Теория типов

О курсе

Краткое содержание вступительного занятия

1. История вопроса: что вообще изучает теория типов, типы в математике, типы в лямбда-исчислении. Краткое повторение материала, знание которого ожидается от участников.
2. Содержание текущего курса: изоморфизм Карри-Ховарда и его применение в программировании и математике
3. Особенности преподавания.

Типы в теории множеств

$x \in x$?

class X { X a; X b; }

$x =$ э. м.

- Парадокс Рассела: $\{\ x \mid x \notin x\ \}$. Законна ли эта запись? $a \in b$ — ожидаем, что слева элемент, а справа множество.
- Давайте запретим. Например, введём тип множества: $\varnothing^0$, $\{x^n, y^n, z^n\}^{n+1}$ и т.п.
- B. Russel, A. Whitehead, Ramified type theory, 1908 (разветвлённая теория типов). Все объекты получают тип и порядок. Формулы $m + 1$ порядка работают с объектами, задаваемыми формулами m порядка:

$$^{m+1}F^n : {}^mP^n \to {}^mQ^n$$

0 $\varnothing$

1 $\{\varnothing\}$

2 $\{\varnothing, \{\varnothing\}\}$

- По силе примерно Р.Т.Т. соответствует аксиоматике Цермело-Френкеля, но неудобна. В ZF можно приписать множествам схожую типу характеристику («ранг»), сложностью выражений можно управлять, например, средствами аксиомы конструктивности.

Лямбда-исчисление: история возникновения

- Готлоб Фреге, 1893 год, «карринг». Двуместную функцию $a + b$ можно представить как композицию двух одноместных функций:

$$f(a) = \lambda x.a + x \qquad a + b = f(a)(b)$$

- Моисей Шейнфинкель, 1924, комбинаторы:

$$Kab = a \qquad Sabc = ac(bc)$$

- Алонзо Чёрч, 1932, лямбда-исчисление:

$$(\lambda x.M) = M[x := N]$$

- Алонзо Чёрч, 1932, 1934: В λ-исчислении арифметика выражается естественно. Попробуем λ-исчисление расширить до логики?
- С.Клини и Б.Россер, 1935, противоречие (модификация парадокса Ришара).

$f(5)$ — "прибавь 5"

$f(5)\,4 = 9$

$f(5)\,7 = 12$

$g = f(5)$

$g(7) = 12$

Лямбда-исчисление как логический язык

«Анахроническое» изложение, пересказ современным языком: следуем J. Barkley Rosser: Highlights of the history of the Lambda-Calculus

- В лямбда-исчисление введём логический символ $\to$. Формулы исчисления будем считать логическими высказываниями. Добавим логические аксиомы. Ожидаем такое: $\vdash 0 + 1 = 1$
- Получение противоречия: определим минимальные требования к исчислению. Очевидно, хотя бы следующее мы должны уметь доказывать:
  1. $\vdash A \to A$
  2. $\vdash (A \to (A \to B)) \to (A \to B)$
  3. Если $\vdash A$ и $\vdash A \to B$, то $\vdash B$.
- Менее очевидное: $\vdash A \to B$, если $A =_\beta B$. Мотивация: если $0 + 1 =_\beta 1$, то $X(0 + 1)$ всегда можно заменить на $X(1)$ (равенство по Лейбницу).

Лямбда-исчисление как логический язык

«Анахроническое» изложение, пересказ современным языком: следуем J. Barkley Rosser: Highlights of the history of the Lambda-Calculus

- В лямбда-исчисление введём логический символ $\to$. Формулы исчисления будем считать логическими высказываниями. Добавим логические аксиомы. Ожидаем такое: $\vdash 0 + 1 = 1$
- Получение противоречия: определим минимальные требования к исчислению. Очевидно, хотя бы следующее мы должны уметь доказывать:
  1. $\vdash A \to A$
  2. $\vdash (A \to (A \to B)) \to (A \to B)$
  3. Если $\vdash A$ и $\vdash A \to B$, то $\vdash B$.
- Менее очевидное: $\vdash A \to B$, если $A =_\beta B$. Мотивация: если $0 + 1 =_\beta 1$, то $X(0+1)$ всегда можно заменить на $X(1)$ (равенство по Лейбницу). $1 = 0.5 \cdot 2$ влечёт $\sin 1 = \sin(0.5 \cdot 2)$, а как иначе?

Лямбда-исчисление как логический язык

«Анахроническое» изложение, пересказ современным языком: следуем J. Barkley Rosser: Highlights of the history of the Lambda-Calculus

- В лямбда-исчисление введём логический символ $\rightarrow$. Формулы исчисления будем считать логическими высказываниями. Добавим логические аксиомы. Ожидаем такое: $\vdash 0 + 1 = 1$
- Получение противоречия: определим минимальные требования к исчислению. Очевидно, хотя бы следующее мы должны уметь доказывать:
  (сх.1) 1. $\vdash A \rightarrow A$
  (сх2) 2. $\vdash (A \rightarrow (A \rightarrow B)) \rightarrow (A \rightarrow B)$
  MP 3. Если $\vdash A$ и $\vdash A \rightarrow B$, то $\vdash B$. д.
- Менее очевидное: $\vdash A \rightarrow B$, если $A =_\beta B$. Мотивация: если $0 + 1 =_\beta 1$, то $X(0 + 1)$ всегда можно заменить на $X(1)$ (равенство по Лейбницу). $1 = 0.5 \cdot 2$ влечёт $\sin 1 = \sin(0.5 \cdot 2)$, а как иначе? $(\lambda x.\ x=1)^{0+1} \rightarrow (\lambda x.x=1)'$
- Заметим: $0 + 1 \twoheadrightarrow_\beta 1$, поэтому $\vdash (1 = 1) \rightarrow (0 + 1 = 1)$.

Парадокс Карри

$\Phi_\alpha := Y\ (\lambda x.x \to \alpha)$ - запрет?

$(Y(\lambda x.x \to \alpha)) =_\beta \Phi_\alpha \to \alpha$

Редуцируя Φ_α, получаем:

$$\Phi_\alpha \twoheadrightarrow_\beta (\lambda x.x \to \alpha)\ (Y\ (\lambda x.x \to \alpha)) \twoheadrightarrow_\beta \Phi_\alpha \to \alpha$$

И доказательство:

$\vdash$	1) $\Phi_\alpha \to (\Phi_\alpha \to \alpha)$	$(A \to A)$ и $\Phi_\alpha =_\beta \Phi_\alpha \to \alpha$	(сх. 1) $+\beta$
	2) $(\Phi_\alpha \to \Phi_\alpha \to \alpha) \to (\Phi_\alpha \to \alpha)$	Так как $(A \to (A \to B)) \to (A \to B)$	(сх. 2)
	3) $\Phi_\alpha \to \alpha$	MP 1, 2	
	4) $(\Phi_\alpha \to \alpha) \to \Phi_\alpha$	$(A \to A)$ и $\Phi_\alpha \to \alpha =_\beta \Phi_\alpha$	(сх. 1) $+\beta$
	5) Φ_α	MP 3, 4	
	6) α	MP 5, 3	

Парадокс Карри: «Если данное высказывание верно, то луна сделана из зелёного сыра». То есть,

$$\Phi_\alpha \leftrightarrow (\Phi_\alpha \to \alpha)$$

Лямбда-исчисление как вычислительная модель

- Из исчисления А. Чёрч выделил некоторую часть и доказал её непротиворечивость: Church, A. (1935). “A Proof of Freedom from Contradiction.” Proceedings of the National Academy of Sciences of the United States of America, 21(5):275–281.
- Но затем предложил смотреть на исчисление как на вычислительную модель: Church, A. (1936). “An Unsolvable Problem of Elementary Number Theory.” American Journal of Mathematics, 58(2):345–363, 1936.
- Начала современного понимания теории типов были заложены в этой работе: Church, A. (1940). A formulation of the simple theory of types, Journal of Symbolic Logic 5, pp. 56–68.

Примеры вычислений

$$f^{(n)}(x) = \begin{cases} x, & n = 0 \\ f(f^{(n-1)}(x)), & n > 0 \end{cases}$$

Определение

Чёрчевский нумерал $\overline{n} := \lambda f.\lambda x.f^{(n)}(x)$. *Например,* $\overline{3} = \lambda f.\lambda x.f(f(f(x)))$

Пример

Инкремент: $Inc := \lambda n.\lambda f.\lambda x.n\ f\ (f\ x)$ ~ все f из n плюс одно

$IsZero := \lambda n.n\ (\lambda x.F)\ T$

$Pair\ a\ b := \lambda s.s\ a\ b,\ Fst := \lambda p.p\ T,\ Snd := \lambda p.p\ F$

$Dec := \lambda n.Snd\ (n\ (\lambda p.Pair\ (Snd\ p)\ (\underbrace{Inc}_{+1}\ (\underline{Snd\ p}))))\ (Pair\ \overline{0}\ \overline{0})$

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

= 0,0 +1
0,1 +2
1,2 +3
2,3
⋮

Пример

$Fact\ n := Y\ (\lambda f.\lambda x.(IsZero\ x)\ \overline{1}\ (Mul\ x\ (f\ (Dec\ x))))\ n$

Порядок вычислений

$$Body := \lambda f.\lambda x.(IsZero\ x)\ \overline{1}\ (Mul\ x\ (f\ (Dec\ x)))$$
$$Fact\ n := Y\ Body\ n$$

Пример

Рассмотрим $Fact\ \overline{3} = Y\ Body\ \overline{3} \to_\beta Body\ (Y\ Body)\ \overline{3}$

Вычислять двумя способами:

1. $Body\ (Y\ Body)\ \overline{3} \to_\beta Body\ (Body\ (Y\ Body))\ \overline{3}$
2. $Body\ (Y\ Body)\ \overline{3} = [\lambda f.\lambda x.(IsZero\ x)\ \overline{1}\ (Mul\ x\ (f\ (Dec\ x)))]\ \underbrace{(Y\ Body)}_{f}\ \underbrace{\overline{3}}_{x} \twoheadrightarrow_\beta$
   $(IsZero\ \overline{3})\ \overline{1}\ (Mul\ \overline{3}\ (Y\ Body\ (Dec\ \overline{3})))$

Ну и дальше (при втором способе):

$(Mul\ \overline{3}\ (Y\ Body\ (Dec\ \overline{3}))) \twoheadrightarrow_\beta$

$Mul\ \overline{3}\ (Y\ Body\ \overline{2}) \to_\beta Mul\ \overline{3}\ (Body\ (Y\ Body\ \overline{2})) \to_\beta \ldots$ 6

Теорема Чёрча-Россера гарантирует, что результат будет одинаков, если будет.

Нормальный и аппликативный порядок вычислений

Пример

$\mathcal{K} := \lambda x.\lambda y.x,\ \mathcal{I} := \lambda x.x,\ \Omega := (\lambda x.x\ x)\ (\lambda x.x\ x)$

Выражение $K I \Omega$ можно редуцировать двумя способами:

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

Определение (нормальный порядок редукции)

Редукция самого левого β-редекса.

Определение (аппликативный порядок редукции)

Редукция самого левого β-редекса из самых вложенных.

Теорема (Приводится без доказательства)

Если нормальная форма существует, она может быть достигнута нормальным порядком редукции.

$(\lambda x.P)\quad (P\ Q)$

$\Omega = ((\lambda \underset{x}{x}.\underset{A}{xx})\underset{B}{(\lambda x.xx)})$

$(((\lambda \underset{x}{a}.\underset{A}{\lambda b.a})\underset{B}{I})(\Omega))$

1 сл.

1 сл.

$\underline{(\lambda x.A)\ B}$

редукс.

Нормальный порядок — медленный

Пример

1 ск.

Рассмотрим λ*-выражение* $(\lambda x.x\ x\ x\ x)(\mathcal{I}\,\mathcal{I})$. *Попробуем редуцировать его нормальным порядком:*

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

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

2 ск.

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

Как программировать? Любое значение – замыкание

```
let x = sqrt 256   let x = fun () -> sqrt 256
```

Плюс мемоизация:

```
let x = fun () -> sqrt 256;;
let y = x;;
y () + x () (* вычисляется два раза *)
```

Давайте запоминать результаты!

```
type int-value = Compute of unit -> int | Result of int;;
let compute v = match !v with
    Compute f -> let res = f () in v := Result res; res
  | Result r -> r;;
```

уже есть

```
let x = ref (Compute (fun () -> sqrt 256));;
let y = x;;
compute y + compute x
```

Ленивые и энергичные вычисления

Энерг: fact(15+4) => fact(19) → выч. fact.

Лен:

Энергичные вычисления: аппликативный порядок. Ленивые вычисления: нормальный порядок + мемоизация.

If всегда ленив

```
let fact n = if (n > 1) then n * fact (n-1) else 1
```

Ленивое общение с внешним миром бессмысленно.

Изоморфизм Карри-Ховарда

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

Просто типизированное λ-исчисление	Импликативный фрагмент ИИВ
Тип	Высказывание
Терм	Доказательство высказывания
Проверка типа	Проверка доказательства на корректность
Обитаемый тип	Доказуемое высказывание

$(\Rightarrow)$: изучение языков программирования

- Просто типизрованное исчисление соответствует исчислению высказываний. Малая выразительная сила просто-типизированного лямбда исчисления (полиномы).

  Выр. сила $\lambda_{\to}$ — полиномы

- Метод: усложняем исчисление — смотрим получающийся язык.
- Логика первого порядка: зависимые типы. Какой тип у sprintf?

  1 `sprintf "%d" : int -> string`

  2 `sprintf "%d + %d" : int*int -> string`

  printf(char* fmt, ...)

  Например, Идрис позволяет тип выписать.

- Логика второго порядка: генерики.

Программа	Доказывает
`let id x = x`	$\forall \tau . \tau \to \tau$

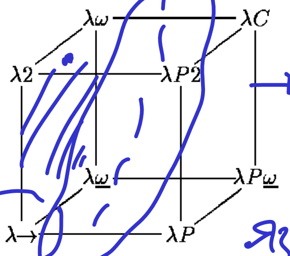
```
id<τ>(τ x){
  return x
}
```

- Классические функциональные языки — типовая система Хиндли-Милнера (разрешимый вариант системы F, соответствующей логике второго порядка).
- Алгоритмы вывода типов, анализ и верификация программ — используют матлог.

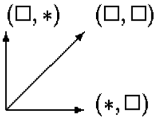
Лямбда-куб Барендрегта

<τ>τ f(τ a)
{return a}

f<int>(5)→5
f<string>("a")→"a"



Зн., зав от типов



типы от знач.

Языки с зависимыми типами

Типовые системы и языки программирования:

Классические и функциональные языки:

$\lambda_{\to}$	Классический Паскаль
$\lambda_{\underline{\omega}}$	Система F
λ_{ω}	Haskell, Ocaml

знач., знач

id : 3=3

Языки с зависимыми типами данных (обычно около λC):
Idris, Coq, Agda, Arend, C++ :).

$(\Leftarrow)$: изучение логики и доказательств через написание программ

7000000

17 7149324

- Пер Мартин-Лёф, Intuonistic Type Theory: версии 1972 и 1979.
- Множество расширений и вариантов.
- Такие инструменты как Coq, Agda, Lean используют варианты этой теории.
- Мы будем рассматривать некоторую родственную теорию, гомотопическую теорию типов.

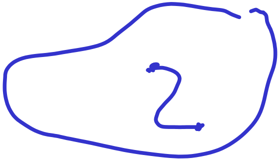
Гомотопическая теория типов

Владимир Александрович Воеводский, 1966-2017.

... Математика находится на пороге кризиса, а точнее двух кризисов. Первый связан с отрывом математики «чистой» от математики прикладной. Понятно, что рано или поздно встанет вопрос о том, а почему общество должно платить деньги людям, которые занимаются вещами, не имеющими никаких практических приложений. Второй, менее очевидный, связан с усложнением чистой математики, которое ведет к тому, что, опять же рано или поздно, статьи станут слишком сложными для детальной проверки и начнется процесс накопления незамеченных ошибок ...

Гомотопическая теория типов

refl : 5=5



- Центральный вопрос — что такое равенство.
- Классический матлог: это предикат, удовлетворяющий свойствам.
- Однако, свойства обычно слишком общие (класс эквивалентности?). Интуитивно хочется большего, равенство не всегда просто эквивалентность.
- Изоморфизм Карри-Ховарда-Воеводского:

Логика	λ-исчисление	Топология
Утверждение	Тип	Пространство
Доказательство	Значение	Точка в пространстве
Предикат $(=)$	Зависимый тип $(=)$	Путь между точками

- Реализация: кубическая теория типов, Аренд.

Построение курса

1. Аналогично с матлогом, будет разделение на теорию и практику.
2. Теория: знание определений, идей, теорем.
3. Практика: лабы на доказательства теорем с использованием языка Аренд, возможны дополнительные околокомпиляторные лабы.
4. Для закрытия предмета надо набрать баллы практическими заданиями и сдать зачёт/экзамен.

$$\text{Not } x \equiv x \to \bot$$

Логика 1 порядка: кванторы по предм. перем.

$$[\![\forall x.\ x^2 \geq 0]\!]^{\text{матан.}} = И \qquad [\![x^2 \geq 0]\!]^{x:=5}_{x \in D} = И$$

Логика 2 порядка: кванторы по пред.

Самый прост. пред — проп. перем.

$\langle p \rangle\ f\ (p\ x) \to \cancel{\text{Not } p}\ (p \to \bot)$

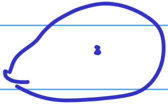
$\{ \dots \}$

$\forall p.\ p \to \neg p = \langle p \rangle\ p \to (p \to \bot)$

"Простая" лог. 2 пор: кванторы по проп. перем.

$\mathbb{Z} = \langle a, b \rangle \Big/ {\langle a,b\rangle \simeq \langle c,d\rangle}$

$a+d=b+c$?



$\bar{1} = \{\langle 1,0\rangle, \langle 2,1\rangle, \langle 3,2\rangle \dots\}$

$\bar{1} \not\updownarrow \bar{2}$

$\cdot\ 0$

$\cdot\ 1$

$\cdot\ 2$

$\cdot\ 3$

$\cdot\ 4$

$\cdot\ 5$

Путь: $[0;1] \to N$

$a=b$ — если сущ. путь, что $\pi(0)=a$, $\pi(1)=b$

$\lambda x.2 \quad : 2=2$

$\lambda a.\lambda x.a \quad : \forall a.a=a$