A_2$ и $B_1 \rightrightarrows_\beta B_2$. Рассмотрим два случая:
  1. $x \in \mathrm{FV}(A_1)$. По индукционному предположению $A_1[x := Q_1] \rightrightarrows_\beta A_2[x := Q_2]$. Тогда $A_1[x := Q_1]B_1 \rightrightarrows_\beta A_2[x := Q_2]B_2$. Тогда $A_1B_1[x := Q_1] \rightrightarrows_\beta A_2B_2[x := Q_2]$.
  2. $x \in \mathrm{FV}(B_1)$. По индукционному предположению $B_1[x := Q_1] \rightrightarrows_\beta B_2[x := Q_2]$. Тогда $A_1B_1[x := Q_1] \rightrightarrows_\beta A_2B_2[x := Q_2]$.
- Пусть $P_1 \equiv \lambda y.A_1$, $P_2 \equiv \lambda y.A_2$. по определению $(\rightrightarrows_\beta)$ $A_1 \rightrightarrows_\beta A_2$. Тогда по индукционному предположению $A_1[x := Q_1] \rightrightarrows_\beta A_2[x := Q_2]$. Тогда $\lambda y.(A_1[x := Q_1]) \rightrightarrows_\beta \lambda y.(A_2[x := Q_2])$ по определению $(\rightrightarrows_\beta)$. Следовательно $\lambda y.A_1[x := Q_1] \rightrightarrows_\beta \lambda y.A_2[x := Q_2]$ по определению подстановки.
- Пусть $P_1 =_\alpha (\lambda y.A_1)B_1$, $P_2 =_\alpha A_2[y := B_2]$ и $A_1 \rightrightarrows_\beta A_2$, $B_1 \rightrightarrows_\beta B_2$. По индукционному предположению получаем, что $A_1[x := Q_1] \rightrightarrows_\beta A_2[x := Q_2]$, $B_1[x := Q_1] \rightrightarrows_\beta B_2[x := Q_2]$. Следовательно по определению $(\rightrightarrows_\beta)$ получаем, что $(\lambda y.A_1[x := Q_1])B_1[x := Q_1] \rightrightarrows_\beta A_2[y := B_2][x := Q_2]$

□

Лемма 3.8. $(\rightrightarrows_\beta)$ обладает ромбовидным свойством.

Доказательство. Будем доказывать индукцией по определению $(\rightrightarrows_\beta)$. Покажем, что если $M \rightrightarrows_\beta M_1$ и $M \rightrightarrows_\beta M_2$, то существует M_3, что $M_1 \rightrightarrows_\beta M_3$ и $M_2 \rightrightarrows_\beta M_3$. Рассмотрим случаи:

- Если $M \equiv M_1$, то просто возьмем $M_3 \equiv M_2$.
- Если $M \equiv \lambda x.P, M_1 \equiv \lambda x.P_1, M_2 \equiv \lambda x.P_2$ и $P \rightrightarrows_\beta P_1, P \rightrightarrows_\beta P_2$, то по предположению индукции существует P_3, что $P_1 \rightrightarrows_\beta P_3, P_2 \rightrightarrows_\beta P_3$, тогда возьмем $M_3 \equiv \lambda x.P_3$.
- Если $M \equiv PQ, M_1 \equiv P_1Q_1$ и по определению $(\rightrightarrows_\beta)$ $P \rightrightarrows_\beta P_1, Q \rightrightarrows_\beta Q_1$, то рассмотрим два случая:
  1. $M_2 \equiv P_2Q_2$. Тогда по предположению индукции существует P_3, что $P_1 \rightrightarrows_\beta P_3, P_2 \rightrightarrows_\beta P_3$. Аналогично для Q. Тогда возьмем $M_3 \equiv P_3Q_3$.
  2. $P \equiv \lambda x.P'$ значит $P_1 \equiv \lambda x.P_1'$ и $P' \rightrightarrows_\beta P_1'$. Пусть тогда $M_2 \equiv P_2[x := Q_2]$, по определению $(\rightrightarrows_\beta)$ $P' \rightrightarrows_\beta P_2, Q \rightrightarrows_\beta Q_2$. Тогда по предположению индукции и лемме 3.7 существует $M_3 \equiv P_3[x := Q_3]$ такой, что $P_1' \rightrightarrows_\beta P_3$, $Q_1 \rightrightarrows_\beta Q_3$ и $P_2 \rightrightarrows_\beta P_3$, $Q_2 \rightrightarrows_\beta Q_3$.

- Если $M \equiv (\lambda x.P)Q$, $M_1 \equiv P_1[x := Q_1]$ и $P \rightrightarrows_\beta P_1$, $Q \rightrightarrows_\beta Q_1$, то рассмотрим случаи:
  1. $M_2 \equiv (\lambda x.P_2)Q_2$, $P \rightrightarrows_\beta P_2$, $Q \rightrightarrows_\beta Q_2$. Тогда по предположению индукции и лемме 3.7 существует такой $M_3 \equiv P_3[x := Q_3]$, что $P_1 \rightrightarrows_\beta P_3$, $Q_1 \rightrightarrows_\beta Q_3$ и $P_2 \rightrightarrows_\beta P_3$, $Q_2 \rightrightarrows_\beta Q_3$.
  2. $M_2 \equiv P_2[x := Q_2]$, $P \rightrightarrows_\beta P_2$, $Q \rightrightarrows_\beta Q_2$. Тогда по предположению индукции и лемме 3.7 существует такой $M_3 \equiv P_3[x := Q_3]$, что $P_1 \rightrightarrows_\beta P_3$, $Q_1 \rightrightarrows_\beta Q_3$ и $P_2 \rightrightarrows_\beta P_3$, $Q_2 \rightrightarrows_\beta Q_3$.

□

Лемма 3.9.

1. $(\rightrightarrows_\beta)^* \subseteq (\rightarrow_\beta)^*$
2. $(\rightarrow_\beta)^* \subseteq (\rightrightarrows_\beta)^*$

Следствие 3.2. $(\rightarrow_\beta)^* = (\rightrightarrows_\beta)^*$

Из приведенных выше лемм и следствия докажем теорему Черча-Россера.

Доказательство. $(\rightarrow_\beta)^* = (\twoheadrightarrow_\beta)$. Тогда $(\twoheadrightarrow_\beta) = (\rightrightarrows_\beta)^*$. Значит из того, что $(\rightrightarrows_\beta)$ обладает ромбовидным свойством и леммы 3.5 следует, что $(\twoheadrightarrow_\beta)$ обладает ромбовидным свойством.

□

3.6 Нормальный и аппликативный порядок вычислений

Пример. Выражение $KI\Omega$ можно редуцировать двумя способами:

1. $\mathcal{K}\mathcal{I}\,\Omega =_\alpha ((\lambda a.\lambda b.a)\,\mathcal{I})\Omega \rightarrow_\beta (\lambda b.\,\mathcal{I})\Omega \rightarrow_\beta \mathcal{I}$
2. $\mathcal{K}\mathcal{I}\,\Omega =_\alpha ((\lambda a.\lambda b.a)I)((\lambda x.x\ x)(\lambda x.x\ x)) \twoheadrightarrow_\beta ((\lambda a.\lambda b.a)I)((\lambda x.x\ x)(\lambda x.x\ x)) \rightarrow_\beta \mathcal{K}\mathcal{I}\,\Omega$

Как мы видим, в первом случае мы достигли нормальной формы, в то время как во втором мы получаем бесонечную редукцию. Разница двух этих способов в порядке редукции. Первый называется нормальный порядок, а второй аппликативный.

Определение 3.9 (нормальный порядок редукции)**.** Редукция самого левого β-редекса.

Определение 3.10 (аппликативный порядок редукции)**.** Редукция самого левого β-редекса из самых вложенных.

Теорема 3.2 (Приводится без доказательсвта)**.** Если нормальная форма существует, она может быть достигнута нормальным порядком редукции.

Нормальный порядок хоть и приводит к нормальной форме ,если она существует, но бывает ситуации в которых аппликативный порядок вычисляется быстрее чем нормальный

Пример. Рассмотрим лямбда выражение $(\lambda x.x\ x\ x\ x)(\mathcal{I}\mathcal{I})$. Попробуем редуцировать его нормальным порядком:

$$(\lambda x.x\,x\,x\,x)(\mathcal{I}\mathcal{I}) \rightarrow_\beta (\mathcal{I}\mathcal{I})(\mathcal{I}\mathcal{I})(\mathcal{I}\mathcal{I})(\mathcal{I}\mathcal{I}) \rightarrow_\beta \mathcal{I}(\mathcal{I}\mathcal{I})(\mathcal{I}\mathcal{I})(\mathcal{I}\mathcal{I}) \rightarrow_\beta (\mathcal{I}\mathcal{I})(\mathcal{I}\mathcal{I})(\mathcal{I}\mathcal{I}) \rightarrow_\beta ... \rightarrow_\beta \mathcal{I}$$

Как мы увидим, в данной ситуации аппликативный порядок редукции оказывается значительно эффективней:

$$(\lambda x.x\ x\ x\ x)(\mathcal{I}\mathcal{I}) \rightarrow_\beta (\lambda x.x\ x\ x\ x)\,\mathcal{I} \rightarrow_\beta \mathcal{I}\mathcal{I}\mathcal{I}\mathcal{I} \rightarrow_\beta \mathcal{I}\mathcal{I}\mathcal{I} \rightarrow_\beta \mathcal{I}\mathcal{I} \rightarrow_\beta \mathcal{I}$$

4 Лекция 3

4.1 Y-комбинатор

Определение 4.1. Комбинатором называется λ-выражение, не имеющее свободных переменных

Определение 4.2. (Y-комбинатор)

$$Y = \lambda f.(\lambda x.f(xx))(\lambda x.f(xx))$$

Очевидно, Y-комбинатор является комбинатором.

Теорема 4.1. $Yf =_\beta f(Yf)$

Доказательство. β-редуцируем выражение Yf

$$\begin{aligned} &=_\beta (\lambda f.(\lambda x.f(xx))(\lambda x.f(xx)))f \\ &=_\beta (\lambda x.f(xx))(\lambda x.f(xx)) \\ &=_\beta f((\lambda x.f(xx))(\lambda x.f(xx))) \\ &=_\beta f(Yf) \end{aligned}$$

Так как при второй редукции мы получили, что $Yf =_\beta (\lambda x.f(xx))(\lambda x.f(xx))$ □

Следствием этого утверждения является теорема о неподвижной точки для бестипового лямбда-ичисления

Теорема 4.2. В лямбда-исчислении каждый терм f имеет неподвижную точку, то есть такое p, что $f\ p =_\beta p$

Доказательство. Возьмём в качестве p терм Yf. По предыдущей теореме, $f(Yf) =_\beta Yf$, то есть Yf является неподвижной точкой для f. Для любого терма f существует терм Yf, значит, у любого терма есть неподвижная точка. □

4.2 Рекурсия

С помощью Y-комбинатора можо определять рекурсивные функции, например, функцию, вычисляющую факториал Чёрчевского нумерала. Для этого определим вспомогательную функцию

$fact' \equiv \lambda f.\lambda n.isZero\ n\ \overline{1}(mul\ n\ f((-1)n))$

Тогда $fact \equiv Y fact'$

Заметим, что $fact\ \overline{n} =_\beta fact'\ (Y\ fact')\ \overline{n} =_\beta fact'\ fact\ \overline{n}$, то есть в тело функции $fact'$ вместо функции f будет подставлена $fact$ (заметим, что это значит, что именно функция $fact$ будет применена к $\overline{n-1}$, то есть это соответсувует нашим представлениям о рекурсии.)

Для понимания того, как это работает, посчитаем $fact\ \overline{2}$

$$\begin{aligned}
&fact\ \overline{2}\\
&=_\beta Y\ fact'\ \overline{2}\\
&=_\beta fact'(Y\ fact'\ \overline{2})\\
&=_\beta (\lambda f.\lambda n.isZero\ n\ \overline{1}(mul\ n\ f((-1)n)))(Y\ fact')\overline{2}\\
&=_\beta isZero\ \overline{2}\ \overline{1}(mul\ \overline{2}\ ((Y\ fact')((-1)\overline{2})))\\
&=_\beta mul\ \overline{2}\ ((Y\ fact')((-1)\overline{2}))\\
&=_\beta mul\ \overline{2}\ (Y\ fact'\ \overline{1})\\
&=_\beta mul\ \overline{2}\ (fact'\ (Y\ fact'\ \overline{1}))
\end{aligned}$$

Раскрывая $fact'\ (Y\ fact'\ \overline{1})$ так же, как мы раскрывали $fact'\ (Y\ fact'\ \overline{2})$, получаем

$$=_\beta mul\ \overline{2}\ (mul\ \overline{1}\ (Y\ fact'\ \overline{0}))$$

Посчитаем $(Y\ fact'\ \overline{0})$.

$$\begin{aligned}
&(Y\ fact'\ \overline{0})\\
&=_\beta fact'\ (Y\ fact')\ \overline{0}\\
&=_\beta (\lambda f.\lambda n.isZero\ n\ \overline{1}(mul\ n\ f((-1)n)))\ (Y\ fact')\ \overline{0}\\
&=_\beta isZero\ \overline{0}\ \overline{1}(mul\ \overline{0}\ ((Y\ fact'))((-1)\overline{0})) =_\beta \overline{1}
\end{aligned}$$

Таким образом,

$$\begin{aligned}
&fact\ \overline{2}\\
&=_\beta mul\ \overline{2}\ (mul\ \overline{1}\ (Y\ fact'\ \overline{0}))\\
&=_\beta mul\ \overline{2}\ (mul\ \overline{1}\ \overline{1}) =_\beta mul\ \overline{2}\ \overline{1} =_\beta \overline{2}
\end{aligned}$$

4.3 Парадокс Карри

Попробуем построить логику на основе λ-исчисления. Введём логический символ $\rightarrow$.

Будем требовать от этого исчисления наличия следующих схем аксиом:

1. $\vdash A \rightarrow A$
2. $\vdash (A \rightarrow (A \rightarrow B)) \rightarrow (A \rightarrow B)$
3. $\vdash A =_\beta B$, тогда $A \rightarrow B$

А так же правила вывода MP:

$$\frac{\vdash A \rightarrow B,\ \vdash A}{\vdash B}$$

Не вводя дополнительные правила вывода и схемы аксиом, покажем, что данная логика является противоречивой. Для чего введём следующие условные обозначения:

$F_\alpha \equiv \lambda x.(x\ x) \rightarrow \alpha$

$\Phi_\alpha \equiv F_\alpha\ F_\alpha \equiv (\lambda x.(x\ x) \rightarrow \alpha)\ (\lambda x.(x\ x) \rightarrow \alpha)$

Редуцируя Φ_α, получаем

$$\begin{aligned} &\Phi_\alpha \\ =_\beta\ &(\lambda x.(x\ x) \to \alpha)\ (\lambda x.(x\ x) \to \alpha) \\ =_\beta\ &(\lambda x.(x\ x) \to \alpha)\ (\lambda x.(x\ x) \to \alpha) \to \alpha \\ =_\beta\ &\Phi_\alpha \to \alpha \end{aligned}$$

Теперь докажем противоречивость введённой логики. Для этого докажем, что в ней выводимо любое утверждение.

1) $\vdash \Phi_\alpha \to \Phi_\alpha \to \alpha$	Так как $\Phi_\alpha =_\beta \Phi_\alpha \to \alpha$
2) $\vdash (\Phi_\alpha \to \Phi_\alpha \to \alpha) \to (\Phi_\alpha \to \alpha)$	Так как $\vdash (A \to (A \to B)) \to (A \to B)$
3) $\vdash \Phi_\alpha \to \alpha$	MP 2, 3
4) $\vdash (\Phi_\alpha \to \alpha) \to \Phi_\alpha$	Так как $\vdash \Phi_\alpha \to \alpha =_\beta \Phi_\alpha$
5) $\vdash \Phi_\alpha$	MP 3, 4
6) $\vdash \alpha$	MP 3, 5

Таким образом, введённая логика оказывается противоречивой.

4.4 Импликационный фрагмент интуиционистского исчисления высказываний

Рассмотрим подмножество ИИВ, со следующей грамматикой:

$\Phi ::= x \mid \Phi \to \Phi \mid (\Phi)$

То есть состоящее только из мерементых и импликаций.

Добавим в него одну схему аксиом

$$\Gamma, \varphi \vdash \varphi$$

И два правила вывода

1. Правило введения импликации:
$$\frac{\Gamma, \varphi \vdash \psi}{\Gamma \vdash \varphi \to \psi}$$

2. Правило удаления импликации:
$$\frac{\Gamma \vdash \varphi \to \psi \qquad \Gamma \vdash \varphi}{\Gamma \vdash \psi}$$

Пример. Докажем $\vdash \varphi \to \psi \to \varphi$

$$\cfrac{\cfrac{\varphi, \psi \vdash \varphi}{\varphi \vdash \psi \to \varphi}\ \text{(Введение импликации)}}{\vdash \varphi \to (\psi \to \varphi)}\ \text{(Введение импликации)}$$

Пример. Докажем $\alpha \to \beta \to \gamma,\ \alpha,\ \beta \vdash \gamma$

$$\cfrac{\cfrac{\alpha \to \beta \to \gamma,\ \alpha,\ \beta \vdash \alpha \to \beta \to \gamma \qquad \alpha \to \beta \to \gamma,\ \alpha,\ \beta \vdash \alpha}{\alpha \to \beta \to \gamma,\ \alpha,\ \beta \vdash \beta \to \gamma} \qquad \alpha \to \beta \to \gamma, \alpha,\ \beta \vdash \beta}{\alpha \to \beta \to \gamma,\ \alpha,\ \beta \vdash \gamma}$$

4.5 Просто типизированное по Карри лямбда-исчисление

Определение 4.3. Тип в просто типизированном лямбда-исчислении по Карри это либо маленькая греческая буква $(\alpha, \phi, \theta, \ldots)$, либо импликация $(\theta_1 \to \theta_2)$

Таким образом, $\Theta ::= \theta_i | \Theta \to \Theta | (\Theta)$

Импликация при этом считается правоассоциативной операцией.

Определение 4.4. Язык просто типизированного лямбда-исчисления это язык бестипового лямбда-исчисления.

Определение 4.5. Контекст Γ это список выражений вида $A : \theta$, где A - лямбда-терм, а θ - тип

Определение 4.6. Просто типизипрованное лямбда-исчисление по Карри.

Рассмотрим исчисление с единственной схемой аксиом:

$$\Gamma, x : \theta \vdash x : \theta, \text{если } x \text{ не входит в } \Gamma$$

И следующими правилами вывода

1. Правило типизации абстракции

$$\frac{\Gamma, x : \varphi \vdash P : \psi}{\Gamma \vdash (\lambda\, x.\, P) : \varphi \to \psi} \text{ если } x \text{ не входит в } \Gamma$$

2. Правило типизации аппликации:

$$\frac{\Gamma \vdash P : \varphi \to \psi \qquad \Gamma \vdash Q : \varphi}{\Gamma \vdash PQ : \psi}$$

Если λ-выражение типизируется с использованием этих двух правил и одной схемы аксиом, то будем говорить, что оно типизируется по Карри.

Пример. Докажем $\vdash \lambda\, x.\, \lambda\, y.\, x : \alpha \to \beta \to \alpha$

$$\cfrac{\cfrac{x : \alpha, y : \beta \vdash x : \alpha}{x : \alpha \vdash \lambda\, y.\, x : \beta \to \alpha} \text{ (Правило типизации абстракции)}}{\vdash \lambda\, x.\, \lambda\, y.\, x : \alpha \to \beta \to \alpha} \text{ (Правило типизации абстракции)}$$

Пример. Докажем $\vdash \lambda\, x.\, \lambda\, y.\, x\ y : (\alpha \to \beta) \to \alpha \to \beta$

$$\cfrac{\cfrac{\cfrac{x : \alpha \to \beta, y : \alpha \vdash x : \alpha \to \beta \qquad x : \alpha \to \beta, y : \alpha \vdash y : \alpha}{x : \alpha \to \beta, y : \alpha \vdash x\ y : \beta}}{x : \alpha \to \beta \vdash \lambda\, y.\, x\ y : \alpha \to \beta}}{\vdash \lambda\, x.\, \lambda\, y.\, x\ y : (\alpha \to \beta) \to \alpha \to \beta}$$

4.6 Отсутствие типа у Y-комбинатора

Теорема 4.3. Y-комбинатор не типизируется в просто типизированном по Карри лямбда исчислении

Неформальное доказательство $Y\ f =_\beta f\ (Y\ f)$, поэтому $Y\ f$ и $f\ (Y\ f)$ должны иметь одинаковые типы.

Пусть $Y\ f : \alpha$

Тогда $Y : \beta \to \alpha, f : \beta$

Из $f\ (Y\ f) : \alpha$ получаем $f : a \to \alpha$ (так как $Yf : \alpha$)

Тогда $\beta = \alpha \to \alpha$, из этого получаем $Y : (\alpha \to \alpha) \to \alpha$

Можно доказать, что $\lambda\ x.\ x : \alpha \to \alpha$. Тогда $Y\ \lambda\ x.\ x : \alpha$, то есть любой тип является обитаемым. Так как это невозможно, Y-комбинатор не может иметь типа, так как тогда он сделает нашу логику противоречивой.

Формальное доказательство Докажем от противного. Пусть Y-комбинатор типизируем. Тогда в выводе его типа есть вывод типа выражения $x\ x$. Так как $x\ x$ - абстракция, то и типизированна она может быть только по правилу абстракции. Значит, в выводе типа Y-комбинатора есть такой вывод:

$$\frac{\Gamma \vdash x : \varphi \to \psi \qquad \Gamma \vdash x : \varphi}{\Gamma \vdash xx : \psi}$$

Рассмотрим типизацию $\Gamma \vdash x : \varphi \to \psi$ и $\Gamma \vdash x : \varphi$. x это атомарная переменная, значит, она могла быть типизирована только по единственной схеме аксиом.

Следовательно, x типизируется следующим образом.

$$\frac{\Gamma', x : \varphi \to \psi, x : \varphi \vdash x : \varphi \to \psi \qquad \Gamma', x : \varphi \to \psi, x : \varphi \vdash x : \varphi}{\Gamma', x : \varphi \to \psi, x : \varphi \vdash xx : \psi}$$

Следовательно, в контексте Γ переменная x встречается два раза, что невозможно по схеме аксиом.

4.7 Изоморфизм Карри-Ховарда

Заметим, что аксиомы и правила вывода импликационного фрагмента ИИВ и просто типизированного по Карри лямбда-исчисления точно соответсвуют друг другу.

Просто типизирпованное λ-исчисление	Импликативный фрагмент ИИВ
$\Gamma, x : \theta \vdash x : \theta$	$\Gamma, \varphi \vdash \varphi$
$\dfrac{\Gamma, x : \varphi \vdash P : \psi}{\Gamma \vdash (\lambda\ x.\ P) : \varphi \to \psi}$	$\dfrac{\Gamma, \varphi \vdash \psi}{\Gamma \vdash \varphi \to \psi}$
$\dfrac{\Gamma \vdash P : \varphi \to \psi \qquad \Gamma \vdash Q : \varphi}{\Gamma \vdash PQ : \psi}$	$\dfrac{\Gamma \vdash \varphi \to \psi \qquad \Gamma \vdash \varphi}{\Gamma \vdash \psi}$

Установим соответствие и между прочими сущностями ИИВ и просто типизированного по Карри лямбда-исчисления.

Просто типизирпованное λ-исчисление	Импликативный фрагмент ИИВ
Тип	Высказывание
Терм	Доказательство высказывания
Проверка того, что терм имеет заданный тип	Проверка доказательства на корректность
Обитаемый тип	Доказуемое высказывание
Проверка того, что существует терм, имеющий заданный тип	Проверка того, что заданное высказывание имеет доказательство

5 Лекция 4

5.1 Расширение просто типизированного λ-исчисления до изоморфного ИИВ

Заметим, что между просто типизированным по Карри λ-исчислением и имликационным фрагментом ИИВ существует изоморфизм, но при этом в просто типизированном λ-исчислении нет аналогов лжи, а также связок $\vee$ и $\&$.

Для установления полного изоморфизма между ИИВ и просто типизированным λ-исчислением введём три необходимые для установления этого изоморфизма сущности:

1. Тип "Ложь"($\perp$)

2. Тип упорядоченной пары $A\&B$, соответсвующий логическому "И"

3. Алгебраический тип $A|B$, сотответсвующий логическому "ИЛИ"

Тип $\perp$ Введём тип $\perp$, сотттветствующий лжи в ИИВ. Поскольку из лжи может следовать что угодно, добавим в исчисление новое правило вывода

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

То есть выражение, типизированное как $\perp$, может быть типизированно так же любым другим типом.

В программировании аналогом этого типа может являться тип `n+ncNothing`, который является подтипом любого другого типа.

Тип `n+ncNothing` является необитаемым, им типизируется выражение, никогда не возвращающее свой результат (например, `kthrow knew n+ncErroro() k: k+ktNothing`).

Тот факт, что выражение, типизированное как `n+ncNothing`, может быть типизировано любым другим типом, позволяет писать следующие функции:

```
kdef nassertStringNotEmptyo(nsk: k+ktStringo)k: k+ktString o= o
  kif o(nso.nlength o!= l+m+mi0o) o
    ns
  o kelse o
    kthrow knew n+ncErroro(l+sEmpty stringo)
  o
o
```

так как `kthrow knew n+ncErroro(l+sEmpty stringo)k: k+ktNothing`, то

`kthrow knew n+ncErroro(l+sEmpty stringo)k: k+ktString`, поэтому функция может иметь тип `n+ncString`.

Теперь, имея тип $\perp$, можно ввести связку "Отрицание". Обозначим $\neg A = A \to \perp$, то есть в программировании это будет соответствовать функции

```
kdef nthrowErroro(nak: k+ktAo)k: k+ktNothing o= kthrow knew n+ncErroro()
```

Упорядоченные пары Введём возможность запаковывать значения в пары. Функция $makePair$ будет выглядеть следующим образом:

$$makePair \equiv \lambda\ first.\ \lambda\ second.\ \lambda\ f.\ f\ first\ second$$

Тогда

$$< first, second > \equiv makePair\ first\ second$$

Надо также написать функции, которые будут доставать из пары упакованные в неё значения. Назовём иъ Π_1 и Π_2.

Пусть

$$\Pi_1 \equiv \lambda\ Pair.\ Pair\ (\lambda\ a.\lambda\ b.\ a)$$

$$\Pi_2 \equiv \lambda\ Pair.\ Pair\ (\lambda\ a.\lambda\ b.\ b)$$

Заметим, что

$$\begin{aligned}
&\Pi_1 < A, B > \\
&=_\beta (\lambda\ Pair.\ Pair\ (\lambda\ a.\lambda\ b.\ a))(makePair\ A\ B) \\
&=_\beta (\lambda\ Pair.\ Pair\ (\lambda\ a.\lambda\ b.\ a))((\lambda\ first.\ \lambda\ second.\ \lambda\ f.\ f\ first\ second)\ A\ B) \\
&=_\beta (\lambda\ Pair.\ Pair\ (\lambda\ a.\lambda\ b.\ a))((\lambda\ second.\ \lambda\ f.\ f\ A\ second)\ B) \\
&=_\beta (\lambda\ Pair.\ Pair\ (\lambda\ a.\lambda\ b.\ a))(\lambda\ f.\ f\ A\ B) \\
&=_\beta (\lambda\ f.\ f\ A\ B)\ (\lambda\ a.\lambda\ b.\ a) \\
&=_\beta (\lambda\ a.\lambda\ b.\ a)\ A\ B \\
&=_\beta (\lambda\ b.\ A)\ B \\
&=_\beta A
\end{aligned}$$

Аналогично, $\Pi_2 < A, B > =_\beta B$

Таким образом, мы умеем запаковывать элементы в пары и доставать элементы из пар. Теперь, добавим к просто типизированному λ-исчислению правила вывода, позволяющие типизировать такие конструкции.

Добавим три новых правила вывода:

1. Правило типизации пары

$$\frac{\Gamma \vdash A : \varphi \qquad \Gamma \vdash B : \psi}{\Gamma \vdash < A, B >: \varphi \& \psi}$$

2. Правило типизации первого проектора:

$$\frac{\Gamma \vdash < A, B >: \varphi \& \psi}{\Gamma \vdash \Pi_1 < M, N >: \varphi}$$

3. Правило типизации второго проектора:

$$\frac{\Gamma \vdash < A, B >: \varphi \& \psi}{\Gamma \vdash \Pi_2 < M, N >: \psi}$$

Алгебраические типы Добавим тип, который является аналогом `kunion` в C++, или алгебраического типа в любом функциональном языке. Это тип, который может содержать одну из двух альтернатив.

Например, тип `n+ncOptionInt k= n+ncNone o| n+ncSome nof n+ncInt` может содержать либо `n+ncNone`, либо `n+ncSome nof n+ncInt`, но не обе альтернативы разом, причём в каждый момент времени известно, какую альтернативу он содержит.

Заметим, что определение алгебраического типа похоже на определение дизъюнкции в ИИВ (в ИИВ если выполнено $\vdash a \vee b$, известно, что из $\vdash a$ и $\vdash b$ выполнено).

Для реализации алгебраических типов в λ-исчислении напишем три функции:

1. in_1, создающее экземпляр алгебраического типа из первой альтернативы, то есть запаковывающее первую альтернативу в алгебраический тип

2. in_2, выполняющее аналогичные действия, но со второй альтернативой.

3. *case*, принимающую три параметра: экземпляр алгебраического типа, функцию, определяющую, что делать, если этот экземпляр был создан из первой альтернативы (то есть с использованием in_1), и функцию, определяющую, что делать, если этот экземпляр был создан из второй альтернативы (то есть с использованием in_2)

Аналогом *case* в программировании является конструкция, известная как pattern-mathcing, или сопоставление с образцом.

```
klet nisEmptyList k+ktlist o= kmatch k+ktlist kwith
       o| n+ncNil o n+nb+bptrue
       o| n+ncConso(, o) o n+nb+bpfalse
o;;
```

Функция in_1 будет выглядеть следующим образом:

$$in_1 \equiv \lambda\, x.\, \lambda\, f.\, \lambda\, g.\, f\ x$$

А in_1 - следующим:

$$in_2 \equiv \lambda\, x.\, \lambda\, f.\, \lambda\, g.\, g\ x$$

То есть in_1 принимает две функции, и применяет первую к x, а in_2 применяет вторую. Тогда *case* будет выглядеть следующим образом:

$$case \equiv \lambda\, algebraic.\, \lambda\, f.\, \lambda\, g.\, algebraic\ f\ g$$

Заметим, что

$$\begin{aligned}
&case\ (in_1 A)\ F\ G \\
=_\beta\ &(\lambda\, algebraic.\, \lambda\, f.\, \lambda\, g.\, algebraic\ f\ g)\ ((\lambda\, x.\, \lambda\, h.\, \lambda\, s.\, h\ x) A)\ F\ G \\
=_\beta\ &(\lambda\, algebraic.\, \lambda\, f.\, \lambda\, g.\, algebraic\ f\ g)\ (\lambda\, h.\, \lambda\, s.\, h\ A)\ F\ G \\
=_\beta\ &(\lambda\, f.\, \lambda\, g.\, (\lambda\, h.\, \lambda\, s.\, h\ A)\ f\ g)\ F\ G \\
=_\beta\ &(\lambda\, g.\, (\lambda\, h.\, \lambda\, s.\, h\ A)\ F\ g)\ G \\
=_\beta\ &(\lambda\, h.\, \lambda\, s.\, h\ A)\ F\ G \\
=_\beta\ &(\lambda\, s.\, F\ A)\ G \\
=_\beta\ &F\ A
\end{aligned}$$

Аналогично, $case\ (in_2 B)\ F\ G =_\beta G\ B$.

То есть $case, in_1$ и in_2 умеют применять нужную функцию к запакованной в экземпляр алгебраического типа одной из альтернатив.

Теперь добавим к просто типизированному λ-исчислению правила вывода, позволяющие типизировать эти конструкции.

Добавим три новых правила вывода:

1. Правило типизации левой инъекции

$$\frac{\Gamma \vdash A : \varphi}{\Gamma \vdash in_1\ A : \varphi \vee \psi}$$

2. Правило типизации правой инъекции:

$$\frac{\Gamma \vdash B : \psi}{\Gamma \vdash in_2\ B : \varphi \vee \psi}$$

3. Правило типизации case:

$$\frac{\Gamma \vdash L : \varphi \vee \psi, \quad \Gamma \vdash f : \varphi \to \tau, \quad \Gamma \vdash g : \psi \to \tau}{case\ L\ f\ g : \tau}$$

5.2 Изоморфизм Карри-Ховарда для расширения просто типизированного λ-исчисления

Заметим точное соответствие только что введённых конструкций аксиомам ИИВ.

Расширенное просто типизирпованное λ-исчисление	ИИВ
$\dfrac{\Gamma \vdash A : \varphi \qquad \Gamma \vdash B : \psi}{\Gamma \vdash < A, B >: \varphi \& \psi}$	$\vdash \varphi \to \psi \to \varphi \& \psi$
$\dfrac{\Gamma \vdash < A, B >: \varphi \& \psi}{\Gamma \vdash \Pi_1 < M, N >: \varphi}$	$\vdash \varphi \& \psi \to \varphi$
$\dfrac{\Gamma \vdash < A, B >: \varphi \& \psi}{\Gamma \vdash \Pi_2 < M, N >: \psi}$	$\vdash \varphi \& \psi \to \psi$
$\dfrac{\Gamma \vdash A : \varphi}{\Gamma \vdash in_1\ A : \varphi \vee \psi}$	$\vdash \varphi \to \varphi \vee \psi$
$\dfrac{\Gamma \vdash B : \psi}{\Gamma \vdash in_2\ B : \varphi \vee \psi}$	$\vdash \psi \to \varphi \vee \psi$
$\dfrac{\Gamma \vdash L : \varphi \vee \psi,\ \ \Gamma \vdash f : \varphi \to \tau,\ \ \Gamma \vdash g : \psi \to \tau}{case\ L\ f\ g : \tau}$	$\vdash (\varphi \to \tau) \to (\psi \to \tau) \to (\varphi \vee \psi) \to \tau$

5.3 Просто типизированное по Чёрчу λ-исчисление

Определение 5.1. Тип в просто типизированном по Чёрчу λ-исчислении это то же самое, что тип в просто типизированном по Карри λ-исчислении

Определение 5.2. Язык просто типизированного по Чёрчу λ-исчисления удовлетворяет следующей грамматике

$\Lambda_ч ::= x \mid \Lambda_ч\ \Lambda_ч \mid \lambda\ x^\tau.\ \Lambda_ч \mid (\Lambda_ч)$

Замечание 5.1. Иногда абстракция записывается не как $\lambda\ x^\tau.\ \Lambda_ч$, а как $\lambda\ x : \tau.\ \Lambda_ч$

Определение 5.3. Просто типизипрованное по Чёрчу λ-исчисление.

Рассмотрим исчисление с единственной схемой аксиом:

$$\Gamma, x : \theta \vdash x : \theta, \text{если } x \text{ не входит в } \Gamma$$

И следующими правилами вывода

1. Правило типизации абстракции

$$\frac{\Gamma, x : \varphi \vdash P : \psi}{\Gamma \vdash (\lambda\ x : \varphi.\ P) : \varphi \to \psi} \text{ если } x \text{ не входит в } \Gamma$$

2. Правило типизации аппликации:

$$\frac{\Gamma \vdash P : \varphi \to \psi \qquad \Gamma \vdash Q : \varphi}{\Gamma \vdash PQ : \psi}$$

Если λ-выражение типизируется с использованием этих двух правил и одной схемы аксиом, то будем говорить, что оно типизируется по Чёрчу.

В исчислении по Чёрчу остаются верными все предыдущие теоремы (в том числе теорема Чёрча-Россера), но правило строгой типизации абстракций позволяет доказать ещё одну теорему:

Теорема 5.1 (Уникальность типов в исчислении по Чёрчу)**.**

1. Если $\Gamma \vdash_ч M : \theta$ и $\Gamma \vdash_ч M : \tau$, то $\theta = \tau$

2. Если $\Gamma \vdash_ч M : \theta$ и $\Gamma \vdash_ч N : \tau$,и $M =_\beta N$ то $\theta = \tau$

5.4 Связь типизации по Чёрчу и по Карри

Определение 5.4 (Стирание)**.** Функцией стирания называется следующая функция:

$|\cdot| : \Lambda_ч \to \Lambda_к$:

$$|A| = \begin{cases} x & A \equiv x \\ |M|\ |N| & A \equiv M\ N \\ \lambda x.|P| & A \equiv \lambda\ x : \tau.\ P \end{cases}$$

Лемма 5.1. Пусть $M, N \in \Lambda_ч, M \to_\beta N$, тогда $|M| \to_\beta |N|$

Лемма 5.2. Если $\Gamma \vdash_ч M : \tau$, тогда $\Gamma' \vdash_к |M| : \tau$, где Γ' получается из Γ применением функции стирания к каждому терму из Γ

Теорема 5.2 (Теорема о поднятии)**.**

1. Пусть $M, N \in \Lambda_к, P \in \Lambda_ч, |P| = M, M \to_\beta N$. Тогда найдётся такое $Q \in \Lambda_ч$, что $|Q| = N$, и $P \to_\beta Q$

2. Пусть $M \in \Lambda_к, \Gamma \vdash_к M : \tau$. Тогда существует $P \in \Lambda_ч$, что $|P| = M$, и $\Gamma \vdash_ч P : \tau$