

Теория типов: Конспект

Шубин Владислав

23 января 2025 г.

Оглавление

1	Введение	3
1.1	Простая теория типов	4
1.1.1	Аксиомы	4

Глава 1

Введение

Теория типов представляет собой новое направление в логике, изучающее системы типов. С математической точки зрения существует два пути происхождения теории типов. Первый из них есть анализ математического текста, например из книги Шафаревича [1] мы видим следующее «для $y \in Y$ и $x \in f^{-1}(y)$ мы получаем уравнение

$$t_i(x)^k + a_1(y)t_i(x)^{k-1} + \dots + a_k(y) = 0.»$$

В этом контексте мы хотим анализировать подобные высказывания, которые понимаются, обычно, интуитивно. Например, здесь бессмысленно было бы сказать $x \in \mathbb{T}$, по причине того, что x — это точка аффинной схемы, но не элемент поля. В этом смысле они обладают разными «типами». Человек, знакомый с алгебраической геометрией, понимает, что $k \in \mathbb{N}$, $t_i : X \rightarrow \mathbb{T}$, $a_i : Y \rightarrow \mathbb{T}$, где \mathbb{T} есть поле.

Подобные выражения недоступны в языке логики первого порядка и для того, чтобы оперировать с такой «математической грамматикой» нам понадобится теория типов

В теории типов мы, подобно символу \in , используем символ $:$, означающий «иметь тип». Например, $x : A$ означает « x имеет тип A ». Пусть у нас есть несколько типов $x : A, y : B$, которые мы отделяем запятой. Если из них возможен вывод, мы его будем называть суждением, а посылку контекстом. То есть

$$\underbrace{x : A, y : B}_{\text{контекст}} \vdash \underbrace{t : C}_{\text{суждение}} .$$

Кроме чисто математической точки зрения мы имеем довольно важную практическую сторону этой теории — это языки программирования. С точки зрения компьютера (если у него, конечно, может быть какая-либо точка зрения) любые данные представляют из себя кусок бинарного кода. Картинки, программы, музыка — все это есть лишь последовательность нулей и единиц. Поэтому в ранних языках программирования существовала проблема типизации. Для примера возьмем Си:

```
void rev(char *str, int len){
    int start = 0;
    int end = len - 1;
    while(start < end) {
        char tmp = str[start];
        str[start] = str[end];
        str[end] = tmp;
        end--;
        start++;
    }
}
```

Современный язык Си использует систему типов, в которой тип `char` представляет символьный тип. Однако, нет никакой существенной разницы между типами `char` и `short`, например (второй

тип представляет однобайтовое число). Поэтому эту функцию можно применить и к массиву чисел, что создает существенные проблемы при разработке сложного ПО.

Из-за этой проблемы сформировалось, своего рода, ad hoc решение, называемое строгой типизацией.

Замечание 1.0.1. В математике любой объект можно представить как некоторое множество, однако это не лишает теорию типов приложимости. Нам важно, что натуральное число — это не просто множество, а что-то новое, обладающее свойствами «числа».

1.1 Простая теория типов

Введем нашу первую модель теории типов формально. Типы мы определим как

$$A := 1 \mid A_1 \times A_2 \mid \text{Nat}$$

, то есть свободно порожденное множество с элементами $1 \in \text{Ty}$, $\text{Nat} \in \text{Ty}$, $\times : \text{Ty}^2 \rightarrow \text{Ty}$. Под 1 мы имеем ввиду унарный тип. И, как и во всякой теории, нам понадобятся термы

$$t := x \mid \langle \rangle \mid \langle t_1, t_2 \rangle \mid p_1(t) \mid p_2(t) \mid 0 \mid S(t) \mid \text{rec}(\dots),$$

где rec — рекурсия, чьи аргументы мы уточним позже.

Далее нам понадобится контекст, определяемый как $\Gamma := \langle \rangle \mid \Gamma, x : A$, то есть контексты суть конечные списки переменных с типами (например, $x_1 : A_1, \dots, x_n : A_n$).

Теперь займемся суждением о типе — отношением $\Gamma \vdash t : A$ или «в контексте Γ переменная t имеет тип A » что аналогично «выводимости» в логике первого порядка.

1.1.1 Аксиомы

Первая часть правил будет относиться к так называемым структурным правилам. Под i мы будем иметь ввиду любой тип в контексте.

Перечислим их:

- (Axiom, Identity, Assumption)

$$\frac{}{x_1 : A_1, \dots, x_n : A_n \vdash x_i : A_i}$$

- Подстановка

$$\frac{\Delta \vdash s_i : A_i \quad \Gamma \vdash t : C}{\Delta \vdash t[s_i/x_i] : C}$$

Аксиому подстановки можно ослабить следующими способами:

- Ослабление:

$$\frac{\Gamma \vdash t : A}{\Gamma, x : B \vdash t : A}$$

- Замена:

$$\frac{\Delta \vdash s_i : A_i \quad \Gamma \vdash t : C}{\Gamma, x : A, y : B \vdash t : C}$$

- Замена одной переменной:

$$\frac{\Gamma, x : A \vdash t : B \quad \Gamma \vdash a : A}{\Gamma \vdash t[a/x] : B}$$

В некоторых версиях системы типов мы можем использовать один из этих вариантов.

Пример 1. Пример работы аксиомы подстановки

$$\frac{y : \mathbb{N} \vdash y \cdot y : \mathbb{N} \quad x_1, x_2 : \mathbb{N} \vdash x_1 + x_2 : \mathbb{N}}{x_1, x_2 : \mathbb{N} \vdash x_1 \cdot x_1 + x_2 \cdot x_2 : \mathbb{N}}$$

Наша система все еще не является теорией типов, поскольку у нас нет правил вывода типов. Давайте их введем:

- Произведение

$$\frac{\Gamma \vdash t_1 : A_1, \Gamma \vdash t_2 : A_2}{\Gamma \vdash \langle t_1, t_2 \rangle : A_1 \times A_2}$$

- Проекция

$$\frac{\Gamma \vdash t : A_1 \times A_2}{\Gamma \vdash p_i(t) : A_i}$$

- 0

$$\overline{\Gamma \vdash 0 : \text{Nat}}$$

- $S(n)$

$$\frac{\Gamma \vdash n : \text{Nat}}{\Gamma \vdash S(n) : \text{Nat}}$$

- 1

$$\overline{\Gamma \vdash \langle \rangle : 1}$$

- rec

$$\frac{\Gamma \vdash n : \text{Nat} \quad \Gamma \vdash t_0 : C \quad \Gamma, x : \text{Nat}, y : C \vdash t_s(x, y) : C}{\Gamma \vdash \text{rec}(t_0; (x, y, t_s); n) : C}$$

Теперь мы ввели все, что нужно для теории типов. Однако, стоит учитывать один нюанс.

Аннотации

Давайте посмотрим внимательно на аксиому проекции

$$\frac{\Gamma \vdash t : A_1 \times A_2}{\Gamma \vdash p_i(t) : A_i}$$

Из этой аксиомы нельзя точно понять, на каком типе определена функция p_i , но мы видим этот тип из контекста. В теории типов существует аннотирование, которое приписывает на каком типе определена функция. С ней наша аксиома должна выглядеть как

$$\frac{\Gamma \vdash t : A_1 \times A_2}{\Gamma \vdash p_i^{A_1, A_2}(t) : A_i}$$

Аннотация присутствует всегда, но для читаемости ее не выписывают.

Упрощение выражений

Мы уже ввели теорию типов. Для нее не обязательны аксиомы для упрощения выражений, но давайте их введем для удобства

Литература

- [1] Шафаревич И.Р. *Основы алгебраической геометрии*. УМН, 24:6(150) (1969), 3–184; Russian Math. Surveys, 24:6 (1969), 1–178.
- [2] Robert Harper. *Type Systems for Programming Languages*. School of Computer Science, Carnegie Mellon University, Spring, 2000, url: <https://people.mpi-sws.org/~dreyer/ats/papers/harper-tspl.pdf>
- [3] Per Martin-Löf. *Intuitionistic Type Theory*. Bibliopolis. url: <https://archive-pml.github.io/martin-lof/pdfs/Bibliopolis-Book-retypeset-1984.pdf>
- [4] Bengt Nordström, Kent Petersson, Jan M. Smith. *Programming in Martin-Löf's Type Theory*. Department of Computing Sciences, University of Göteborg, Sweden.