Типы в языках программирования
Лекция 9. Система $\lambda\omega$: операторы над типами

Денис Николаевич Москвин

ИТМО, корпоративная магистратура JetBrains
Разработка ПО / Software Engineering

21.04.2021

План лекции

1. λ-куб
2. Система $\lambda\omega$: прагматика
3. Система $\lambda\underline{\omega}$: формализм
4. Примеры для $\lambda\underline{\omega}$ и $\lambda\omega$
5. Связь $\lambda\omega$ с логикой

План лекции

1. λ-куб
2. Система $\lambda\omega$: прагматика
3. Система $\lambda\underline{\omega}$: формализм
4. Примеры для $\lambda\underline{\omega}$ и $\lambda\omega$
5. Связь $\lambda\omega$ с логикой

Зависимости

- Терм от терма $P\,Q$ (система $\lambda\!\to$ или STT)

$\lambda x^{\sigma}.\,F[x] : \sigma \to \tau$ — функция, отображающая терм $M{:}\sigma$ в терм $F[x := M] : \tau$.

- Терм от типа $P\,\rho$ (система $\lambda 2$ или System F)

$\lambda \alpha^{*}.\,F[\alpha] : \forall \alpha.\,\sigma[\alpha]$ — функция, отображающая тип $\tau{:}*$ в терм $F[\alpha := \tau] : \sigma[\alpha := \tau]$.

Возможны и зависимости со значениями в «царстве типов»:

- Тип от типа $\varphi\,\rho$ ($\lambda\underline{\omega}$ — *операторы над типами*)

$\lambda \alpha^{*}.\,\sigma[\alpha] : * \to *$ — функция, отображающая тип $\tau{:}*$ в тип $\sigma[\alpha := \tau] : *$.

- Тип от терма $\varphi\,Q$ (λP — *зависимые типы, семейства типов*)

$\lambda x^{\sigma}.\,\tau[x] : \sigma \to *$ — функция, отображающая терм $M{:}\sigma$ в тип $\tau[x := M] : *$.

λ-куб Барендрегта

$$
\begin{tikzcd}
 & \lambda\omega \arrow[rr] & & \lambda P\omega \\
\lambda 2 \arrow[ur] \arrow[rr] & & \lambda P2 \arrow[ur] & \\
 & \lambda\underline{\omega} \arrow[uu] \arrow[rr] & & \lambda P\underline{\omega} \arrow[uu] \\
\lambda_{\to} \arrow[uu] \arrow[ur] \arrow[rr] & & \lambda P \arrow[uu] \arrow[ur] &
\end{tikzcd}
$$

План лекции

1. λ-куб
2. Система $\lambda\omega$: прагматика
3. Система $\lambda\underline{\omega}$: формализм
4. Примеры для $\lambda\underline{\omega}$ и $\lambda\omega$
5. Связь $\lambda\omega$ с логикой

$$\begin{array}{rcl} \mathtt{Pair}\sigma\tau & \equiv & \forall\alpha.\,(\sigma\to\tau\to\alpha)\to\alpha \\ \sigma{:}*,\tau{:}* \vdash \mathtt{pair} & : & \sigma\to\tau\to\mathtt{Pair}\sigma\tau \\ \mathtt{pair} & \equiv & \lambda a^{\sigma}\,b^{\tau}.\,\Lambda\,\alpha.\,\lambda\,f^{\sigma\to\tau\to\alpha}.\,f\,a\,b \end{array}$$

Полиморфизм по α позволяет «итерировать» универсально:

$$\begin{array}{rcl} \mathtt{uncurry} & : & (\sigma\to\tau\to\rho)\to\mathtt{Pair}\sigma\tau\to\rho \\ \mathtt{uncurry} & \equiv & \lambda f^{\sigma\to\tau\to\rho}\,p^{\mathtt{Pair}\sigma\tau}.\,p\,\rho\,f \end{array}$$

$\forall$ для σ и τ даст лишь лишнюю упаковку/распаковку терма:

$$\begin{array}{rcl} \mathtt{Pair2} & \equiv & \forall\sigma\,\tau\,\alpha.\,(\sigma\to\tau\to\alpha)\to\alpha \\ \mathtt{pair2?} & \equiv & \lambda a^{\sigma}\,b^{\tau}.\,\Lambda\sigma\,\tau\,\alpha.\,\lambda\,f^{\sigma\to\tau\to\alpha}.\,f\,a\,b \\ \mathtt{pair2} & \equiv & \Lambda\sigma\,\tau.\,\lambda a^{\sigma}\,b^{\tau}.\,\Lambda\alpha.\,\lambda\,f^{\sigma\to\tau\to\alpha}.\,f\,a\,b \\ \mathtt{pair2}\ \mathtt{Nat}\ \mathtt{Bool}\ \overline{3}\ \mathtt{tru} & : & \mathtt{Pair}\sigma\tau[\sigma:=\mathtt{Nat}][\tau:=\mathtt{Bool}] \end{array}$$

Такая же проблема у списков: $[\overline{3},\overline{7},\overline{42}] = \lambda c\, n.\, c\, \overline{3}\, (c\, \overline{7}\, (c\, \overline{42}\, n))$
Если $n : \alpha$, то $c : \mathtt{Nat} \to \alpha \to \alpha$ и

$$\begin{aligned} \mathtt{ListNat} &\equiv \forall\alpha.\, (\mathtt{Nat} \to \alpha \to \alpha) \to \alpha \to \alpha \\ \mathtt{List}\sigma &\equiv \forall\alpha.\, (\sigma \to \alpha \to \alpha) \to \alpha \to \alpha \end{aligned}$$

Конструкторы для $\forall\sigma.\, \mathtt{List}\sigma$

$$\begin{aligned} \mathtt{nil} &: \forall\sigma.\, \mathtt{List}\sigma \\ \mathtt{nil} &\equiv \Lambda\sigma.\, \Lambda\alpha.\, \lambda c^{\sigma\to\alpha\to\alpha}\, n^{\alpha}.\, n \\ \mathtt{cons} &: \forall\sigma.\, \sigma \to \mathtt{List}\sigma \to \mathtt{List}\sigma \\ \mathtt{cons} &\equiv \Lambda\sigma\, .\lambda x^{\sigma}\, l^{\mathtt{List}\sigma}.\, \Lambda\alpha.\, \lambda c^{\sigma\to\alpha\to\alpha}\, n^{\alpha}.\, c\, x\, (l\, \alpha\, n\, c) \end{aligned}$$

Но хочется уметь превращать σ в $\mathtt{Nat}$ без аппликаций к $\mathtt{Nat}$

$$[\overline{3},\overline{7},\overline{42}] = \mathtt{cons}\ \mathtt{Nat}\ \overline{3}\ (\mathtt{cons}\ \mathtt{Nat}\ \overline{7}\ (\mathtt{cons}\ \mathtt{Nat}\ \overline{42}\ (\mathtt{nil}\ \mathtt{Nat})))$$

Система $\lambda\omega$: абстракция в типах

Идея: разрешить абстракцию на уровне типов

$$\mathtt{Pair} \quad \equiv \quad \lambda\sigma^*.\,\lambda\tau^*.\,\forall\alpha.\,(\sigma\to\tau\to\alpha)\to\alpha$$

аппликацию типа к типу и редукцию над типами

$$\begin{array}{rcl}
\mathtt{Pair}\ \mathtt{Nat} & \equiv & (\lambda\sigma^*.\,\lambda\tau^*.\,\forall\alpha.\,(\sigma\to\tau\to\alpha)\to\alpha)\ \mathtt{Nat} \\
 & \longrightarrow_\beta & \lambda\tau^*.\,\forall\alpha.\,(\mathtt{Nat}\to\tau\to\alpha)\to\alpha \\
(\mathtt{Pair}\ \mathtt{Nat})\ \mathtt{Bool} & \equiv & (\lambda\tau^*.\,\forall\alpha.\,(\mathtt{Nat}\to\tau\to\alpha)\to\alpha)\ \mathtt{Bool} \\
 & \longrightarrow_\beta & \forall\alpha.\,(\mathtt{Nat}\to\mathtt{Bool}\to\alpha)\to\alpha
\end{array}$$

Но при этом могут возникнуть бессмысленные аппликации:

$$\mathtt{Nat}\ \mathtt{Bool}, \quad \mathtt{Pair}\ \mathtt{Pair}, \quad \mathtt{Pair}\ (\mathtt{Pair}\ \mathtt{Nat}\ \mathtt{Bool})\ \mathtt{Nat}$$

Идея: ввести систему типов над системой типов — ***кайнды*** (***виды***).
Для *простых типов* вид один — $*$. Только такие типы используются для типизации термов:

$$\top : *,\quad \mathtt{Nat} : *,\quad \mathtt{Bool} : *,\quad \mathtt{ListNat} : *$$

Для *операторов над типами* имеется «стрелочный» кайнд

$$\begin{array}{rcl}
\mathtt{List} & : & * \to * \\
\mathtt{List} & \equiv & \lambda\sigma^{*}.\,\forall\alpha.\,\alpha \to (\sigma \to \alpha \to \alpha) \to \alpha \\
\mathtt{List\ Nat} & : & * \\
\mathtt{List\ Nat} & \longrightarrow_{\beta} & \forall\alpha.\,\alpha \to (\mathtt{Nat} \to \alpha \to \alpha) \to \alpha \\
\mathtt{Pair} & : & * \to * \to *
\end{array}$$

План лекции

1. λ-куб
2. Система $\lambda\omega$: прагматика
3. Система $\lambda\underline{\omega}$: формализм
4. Примеры для $\lambda\underline{\omega}$ и $\lambda\omega$
5. Связь $\lambda\omega$ с логикой

Типы и термы в $\lambda\underline{\omega}$

- Мы перестаем различать типы и термы.
- Множество *псевдовыражений* определяется индуктивно:

$$\mathcal{T} = V \mid C \mid \mathcal{T}\,\mathcal{T} \mid \lambda V : \mathcal{T}\,.\,\mathcal{T} \mid \mathcal{T} \to \mathcal{T},$$

  где V — бесконечный набор переменных, а $C = \{*, \Box\}$.
- Константы $*$ и $\Box$ называют *сортами*.
- В дальнейшем переменная s пробегает множество $\{*, \Box\}$.

Контексты для $\lambda\underline{\omega}$

- *Высказывание* в $\lambda\underline{\omega}$ имеет вид $M : A$, где $M, A \in \mathcal{T}$.
- *Контекст* — это конечное, линейно упорядоченное множество высказываний, с различными переменными в качестве субъекта.
- $\langle\rangle$ обозначает пустой контекст.
- Если $\Gamma = \langle x_1 : A_1, \ldots, x_n : A_n \rangle$, то $\Gamma, y : B = \langle x_1 : A_1, \ldots, x_n : A_n, y : B \rangle$.
- Угловые скобки в контекстах обычно опускают.

Правила типизации для $\lambda\underline{\omega}$ (1)

Утверждение $\Gamma \vdash_{\lambda\underline{\omega}} M : A$ задается правилами:

(аксиома)	$\langle\rangle \vdash * : \square$
(начальное правило)	$\dfrac{\Gamma \vdash A : s}{\Gamma, x : A \vdash x : A}, x \notin \Gamma$
(правило ослабления)	$\dfrac{\Gamma \vdash A : B \quad \Gamma \vdash C : s}{\Gamma, x : C \vdash A : B}, x \notin \Gamma$
(формирование типа/вида)	$\dfrac{\Gamma \vdash A : s \quad \Gamma \vdash B : s}{\Gamma \vdash A \to B : s}$

продолжение на следующем слайде...

Правила типизации для $\lambda\underline{\omega}$ (2)

...начало на предыдущем слайде

(правило применения)	$\dfrac{\Gamma \vdash M : A \to B \quad \Gamma \vdash N : A}{\Gamma \vdash M\,N : B}$
(правило абстракции)	$\dfrac{\Gamma, x : A \vdash M : B \quad \Gamma \vdash A \to B : s}{\Gamma \vdash \lambda x^A.\,M : A \to B}$
(правило преобразования)	$\dfrac{\Gamma \vdash A : B \quad \Gamma \vdash B' : s \quad B =_\beta B'}{\Gamma \vdash A : B'}$

Алгоритмически правило преобразования может потребоваться при проверке совпадения типов формального и фактического аргумента в правиле применения.

План лекции

1. λ-куб
2. Система $\lambda\omega$: прагматика
3. Система $\lambda\underline{\omega}$: формализм
4. Примеры для $\lambda\underline{\omega}$ и $\lambda\omega$
5. Связь $\lambda\omega$ с логикой

Примеры для $\lambda\underline{\omega}$

$$
\begin{array}{rcl}
\alpha : * & \vdash_{\lambda\underline{\omega}} & \alpha : * \\
\kappa : \Box & \vdash_{\lambda\underline{\omega}} & \kappa : \Box \\
\alpha : *, x : \alpha & \vdash_{\lambda\underline{\omega}} & x : \alpha \\
\alpha : *, \beta : * & \vdash_{\lambda\underline{\omega}} & \alpha : * \\
\alpha : *, \beta : * & \vdash_{\lambda\underline{\omega}} & \beta : * \\
\alpha : *, \beta : * & \vdash_{\lambda\underline{\omega}} & \alpha \to \beta : * \\
\alpha : *, \beta : *, x : \alpha \to \beta & \vdash_{\lambda\underline{\omega}} & x : \alpha \to \beta \\
\alpha : *, \beta : * & \vdash_{\lambda\underline{\omega}} & \lambda x^{\alpha \to \beta} . x : (\alpha \to \beta) \to (\alpha \to \beta)
\end{array}
$$

Пусть $D \equiv \lambda\beta : * . \beta \to \beta$. Тогда

$$
\begin{array}{rcl}
 & \vdash_{\lambda\underline{\omega}} & D : * \to * \\
\alpha : * & \vdash_{\lambda\underline{\omega}} & \lambda x^{D\,\alpha} . x : D(D\alpha) \\
\alpha : *, \varphi : * \to * & \vdash_{\lambda\underline{\omega}} & \varphi\,(\varphi\,\alpha) : *; \\
\alpha : * & \vdash_{\lambda\underline{\omega}} & \lambda\varphi^{* \to *} . \varphi\,(\varphi\,\alpha) : (* \to *) \to *
\end{array}
$$

Примеры для $\lambda\omega$ (ДЗ)

Рассмотрим число Черча $\overline{2} \equiv \lambda s\, z.\, s\,(s\, z)$. Покажите, что ему в $\lambda 2$ можно приписать следующие типы:

$$(\forall\alpha.\, \alpha \to \beta \to \alpha) \to \forall\alpha.\, \alpha \to \beta \to \beta \to \alpha$$
$$(\forall\alpha.\, \alpha \to \beta \to \gamma \to \alpha) \to \forall\alpha.\, \alpha \to \beta \to \gamma \to \beta \to \gamma \to \alpha$$

Примеры для $\lambda\omega$ (ДЗ)

Рассмотрим число Черча $\overline{2} \equiv \lambda s\, z.\, s\,(s\, z)$. Покажите, что ему в $\lambda 2$ можно приписать следующие типы:

$$(\forall \alpha.\, \alpha \to \beta \to \alpha) \to \forall \alpha.\, \alpha \to \beta \to \beta \to \alpha$$
$$(\forall \alpha.\, \alpha \to \beta \to \gamma \to \alpha) \to \forall \alpha.\, \alpha \to \beta \to \gamma \to \beta \to \gamma \to \alpha$$

В $\lambda\omega$ можно обобщить эту идею, приписав $\overline{2} \equiv \lambda s\, z.\, s\,(s\, z)$ тип

$$\forall \varphi.\, (\forall \alpha.\, \alpha \to \varphi\, \alpha) \to \forall \alpha.\, \alpha \to \varphi\,(\varphi\, \alpha)$$

Здесь мы в терме абстрагируемся не по простому типу, а по оператору над типами.
Это позволяет, например, типизировать $\overline{2}\,\overline{2}\,\mathbf{K}$, нетипизируемое в $\lambda 2$.

План лекции

1. λ-куб
2. Система $\lambda\omega$: прагматика
3. Система $\lambda\underline{\omega}$: формализм
4. Примеры для $\lambda\underline{\omega}$ и $\lambda\omega$
5. Связь $\lambda\omega$ с логикой

Система PROPω

- Существует изоморфизм между $\lambda\omega$ и «минимальной» интуиционистская пропозициональная логикой высших порядков PROPω.
- Типы в $\lambda\omega$ — формулы в PROPω.
  Термы в $\lambda\omega$ — доказательства в PROPω.
- PROPω — конструктивная система; в ней, например, закон Пирса

$$\forall \alpha\, \beta.\, ((\alpha \to \beta) \to \alpha) \to \alpha$$

  невыводим.

Импликация унаследована от PROP:

	Правило удаления $\rightarrow$	Правило введения $\rightarrow$
PROPω	$\dfrac{\sigma \rightarrow \tau \quad \sigma}{\tau}$	$\dfrac{\begin{array}{c}[\sigma] \\ \vdots \\ \tau\end{array}}{\sigma \rightarrow \tau}$
$\lambda\omega$	$\dfrac{\Gamma \vdash M : \sigma \rightarrow \tau \quad \Gamma \vdash N : \sigma}{\Gamma \vdash M\,N : \tau}$	$\dfrac{\Gamma, x:\sigma \vdash M:\tau}{\Gamma \vdash \lambda x^{\sigma}.\,M : \sigma \rightarrow \tau}$

PROPω: $\forall$

Квантор $\forall$ унаследован от PROP2:

	Правило удаления $\forall$	Правило введения $\forall$
PROPω	$\dfrac{\forall\alpha.\,\sigma}{\sigma[\alpha := \tau]}$	$\dfrac{\begin{matrix}\tau_1 \dots \tau_n \\ \vdots \\ \sigma\end{matrix}}{\forall\alpha.\,\sigma}, \alpha \notin FV(\tau_1, \dots, \tau_n)$
$\lambda\omega$	$\dfrac{\Gamma \vdash M : \forall\alpha.\,\sigma \quad \Gamma \vdash \tau : *}{\Gamma \vdash M\,\tau : \sigma[\alpha := \tau]}$	$\dfrac{\Gamma, \alpha : * \vdash M : \sigma}{\Gamma \vdash \Lambda\alpha.\,M : \forall\alpha.\,\sigma}$

Например, можно универсально абстрагироваться по γ в
$\alpha : *, \beta : *, g : (\alpha \to \beta), \gamma : * \vdash \lambda\, f^{\beta \to \gamma}\, x^{\alpha}.\, f(g\, x) : (\beta \to \gamma) \to \alpha \to \gamma$

Прочие стандартные связки

Система $\lambda\omega$ позволяет определить стандартные логические связки:

$$
\begin{array}{rcl}
\bot & \equiv & \forall\alpha\,.\,\alpha \\
\neg & \equiv & \lambda\sigma^{*}\,.\,\sigma\to\bot \\
\wedge & \equiv & \lambda\sigma^{*}\tau^{*}\,.\,\forall\alpha.\,(\sigma\to\tau\to\alpha)\to\alpha \\
\vee & \equiv & \lambda\sigma^{*}\tau^{*}\,.\,\forall\alpha.\,(\sigma\to\alpha)\to(\tau\to\alpha)\to\alpha \\
\exists\alpha.\,\sigma & \equiv & \forall\beta.\,(\forall\alpha.\,\sigma\to\beta)\to\beta
\end{array}
$$

Какие у них кайнды?

Свойства $\bot$

Напомним, что $\bot \equiv \forall\alpha.\,\alpha$.
Терм этого типа позволяет населить терм любого типа

$$\begin{array}{rcl} \sigma{:}{*}, x{:}\bot & \vdash & x\,\sigma : \sigma \\ \sigma{:}{*} & \vdash & \lambda x^{\bot}.\, x\,\sigma : \bot \to \sigma \end{array}$$

Однако тип $\bot$ не населён в $\lambda\omega$ — не существует *замкнутого* терма с таким типом.
С логической точки зрения $\bot$ — это абсурд, заведомо ложное утверждение.

В $\lambda\omega$ связка $\neg$ может быть определена так:

$$\neg\sigma \;\equiv\; \sigma \to \bot$$

Правило удаления $\neg$	Правило «введения» $\neg$
$\dfrac{\sigma \qquad \neg\sigma}{\tau}$	$\dfrac{\begin{matrix}[\sigma] \\ \vdots \\ \tau\end{matrix} \qquad \begin{matrix}[\sigma] \\ \vdots \\ \neg\tau\end{matrix}}{\neg\sigma}$

Покажем, что они выразимы в $\lambda\omega$.

- **Удаление $\neg$.** Тип

$$\sigma \to \neg\sigma \to \tau \equiv \sigma \to (\sigma \to \bot) \to \tau \equiv \sigma \to (\sigma \to \forall\alpha.\,\alpha) \to \tau$$

Терм этого типа (в контексте $\Gamma = \sigma{:}*, \tau{:}*$)

$$\lambda x^{\sigma}.\,\lambda f^{\,\sigma\to\bot}.\,f\,x\,\tau$$

Из противоречия следует все что угодно.

- **«Введение»** $\neg$. Тип

$$(\sigma \to \tau) \to (\sigma \to \neg\tau) \to \neg\sigma \equiv (\sigma \to \tau) \to (\sigma \to \tau \to \bot) \to \sigma \to \bot$$

Терм этого типа (в контексте $\Gamma = \sigma{:}*, \tau{:}*$)

$$\lambda f^{\,\sigma\to\tau}.\,\lambda g^{\sigma\to\tau\to\bot}.\,\lambda x^{\sigma}.\,g\,x\,(f\,x)$$

Доказательство приведением к нелепости — *reductio ad absurdum*.

В $\lambda\omega$ связка $\wedge$ может быть определена так:

$$\sigma \wedge \tau \;\equiv\; \forall \alpha.\, (\sigma \to \tau \to \alpha) \to \alpha$$

$\alpha \notin FV(\sigma)$, $\alpha \notin FV(\tau)$.

Правила удаления $\wedge$	Правило введения $\wedge$
$\dfrac{\sigma \wedge \tau}{\sigma} \qquad \dfrac{\sigma \wedge \tau}{\tau}$	$\dfrac{\sigma \qquad \tau}{\sigma \wedge \tau}$

Покажем, что они выразимы в $\lambda\omega$.

Введение $\wedge$. Тип

$$\sigma \to \tau \to (\sigma \wedge \tau) \equiv \sigma \to \tau \to (\forall \alpha.\, (\sigma \to \tau \to \alpha) \to \alpha)$$

Терм этого типа

$$\lambda x^{\sigma}\, y^{\tau}.\, \Lambda\, \alpha.\, \lambda f^{\,\sigma \to \tau \to \alpha}.\, f\, x\, y$$

При использовании $\lambda\omega$ как языка программирования, это пара (двухэлементный кортеж):

$$\sigma{:}*, \tau{:}*, x{:}\sigma, y{:}\tau \vdash \langle x, y \rangle : \sigma \times \tau$$

Удаление $\wedge$ (1). Тип $(\sigma \wedge \tau) \to \sigma \equiv (\forall \alpha.\, (\sigma \to \tau \to \alpha) \to \alpha) \to \sigma$
Терм этого типа

$$\lambda f^{\sigma \wedge \tau}.\, f\, \sigma\, (\lambda x^{\sigma} y^{\tau}.\, x)$$

$$\begin{array}{lcl} \sigma{:}*, \tau{:}*, f : \sigma \wedge \tau & \vdash & f\, \sigma : (\sigma \to \tau \to \sigma) \to \sigma \\ \sigma{:}*, \tau{:}*, f : \sigma \wedge \tau & \vdash & f\, \sigma\, (\lambda x^{\sigma} y^{\tau}.\, x) : \sigma \end{array}$$

Удаление $\wedge$ (2). Тип $(\sigma \wedge \tau) \to \tau \equiv (\forall \alpha.\, (\sigma \to \tau \to \alpha) \to \alpha) \to \tau$
Терм этого типа

$$\lambda f^{\sigma \wedge \tau}.\, f\, \tau\, (\lambda x^{\sigma} y^{\tau}.\, y)$$

$$\begin{array}{lcl} \sigma{:}*, \tau{:}*, f : \sigma \wedge \tau & \vdash & f\, \tau : (\sigma \to \tau \to \tau) \to \tau \\ \sigma{:}*, \tau{:}*, f : \sigma \wedge \tau & \vdash & f\, \tau\, (\lambda x^{\sigma} y^{\tau}.\, y) : \tau \end{array}$$

Програмистская интерпретация — проекции пары `fst` и `snd`.

Связка $\vee$ (дизъюнкция)

В $\lambda\omega$ связка $\vee$ может быть определена так:

$$\sigma \vee \tau \;\equiv\; \forall\alpha.\,(\sigma \to \alpha) \to (\tau \to \alpha) \to \alpha$$

$\alpha \notin \mathrm{FV}(\sigma)$, $\alpha \notin \mathrm{FV}(\tau)$.

Правило удаления $\vee$	Правила введения $\vee$
$$\dfrac{\sigma \vee \tau \qquad \begin{matrix}[\sigma]\\ \vdots \\ \rho\end{matrix} \qquad \begin{matrix}[\tau]\\ \vdots \\ \rho\end{matrix}}{\rho}$$	$$\dfrac{\sigma}{\sigma \vee \tau} \qquad \dfrac{\tau}{\sigma \vee \tau}$$

Покажем, что они выразимы в $\lambda\omega$.

Введение $\vee$ (1). Тип

$$\sigma \to (\sigma \vee \tau) = \sigma \to (\forall \alpha.\, (\sigma \to \alpha) \to (\tau \to \alpha) \to \alpha)$$

Терм этого типа

$$\lambda\, x^{\sigma}.\, \Lambda\, \alpha.\, \lambda\, f^{\sigma\to\alpha}\, g^{\tau\to\alpha}.\, f\, x$$

Введение $\vee$ (2). Тип

$$\tau \to (\sigma \vee \tau) = \tau \to (\forall \alpha.\, (\sigma \to \alpha) \to (\tau \to \alpha) \to \alpha)$$

Терм этого типа

$$\lambda\, y^{\tau}.\, \Lambda\, \alpha.\, \lambda\, f^{\sigma\to\alpha}\, g^{\tau\to\alpha}.\, g\, y$$

Удаление $\vee$. Тип

$$\begin{array}{rl} & (\sigma \vee \tau) \to (\sigma \to \rho) \to (\tau \to \rho) \to \rho \\ = & (\forall \alpha.\, (\sigma \to \alpha) \to (\tau \to \alpha) \to \alpha) \to (\sigma \to \rho) \to (\tau \to \rho) \to \rho \end{array}$$

Терм этого типа

$$\lambda\, h^{\sigma \vee \tau}\, f^{\sigma \to \rho}\, g^{\tau \to \rho}.\, h\, \rho\, f\, g$$

Построение терма

$$\sigma{:}{*}, \tau{:}{*}, \rho{:}{*}, h : \sigma \vee \tau \;\;\vdash\;\; h\, \rho : (\sigma \to \rho) \to (\tau \to \rho) \to \rho$$

Это правило носит название *доказательство разбором случаев*.

Трактовка $\forall\alpha.\,\sigma$:
Функция из типов в термы, которая *любому* типу τ ставит в соответствие терм с типом $\sigma[\alpha := \tau]$.
Для $\sigma = \alpha \to \alpha$ имеем $\sigma[\alpha := \tau] = \tau \to \tau$:

$$\Lambda\alpha.\,\lambda x^{\alpha}.\,x : \forall\alpha.\,\alpha \to \alpha$$
$$(\Lambda\alpha.\,\lambda x^{\alpha}.\,x)\tau \to_{\beta} \lambda x^{\tau}.\,x : \tau \to \tau$$

Трактовка $\exists\alpha.\,\sigma$:
Пара из *некоторого* типа τ и терма, имеющего тип $\sigma[\alpha := \tau]$.
Для $\sigma = \alpha \to \gamma$ имеем:

$$\langle \gamma, \lambda x^{\gamma}.\,x \rangle : \exists\alpha.\,\alpha \to \gamma$$

поскольку $\sigma[\alpha := \gamma] = \gamma \to \gamma$ и $\lambda x^{\gamma}.\,x : \gamma \to \gamma$.

В $\lambda\omega$ квантор $\exists$ может быть определён так:

$$\exists\alpha\,.\,\sigma \;\equiv\; \forall\beta.\,(\forall\alpha.\,\sigma \to \beta) \to \beta$$

$\beta \notin FV(\sigma)$.

Правило удаления $\exists$	Правило введения $\exists$
$\dfrac{\exists\alpha.\,\sigma \qquad \begin{matrix}[\sigma]\\ \vdots \\ \rho\end{matrix}}{\rho}, \quad \alpha \notin FV(\rho)$	$\dfrac{\sigma[\alpha := \tau]}{\exists\alpha.\,\sigma}$

Покажем, что они выразимы в $\lambda\omega$.

Введение ∃. Тип

$$(\lambda\alpha.\,\sigma)\tau \to \exists\alpha.\,\sigma \;=\; \sigma[\alpha := \tau] \to \forall\beta.\,(\forall\alpha.\,\sigma \to \beta) \to \beta$$

Терм этого типа

$$\lambda y^{(\lambda\alpha.\,\sigma)\tau}.\,\Lambda\beta.\,\lambda f^{\forall\alpha.\,\sigma\to\beta}.\,f\,\tau\,y$$

Построение терма

$$\begin{array}{rcl} \beta{:}*, f{:}\forall\alpha.\,\sigma\to\beta & \vdash & f\,\tau : \sigma[\alpha := \tau] \to \beta \\ y{:}\sigma[\alpha := \tau], \beta{:}*, f{:}\forall\alpha.\,\sigma\to\beta & \vdash & f\,\tau\,y : \beta \end{array}$$

y — терм-свидетельство населенности σ при $\alpha := \tau$.

Удаление $\exists$. Тип

$$(\exists\alpha.\,\sigma)\to(\forall\alpha.\,\sigma\to\rho)\to\rho = (\forall\beta.\,(\forall\alpha.\,\sigma\to\beta)\to\beta)\to(\forall\alpha.\,\sigma\to\rho)\to\rho$$

Терм этого типа

$$\lambda\, f^{\exists\alpha.\,\sigma}.\, g^{\forall\alpha.\,\sigma\to\rho}.\, f\,\rho\, g$$

Построение терма

$$f{:}\exists\alpha.\,\sigma \quad\vdash\quad f\,\rho : (\forall\alpha.\,\sigma\to\rho)\to\rho$$

Фактически функция g строится пользователем как нужное ему вычисление $M : \rho$ с использованием публичного интерфейса $\langle\alpha, x^{\sigma[\alpha]}\rangle$ экзистенциального пакета:

$$g = \Lambda\alpha.\,\lambda x^{\sigma}.\, M$$