

# Введение в Теорию Типов

## Конспект лекций

Штукенберг Д. Г.  
Университет ИТМО

14 декабря 2018 г.

## 1 Введение

Эти лекции были рассказаны студентам групп М3334–М3337, М3339 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 5

### Изоморфизм Карри-Ховарда (завершение), Унификация

#### 2.1 Изоморфизм Карри-Ховарда

**Определение 2.1.** Изоморфизм Карри-Ховарда

1.  $\Gamma \vdash M : \sigma$  влечет  $|\Gamma| \vdash \sigma$  т.е.  $|\{x_1 : \Theta_1 \dots x_n : \Theta_n\}| = \{\Theta_1 \dots \Theta_n\}$
2. Если  $\Gamma \vdash \sigma$ , то существует  $M$  и существует  $\Delta$ , такое что  $|\Delta| = \Gamma$ , что  $\Delta \vdash M : \sigma$ , где  $\Delta = \{x_\sigma : \sigma \mid \sigma \in \Gamma\}$

**Пример.**  $\{f : \alpha \rightarrow \beta, x : \beta\} \vdash f x : \beta$

Применив , изоморфизм Карри-Ховарда получим:  $\{\alpha \rightarrow \beta, \beta\} \vdash \beta$

*Доказательство.* П.1 доказывается индукцией по длине выражения

$$1. \Gamma, x : \Theta \vdash x : \Theta \quad \Rightarrow_{KH} \quad |\Gamma|, \Theta \vdash \Theta$$

2.

$$\frac{\Gamma, x : \tau_1 \vdash P : \tau_2}{\Gamma \vdash \lambda x. P : \tau_1 \rightarrow \tau_2} \quad \Rightarrow_{KH} \quad \frac{|\Gamma|, \tau_1 \vdash \tau_2}{|\Gamma| \vdash \tau_1 \rightarrow \tau_2}$$

3.

$$\frac{\Gamma \vdash P : \tau_1 \rightarrow \tau_2 \quad \Gamma \vdash Q : \tau_1}{\Gamma \vdash P Q : \tau_2} \quad \Rightarrow_{KH} \quad \frac{|\Gamma| \vdash \tau_1 \rightarrow \tau_2 \quad |\Gamma| \vdash \tau_1}{|\Gamma| \vdash \tau_2}$$

П.2 доказывается аналогичным способом но действия обратные.

Т.е. отношения между типами в системе типов могут рассматриваться как образ отношений между высказываниями в логической системе, и наоборот.  $\square$

**Определение 2.2.** Расширенный полином:

$$E(p, q) = \begin{cases} C, & \text{if } p = q = 0 \\ p_1(p), & \text{if } q = 0 \\ p_2(q), & \text{if } p = 0 \\ p_3(p, q), & \text{if } p, q \neq 0 \end{cases}$$

, где  $C$  — константа,  $p_1, p_2, p_3$  — выражения, составленные из  $*$ ,  $+$ ,  $p, q$  и констант.

Пусть  $v = (\alpha \rightarrow \alpha) \rightarrow (\alpha \rightarrow \alpha)$ , где  $\alpha$  — произвольный тип и пусть  $F \in \Lambda$ , что  $F : v \rightarrow v \rightarrow v$ , то существует расширенный полином  $E$ , такой что  $\forall a, b \in \mathbb{N} F(\bar{a}, \bar{b}) =_{\beta} \overline{E(a, b)}$ , где  $\bar{a}$  — черчевский нумерал.

**Утверждение 2.1.** Типы черчевских нумералов

1.  $0 : \lambda f \lambda y. x : a \rightarrow b \rightarrow a$
2.  $1 : \lambda f \lambda y. f x : (a \rightarrow b) \rightarrow a \rightarrow b$
3.  $2 : \lambda f \lambda y. f(f x) : (a \rightarrow a) \rightarrow a \rightarrow a$
4.  $\forall i, i \geq 2 \quad \lambda f \lambda y. f(\dots(f x)) : (a \rightarrow a) \rightarrow a \rightarrow a$

*Доказательство.* Пункты 1 — 2 — очевидно. Рассмотрим более подробно пункт 4:

Разберем нумерал и рассмотрим два последних шага —

$$\frac{\dots}{\lambda f \lambda x. f(\dots(f x))} \quad \frac{\frac{f : a \rightarrow b \vdash f x : b}{f : a \rightarrow b \vdash f(f x) : \perp}}{\dots} \quad \frac{f : a \rightarrow b \vdash x : a}{\dots} \quad \begin{matrix} \{1\} \\ \{2\} \\ \{3\} \end{matrix}$$

на шаге 3 становится понятно, что  $f : a \rightarrow a$  и  $x : a$   $\square$

**Теорема 2.1.** У каждого терма в просто типизируемом  $\lambda$  исчислении существует расширенный полином.

**Утверждение 2.2.** Основные задачи типизации  $\lambda$  исчисления

1. *Проверка типа* — выполняется ли  $\Gamma \vdash M : \sigma$  для контекста  $\Gamma$  терма  $M$  и типа  $\sigma$  (для проверки типа обычно откидывают  $\sigma$  и рассматривают п.2).
2. *Реконструкция типа* — можно ли подставить вместо  $?$  и  $?_1$  в  $?_1 \vdash M : ?$  подставить конкретный тип  $\sigma$  в  $?$  и контекст  $\Gamma$  в  $?_1$ .
3. *Обитаемость типа* — пытается подобрать, такой терм  $M$  и контекст  $\Gamma$ , что бы было выполнено  $\Gamma \vdash M : \sigma$ .

**Определение 2.3.** Алгебраический терм Выражение типа

$$\Theta ::= a \mid (f_k \Theta_1 \dots \Theta_n)$$

где  $a$  — переменная,  $(f_k \Theta_1 \dots \Theta_n)$  — применение функции

## 2.2 Уравнение в алгебраических термах $\Theta_1 = \Theta_2$

### Система уравнений в алгебраических термах

**Определение 2.4.** Система уравнений в алгебраических термах

$$\begin{cases} \Theta_1 = \sigma_1 \\ \vdots \\ \Theta_n = \sigma_n \end{cases}$$

где  $\Theta_i$  и  $\sigma_i$  — термы

**Определение 2.5.**  $\{a_i\} = A$  — множество переменных,  $\{\Theta_i\} = T$  — множество термов.

**Определение 2.6.** Подстановка — отображение вида:  $S_0 : A \rightarrow T$ , которое является решением в алгебраических термах.

$S_0(a)$  может быть либо  $S_0(a) = \Theta_i$ , либо  $S_0(a) = a$ .

Доопределим  $S$  на все  $T$  т.е.  $S : T \rightarrow T$ , где

1.  $S(a) = S_0(a)$
2.  $S(f(\Theta_1 \cdots \Theta_k)) = f(S(\Theta_1) \cdots S(\Theta_k))$

$S$  то же самое что и много *if'*ов либо *map* строк.

**Определение 2.7.** Решить уравнение в алгебраических термах — найти такое  $S$ , что  $S(\Theta_1) = S(\Theta_2)$

**Пример.**

Заранее обозначим:  $a, b$  — переменные  $f, g, h$  — функции

1.  $f(a(gb)) = f(he)d$  имеет решение  $S(a) = he$  и  $S(d) = gb$

$$(a) \quad S(fa(gb)) = f(he)(gb)$$

$$(b) \quad S(f(he)d) = f(he)(gb)$$

$$(c) \quad f(he)(gb) = f(he)(gb)$$

2.  $fa = gb$  — решений не имеет

Таким образом, что бы существовало решение необходимо равенство строк полученной подстановки.

## 2.3 Алгоритм Унификации. Определения

1. Система уравнений  $E_1$  эквивалентна  $E_2$ , если они имеют одинаковые решения (унификаторы).
2. Любая система  $E$  эквивалентна некоторому уравнению  $\Sigma_1 = \Sigma_2$ .

*Доказательство.* Возьмем функциональный символ  $f$ , не использующийся в  $E$ ,

$$E = \begin{cases} \Theta_1 = \sigma_1 \\ \vdots \\ \Theta_n = \sigma_n \end{cases}$$

это же уравнение можно записать как —  $f \Theta_1 \dots \Theta_n = f \sigma_1 \dots \sigma_n$

Если существует подстановка  $S$  такая, что

$$S(\Theta_i) = S(\sigma_i) \quad \forall i, \text{ то } S(f \Theta_1 \dots \Theta_n) = f S(\sigma_1) \dots S(\sigma_n)$$

Обратное аналогично. □

### 3. Рассмотрим операции

(a) *Редукция терма*

Заменяем уравнение вида  $f_1 \Theta_1 \dots \Theta_n = f_1 \sigma_1 \dots \sigma_n$  на систему уравнений

$$\Theta_1 = \sigma_1$$

$\vdots$

$$\Theta_n = \sigma_n$$

(b) *Устранение переменной*

Пусть есть уравнение  $x = \Theta$ , заменим во всех остальных уравнениях переменную  $x$  на терм  $\Theta$ .

**Утверждение 2.3.** Эти операции не изменяют множества решений.

**Определение 2.8.** Система уравнений в разрешенной форме если

1. Все уравнения имеют вид  $a_i = \Theta_i$
2. Каждый из  $a_i$  входит в систему уравнений только раз

**Определение 2.9.** Система несовместима если

1. существует уравнение вида  $f \Theta_1 \dots \Theta_n = g \sigma_1 \dots \sigma_n$ , где  $f \neq g$
2. существует уравнение вида  $a = f \Theta_1 \dots \Theta_n$ , причем  $a$  выходит в какой-то из  $\Theta_i$

## 2.4 Алгоритм унификации

1. Пройдемся по системе, выберем такое уравнение, что оно удовлетворяет одному из условий:
  - (a) Если  $\Theta_i = a_i$ , то перепишем, как  $a_i = \Theta_i$ ,  $\Theta_i$  — не переменная
  - (b)  $a_i = a_i$  — удалим
  - (c)  $f \Theta_1 \dots \Theta_n = f \sigma_1 \dots \sigma_n$  — применим редукцию термов
  - (d)  $a_i = \Theta_i$  — Применим подстановку переменной — подставим во все остальные уравнения  $\Theta_i$  вместо  $a_i$
2. Проверим разрешима ли система, совместима ли система (два пункта несовместимости)
3. повторим пункт 1

**Утверждение 2.4.** Алгоритм не изменяет множества решений

**Утверждение 2.5.** Несовместная система не имеет решений

**Утверждение 2.6.** Система в разрешенной форме имеет решение:

$$\left\{ \begin{array}{l} a_1 = \Theta_1 \\ \vdots \\ a_n = \Theta_n \end{array} \right. \text{ имеет решение } - \left\{ \begin{array}{l} S_0(a_1) = \Theta_1 \\ \vdots \\ S_0(a_n) = \Theta_n \end{array} \right.$$

**Утверждение 2.7.** Алгоритм всегда закрывается

*Доказательство.* По индукции, выберем три числа  $\langle x y z \rangle$ , где

$x$ —количество переменных, которые встречаются строго больше одного раза в левой части некоторого уравнения ( $b$  не влияет на  $x$ , а  $a$  влияет в уравнении  $f(a(ga)b) = \Theta$ ),

$y$ — количество функциональных символов в системе,

$z$ —количество уравнений типа  $a = a$  и  $\Theta = b$ .

Определим отношение  $\leq$  между двумя кортежами, как  $\langle x_1 y_1 z_1 \rangle \leq \langle x_2 y_2 z_2 \rangle$  если верно одно из следующих условий:

1.  $x_1 < x_2$
2.  $x_1 = x_2 \ \& \ y_1 < y_2$
3.  $x_1 = x_2 \ \& \ y_1 = y_2 \ \& \ z_1 < z_2$

Заметим, что операции (a) и (b) всегда уменьшают  $z$  и иногда уменьшают  $x$ .

Операция (c) всегда уменьшает  $y$  иногда  $x$  и, возможно, увеличивает  $z$ .

Операция (d) всегда уменьшает  $x$ , и иногда увеличивает  $y$ .

Очевидно, что с каждой операцией  $a - d$  данная тройка уменьшается и так как  $x, y, z \geq 0$ , то данный алгоритм завершится за конечное время.  $\square$

**Пример.**

Исходная система

$$E = \left\{ \begin{array}{l} g(x_2) = x_1 \\ f(x_1, h(x_1), x_2) = f(g(x_3), x_4, x_3) \end{array} \right\}$$

Применим пункт (c) ко второму уравнению верхней системы получим:

$$E = \left\{ \begin{array}{l} g(x_2) = x_1 \\ x_1 = g(x_3) \\ h(x_1) = x_4 \\ x_2 = x_3 \end{array} \right\}$$

Применим пункт (d) ко второму уравнению верхней системы (оно изменит 1ое уравнение) получим:

$$E = \left\{ \begin{array}{l} g(x_2) = g(x_3) \\ x_1 = g(x_3) \\ h(g(x_3)) = x_4 \\ x_2 = x_3 \end{array} \right\}$$

Применим пункт (c) ко первому уравнению и пункт (a) к третьему уравнению верхней системы

$$E = \left\{ \begin{array}{l} x_2 = x_3 \\ x_1 = g(x_3) \\ x_4 = h(g(x_3)) \\ x_2 = x_3 \end{array} \right\}$$

Применим пункт (b) к последнему уравнению и получим систему в разрешенной форме

$$E = \left\{ \begin{array}{l} x_2 = x_3 \\ x_1 = g(x_3) \\ x_4 = h(g(x_3)) \end{array} \right\}$$

Решение системы:

$$S = \left\{ \begin{array}{l} (x_1 = g(x_3)) \\ (x_2 = x_3) \\ (x_4 = h(g(x_3))) \end{array} \right\}$$

**Определение 2.10.**  $S \circ T$ —композиция подстановок, если  $S \circ T = S(T(a))$

**Определение 2.11.**  $S$ —наиболее общий унификатор, если любое решение системы  $R$  может быть получено уточнением:  $\exists T : R = T \circ S$

**Утверждение 2.8.** Алгоритм дает наиболее общий унификатор системы, если у нее есть решения. Если решений нет алгоритм окончится неудачей.

### 3 Лекция 6

## Реконструкция типов в просто типизированном лямбда-исчислении, комбинаторы

### 3.1 Алгоритм вывода типов

Пусть есть:  $? \vdash A : ?$ , хотим найти пару  $\langle \text{контекст, тип} \rangle$

**Алгоритм:**

1. Рекурсия по структуре формулы

Построить по формуле  $A$  пару  $\langle E, \tau \rangle$ , где

$E$ —набор уравнений,  $\tau$ —тип  $A$

2. Решение уравнения, получения подстановки  $S$  и из решения  $E$  и  $S(\tau)$  получение ответа

Т.е. необходимо свести вывод типа к алгоритму унификации.

**Пункт 3.1.** Рассмотрим 3 случая

**Обозначение**  $\rightarrow$  — алгебраический тип

1.  $A \equiv x \implies \langle \{\}, \alpha_A \rangle$ , где  $\{\}$ —пустой контекст,  $\alpha_A$ —новая переменная нигде не встречавшаяся до этого в формуле
2.  $A \equiv P Q \implies \langle E_P \cup E_Q \cup \{\tau_P = \rightarrow (\tau_Q \alpha_A)\}, \alpha_A \rangle$ , где  $\alpha_A$ —новая переменная
3.  $A \equiv \lambda x. P \implies \langle E_P, \alpha_x \rightarrow \tau_P \rangle$

**Пункт 3.2.** Алгоритм унификации

Рассмотрим  $E$ —набор уравнений, запишем все уравнения в алгебраическом виде т.е.

$\alpha \rightarrow \beta \Leftrightarrow \rightarrow \alpha \beta$ , затем применяем алгоритм унификации.

**Лемма 3.1.** Рассмотрим терм  $M$  и пару  $\langle E_M, \tau_M \rangle$ , Если  $\Gamma \vdash M : \rho$ , то существует:

1.  $S$ -решение  $E_M$  тогда  $\Gamma = \{x : S(\alpha_x) \mid x \in FV(M)\}$ ,  $FV$ —множество свободных переменных в терме  $M$ ,  $\alpha_x$ — переменная полученная при разборе терма  $M$

$$\rho = S(\tau_M)$$

2. Если  $S$ — решение  $E_M$ , то  $\Gamma \vdash M : \rho$ ,

*Доказательство.* индукция по структуре терма  $M$

- (a) Если  $M \equiv x$ , то так как решение существует, то существует и  $S(\alpha_x)$ , что:

$$\Gamma, x : S(\alpha_x) \vdash x : S(\alpha_x)$$

- (b) Если  $M \equiv \lambda x. P$ , то по индукции уже известен тип  $P$ , контекст  $\Gamma$  и тип  $x$ , тогда:

$$\frac{\Gamma, x : S(\alpha_x) \vdash P : S(\alpha_P)}{\Gamma \vdash \lambda x. P : S(\alpha_x) \rightarrow S(\alpha_P)}$$

- (c) Если  $M \equiv P Q$ , то по индукции:

$$\frac{\Gamma \vdash P : S(\alpha_P) \equiv \tau_1 \rightarrow \tau_2 \quad \Gamma \vdash Q : S(\alpha_Q) \equiv \tau_1}{\Gamma \vdash P Q : \tau_2}$$

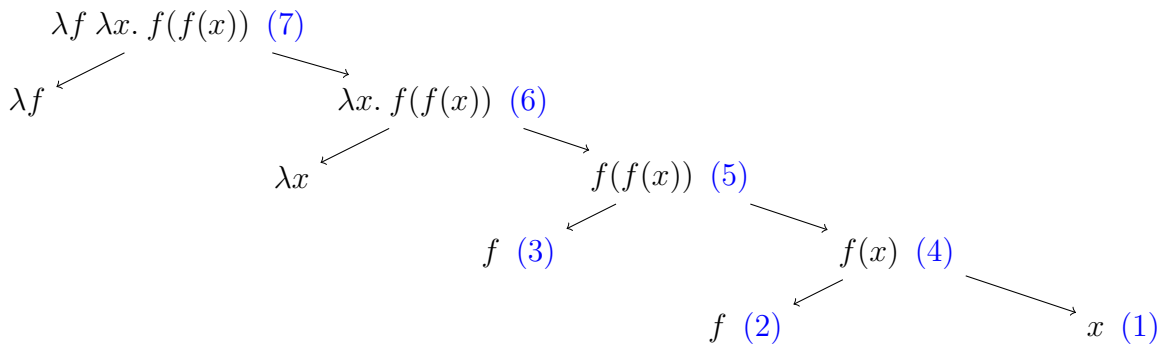
□

$\langle \Gamma, \rho \rangle$  — основная пара для терма  $M$ , если

1.  $\Gamma \vdash M : \tau$
2. Если  $\Gamma' \vdash M : \tau'$ , то существует  $S : S(\Gamma) \subset \Gamma'$

**Пример.**

Рассмотрим терм:  $\lambda f \lambda x. f(f(x))$ , построим и пронумеруем его дерево разбора:



1.  $E_1 = \langle \{\}, \alpha_x \rangle$
2.  $E_2 = \langle \{\}, \alpha_f \rangle$
3.  $E_3 = \langle \{\}, \alpha_f \rangle$
4.  $E_4 = \langle \{\alpha_f = \rightarrow (\alpha_x \alpha_1)\}, \alpha_1 \rangle$
5.  $E_5 = \langle \left\{ \begin{array}{l} \alpha_f = \rightarrow (\alpha_x \alpha_1) \\ \alpha_f = \rightarrow (\alpha_1 \alpha_2) \end{array} \right\}, \alpha_2 \rangle$
6.  $E_6 = \langle \left\{ \begin{array}{l} \alpha_f = \rightarrow (\alpha_x \alpha_1) \\ \alpha_f = \rightarrow (\alpha_1 \alpha_2) \end{array} \right\}, \alpha_x \rightarrow \alpha_2 \rangle$

$$7. E_7 = \langle \left\{ \begin{array}{l} \alpha_f = \rightarrow (\alpha_x \alpha_1) \\ \alpha_f = \rightarrow (\alpha_1 \alpha_2) \end{array} \right\}, \alpha_f \rightarrow (\alpha_x \rightarrow \alpha_2) \rangle$$

$$E = \left\{ \begin{array}{l} \alpha_f = \rightarrow (\alpha_x \alpha_1) \\ \alpha_f = \rightarrow (\alpha_1 \alpha_2) \end{array} \right\}, \text{ решим полученную систему:}$$

1. Решим сисетму:

(a)

$$\left\{ \begin{array}{l} \alpha_f = \rightarrow (\alpha_x \alpha_1) \\ \alpha_f = \rightarrow (\alpha_1 \alpha_2) \end{array} \right.$$

(b)

$$\left\{ \rightarrow (\alpha_1 \alpha_2) = \rightarrow (\alpha_x \alpha_1) \right.$$

(c)

$$\left\{ \begin{array}{l} \alpha_1 = \alpha_x \\ \alpha_2 = \alpha_1 \end{array} \right.$$

(d)

$$\left\{ \begin{array}{l} \alpha_1 = \alpha_x \\ \alpha_2 = \alpha_x \end{array} \right.$$

2. Получим

$$S = \left\{ \begin{array}{l} \alpha_f = \rightarrow (\alpha_x \alpha_1) \\ \alpha_1 = \alpha_x \\ \alpha_2 = \alpha_x \end{array} \right.$$

3.  $\Gamma = \{\}$ , так как в заданной формуле нет свободных переменных

4. тип терма  $\lambda f \lambda x. f(f(x))$  является результат подстановки

$$S(\rightarrow \alpha_f (\alpha_x \rightarrow \alpha_2)), \text{ получаем } \tau = (\alpha_x \rightarrow \alpha_x) \rightarrow (\alpha_x \rightarrow \alpha_x)$$

## 3.2 Сильная и слабая нормализации

**Определение 3.1.** Если существует последовательность редукций, приводящая терм  $M$  в нормальную форму, то  $M$ —слабо нормализуем. (Т.е. при редуцировании терма  $M$  мы можем не прийти в н.ф.)

**Определение 3.2.** Если не существует бесконечной последовательности редукций терма  $M$ , то терм  $M$ — сильно нормализуем.

**Утверждение 3.1.**

1.  $KI\Omega$ — слабо нормализуема

**Пример.**

Перепишем  $KI\Omega$  как  $((\lambda x \lambda y. x)(\lambda x. x))(((\lambda x. x x)(\lambda x. x x)))$ , очевидно, что этот терм можно редуцировать двумя разными способами:

(a) Сначала редуцируем красную скобку



- i.  $((\lambda x \lambda y. x)(\lambda x. x))(((\lambda x. x x)(\lambda x. x x)))$
- ii.  $((\lambda y. (\lambda x. x))(((\lambda x. x x)(\lambda x. x x))))$
- iii.  $(\lambda x. x)$

Видно, что в этом случае количество шагов конечно.

- (b) Редуцируем синюю скобку. Очевидно, что комбинатор  $\Omega$  не имеет нормальной формы, тогда понятно, что в этом случае терм  $KI\Omega$  никогда не редуцируется в нормальную форму.

2.  $\Omega$  — не нормализуема

3.  $II$  — сильно нормализуема

**Лемма 3.2.** Сильная нормализация влечет слабую.

### 3.3 Выразимость комбинаторов

**Утверждение 3.2.** Любое  $\lambda$  выражение можно записать с помощью комбинаторов  $S$  и  $K$ , где

$$S = \lambda x \lambda y \lambda z. (x z) (y z) : (a \rightarrow b \rightarrow c) \rightarrow (a \rightarrow b) \rightarrow a \rightarrow c$$

$$K = \lambda x \lambda y. x : a \rightarrow b \rightarrow a$$

**Утверждение 3.3.** Комбинаторы  $S$  и  $K$  являются аксиомами в ИИВ

**Утверждение 3.4.** Соотношение комбинаторов с  $\lambda$  исчислением:

- 1.  $T(x) = x$
- 2.  $T(P Q) = T(P) T(Q)$
- 3.  $T(\lambda x. P) = K(T(P)), \quad x \notin FV(P)$
- 4.  $T(\lambda x. x) = I$
- 5.  $T(\lambda x \lambda y. P) = T(\lambda x. T(\lambda y. P))$
- 6.  $T(\lambda x. P Q) = S T(\lambda x. P) T(\lambda x. Q)$

**Утверждение 3.5.** Связь комбинаторов с ИИВ

**Утверждение 3.6.** Альтернативный базис:

- 1.  $B = \lambda x \lambda y \lambda z. x (y z) : (a \rightarrow b) \rightarrow (c \rightarrow a) \rightarrow c \rightarrow b$
- 2.  $C = \lambda x \lambda y \lambda z. ((x z) y) : (a \rightarrow b \rightarrow c) \rightarrow b \rightarrow a \rightarrow c$
- 3.  $W = \lambda x \lambda y. ((x y) y) : (a \rightarrow a \rightarrow b) \rightarrow a \rightarrow b$