

# Лекция 5

## Изоморфизм Карри-Ховарда (завершение), Унификация

### Определение

Изоморфизм Карри-Ховарда

1.  $\Gamma \vdash M:\sigma$  влечет  $|\Gamma| \vdash \sigma$
2.  $\Gamma \vdash \sigma$ , то существует  $M$  и существует  $\Delta$ , такое что  $|\Delta| = \Gamma$ , что  $\Delta \vdash M:\sigma$ , где  $\Delta = \{x_\sigma:\sigma \mid \sigma \in \Gamma\}$

Рассмотрим пример:  $\{f:\alpha \rightarrow \beta, x:\beta\} \vdash fx:\beta$

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

П.1 доказывается индукцией по длине выражения т.е. есть 3 правила вывода. убирая Р и Q.

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.

### Определение

расширенный полином определяется формулой:

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

, где  $C$ —константа,  $p_1, p_2, p_3$ —выражения, составленные из  $*$ ,  $+$ ,  $p, q$  и констант по сути расширенный полином это множество функций над натуральными числами (черчевскими нумералами).

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

### Теорема

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

### 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 **замкнутый** терм  $M$  и контекст  $\Gamma$ , что бы было выполнено  $\Gamma \vdash M : \sigma$ .

Определение **Алгебраический терм**—выражение типа  $\Theta = a|(f_k \Theta_1 \cdots \Theta_n)$ , где  $a$ —переменная,  $(f_k \Theta_1 \cdots \Theta_n)$ —применение функции

Примеры:

1.  $(fab(ga))$
2. Известно, что  $\rightarrow$ —функция, тогда выражение  $((a \rightarrow b) \rightarrow c) \iff (\rightarrow (\rightarrow ab)c)$

**Уравнение в алгебраических термах**  $\Theta_1 = \Theta_2$

**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  $\{a_i\} = A$ —множество перменных,  $\{\Theta_i\} = T$ —множество термов.

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

Т.е.  $S_0$ —конечное множество переменных  $a_1 \cdots a_n$  на которых  $S_0(a_i) = \Theta_i$  либо  $S_0(a_i) = a_i$ .

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

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

По сути  $S$  тоже самое что и много  $if'$ ов либо  $тар$  строк

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

Например:

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

### Алгоритм Унификации

1. Система уравнений  $E_1$  эквивалентна  $E_2$ , если они имеют одинаковые решения(унификаторы).

2. Любая система  $E$  эквивалентна некторому 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 сущесвтует подстановка  $S$  такая, что

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

Обратное аналогично.

3. Рассмотрим операции

(a) *Редукция терма*

Заменим 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** — эти операции не изменяют множества решений.

**Определение:** 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:** 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  $\Theta_i$

**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 т.е. 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:** алгоритм не изменяет множества решений

**Утверждение:** несовместимая система не имеет решений

**Утверждение:** 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:** алгоритм всегда заканчивается

**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 (т.е.  $b$  не повлияет на  $x$ , а  $a$  повлияет в уравнении  $f(a(g \ a) \ b) = \Theta$ ),

$y$  — количество функциональных символов в системе,

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

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

(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.

**Пример**

Исходная система

$$E =$$

$$g(x_2) = x_1$$

$$f(x_1, h(x_1), x_2) = f(g(x_3), x_4, x_3)$$

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

$$E =$$

$$g(x_2) = x_1$$

$$x_1 = g(x_3)$$

$$h(x_1) = x_4$$

$$x_2 = x_3$$

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

$$\begin{aligned} E = \\ g(x_2) &= g(x_3) \\ x_1 &= g(x_3) \\ h(g(x_3)) &= x_4 \\ x_2 &= x_3 \end{aligned}$$

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

$$\begin{aligned} E = \\ x_2 &= x_3 \\ x_1 &= g(x_3) \\ x_4 &= h(g(x_3)) \\ x_2 &= x_3 \end{aligned}$$

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

$$\begin{aligned} E = \\ x_2 &= x_3 \\ x_1 &= g(x_3) \\ x_4 &= h(g(x_3)) \end{aligned}$$

Решение системы:

$$\begin{aligned} S = \{ \\ (x_1 = g(x_3)), \\ (x_2 = x_3), \\ (x_4 = h(g(x_3)))) \\ \} \end{aligned}$$

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

**Определение:**  $S$ —наиболее общий унификатор ксли любое решение сисетмы  $R$  может быть получено уточнением:

$$\exists T : R = T \circ S$$

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

#### **Ссылки**

1. <https://www.quora.com/What-is-an-intuitive-explanation-of-the-Curry-Howard-correspondence>
2. <https://habr.com/post/269907/>
3. <https://arxiv.org/pdf/cs/0701022.pdf>
4. <http://moscova.inria.fr/levy/courses/X/IF/03/pi/levy2/martelli-montanari.pdf>