Математические модели вычислений

Общие понятия теории категорий

Индуктивные типы

Правила итерации и индукции

Коиндуктивные типы

Альтернатива Тьюринг-полноте — завершимость и продуктивность

maxim.krivchikov@gmail.com

Материалы курса: https://maxxk.github.io/formal-models-2015/

По следам наших публикаций

Осталось 3 занятия — 1.12, 8.12, 15.12.

Предварительный список литературы курса — на сайте

Как мы говорили ранее, $A \times B$ — это частный случай $\Sigma A.\ B$, а $A \to B$ — частный случай $\Pi A.\ B$, поэтому далее будем использовать и те, и другие обозначения. Для терма приложения, если это уместно, можем использовать $A(x, y, z)$ как $A \cdot x \cdot y \cdot z$.

Теория категорий

— достаточно новая область математики, которая используется для описания абстрактных математических структур (близка к алгебраической топологии).

«наука о стрелочках»

Определение. Категория C состоит из следующих частей:

- класса (набора) *объектов* $\mathrm{Ob}(C)$
- класса *морфизмов* (*стрелок*) между объектами $\mathrm{Hom}_C(a, b)$ с равенством; пример записи: стрелка $f\colon a \to b$ из a в b
- оператора *композиции* ($\circ$): для любых $f\colon a \to b$, $g\colon b \to c$, существует стрелка $g \circ f\colon a \to c$.

  При этом оператор композиции должен обладать свойствами:
- *ассоциативность*: $h \circ (g \circ f) = (h \circ g) \circ f$
- *существование единицы*: для любого объекта x существует единичный морфизм $1_x\colon x \to x$, нейтральный для композиции $1_x \circ f = f$, $g \circ 1_x = g$.

С. Маклейн. Категории для работающего математика. М.:Физматлит, 2004.

Теория категорий

Примеры

Самый простой пример — категория множеств Set, объекты которой — множества, а морфизмы — функции между множествами.
Пример посложнее — направленный граф (объекты — вершины, стрелки — рёбра и единичные стрелки-циклы у вершин).

Определения (и доказательства) в теории категорий часто используют коммутативные диаграммы — направленные графы с вершинами-объектами и рёбрами-морфизмами:

$$\begin{tikzcd} F(X) \arrow[r, "F(i)"] \arrow[d, "\alpha"'] & F(F(X)) \arrow[d, "F(\alpha)"] \\ X \arrow[r, "i"'] & F(X) \end{tikzcd}$$

«Диаграмма коммутирует» означает, что при любом выбранном начальном и конечном объекте, для всех путей композиции составляющих их морфизмов равны.

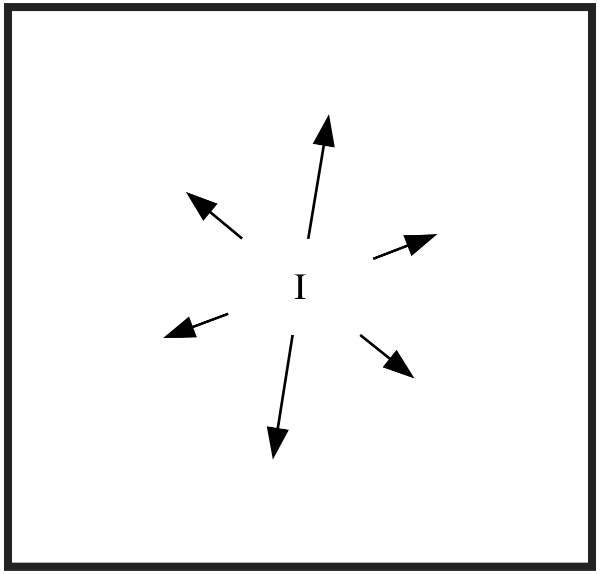
Теория категорий

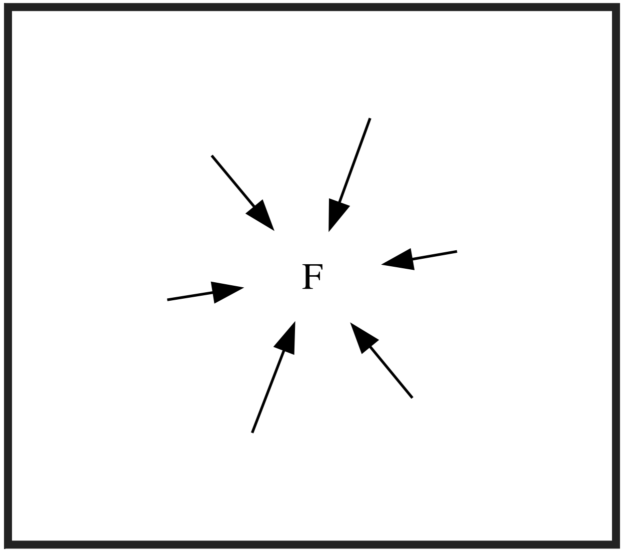
инициальный объект

такой объект I, из которого выходят морфизмы в любой объект категории (единственные для каждого объекта — $\forall X\ \exists!\ f : I \to X$)

терминальный (финальный) объект

такой объект F, в который входят морфизмы из любого объекта категории (единственные для каждого объекта — $\forall X\ \exists!\ g : X \to F$)





Теория категорий

функтор

отображение между категориями C и D, такое, что:

- каждому объекту $X \in Ob(C)$ сопоставляется $F(X) \in Ob(D)$
- каждому морфизму $f\colon X \to Y \in Hom(C)$ сопоставляется $F(f)\colon F(X) \to F(Y)$
- сохраняется идентичность и композиция морфизмов

эндофунктор

функтор из категории C в себя ($F\colon C \to C$)

F-алгебра

для эндофунктора F — пара (A, α), где A — объект, $\alpha\colon F(A) \to A$

F-алгебры образуют категорию с морфизмами f для F-алгебр (A, α) и (B, β) (гомоморфизмами):

$$\begin{tikzcd} F(A) \arrow[r, "\alpha"] \arrow[d, "F(f)"'] & A \arrow[d, "f"] \\ F(B) \arrow[r, "\beta"] & B \end{tikzcd}$$

Индуктивный тип

Мотивация: хочется работать с объектами произвольного размера и задавать функции на них индуктивно по их структуре.

Примеры

Натуральные числа

тип $\mathbb{N}$, с двумя конструкторами 0 и $S(k : \mathbb{N})$ (следующее за данным; $S(k) \equiv k + 1$)

простое удаление (итерация):

$$\mathrm{iter}_{\mathbb{N}}(T : \mathbf{Type}, c_0 : T, c_S : T \to T, n : \mathbb{N}) : T$$

зависимое удаление (индукция):

$$\mathrm{ind}_{\mathbb{N}}(T : \mathbb{N} \to \mathbf{Type}, c_0 : T(0), c_S : \Pi(n : \mathbb{N}).\ T(n) \to T(S(n)), n : \mathbb{N}) : T(n)$$

редукция:

- $\mathrm{ind}_{\mathbb{N}}(\ldots, 0) \longrightarrow c_0$
- $\mathrm{ind}_{\mathbb{N}}(\ldots, k + 1) \longrightarrow c_S(k, \mathrm{ind}_{\mathbb{N}}(\ldots, k))$

Индуктивный тип

Контрпример

Не-индуктивный тип на основе натуральных чисел

тип булевых (#2) векторов заданного размера $\mathrm{BVector} : \mathbb{N} \to \mathbf{Type}$
определяется через итерацию:
$\mathrm{BVector}(\mathrm{elements} : \mathbb{N}) \equiv$

$$\mathrm{ind}_{\mathbb{N}}(\mathbf{Type}, c_0 \equiv \#1, c_S \equiv \lambda(n : \mathbb{N}).\ (v : \mathrm{BVector}(n).\ \#2 \times \mathrm{BVector}(n)), \mathrm{elements})$$

прототип функции, которая требует непустой вектор:
(существует $n : \mathbb{N}$, для которого длина вектора — $\mathbb{N} + 1$)

$$\mathrm{negateVector} : \Pi(n : \mathbb{N}).\mathrm{BVector}(n{+}1) \to \mathrm{BVector}(n{+}1)$$

прототип функции, которая требует два вектора одинакового размера:

$$\mathrm{andVector} : \Pi(n : \mathbb{N}).\mathrm{BVector}(n) \to \mathrm{BVector}(n) \to \mathrm{BVector}(n)$$

Индуктивный тип

Примеры

Список из элементов типа A

тип List(A) с двумя конструкторами nil_A (пустой список, []) и $\mathrm{cons}_A(h : A, t : \mathrm{List}(A))$ ([h, t])

простое удаление (reduce, fold):

$$\mathrm{iter}_{\mathrm{List}}(T : \mathbf{Type}, c_{\mathrm{nil}} : T, c_{\mathrm{cons}} : T \to A \to T, l : \mathrm{List}(A)) : T$$

Формула алгебры логики

тип Formula с конструкторами:

- var(x : Nat), const(x : #2)
- not(x : Formula)
- or(x, y : Formula), and(x, y : Formula)

Противоречивый индуктивный тип

тип Bad с двумя конструкторами nothing : #1 и fn : Bad → #1

противоречивый терм:

$$\omega = \mathrm{iter}_{\mathrm{Bad}}(\#1, c_{\mathrm{nothing}} = 0\#1, c_{\mathrm{fn}} = \lambda(t : \mathrm{Bad} \to \#1).\ t \cdot \mathrm{fn}(t), \mathrm{fn}(\lambda(t : \mathrm{Bad}).\ 0\#1))$$

Индуктивный тип

Первое определение

Рассмотрим категорию типов Исчисления Конструкций Type с морфизмами — функциями между ними.

Индуктивным типом на основе эндофунктора $F : \mathbf{Type} \to \mathbf{Type}$ назовём *инициальный объект F-алгебры* $\mu F : \mathbf{Type}$.

Для рассмотренных примеров эндофункторы:

- $F_{\mathbb{N}} \equiv \lambda(N : \mathbf{Type}).\ 1 + N$
- $F_{\mathrm{List}} \equiv \lambda(A : \mathbf{Type}).\ \lambda(L : \mathbf{Type}).\ 1 + A \times L$
- $F_{\mathrm{Formula}} \equiv \lambda(F : \mathbf{Type}).\ \mathbb{N} + \#2 + (F \times F) + (F \times F) + F$
- `$F_{Bad}` $\equiv \lambda(B : \mathbf{Type})\ .\ 1 + (B \to 1)$

Индуктивный тип

Теоретико-категориальная семантика

Алгебра — это пара из объекта и морфизма.
Объект μF — это индуктивный тип.
Морфизм α — это набор термов введения объекта.

Инициальность — наличие гомоморфизмов к любым типам — определяет терм простого удаления. Пример на натуральных числах:

$$\begin{tikzcd} 1 + \mathbb{N} \arrow[r, "\alpha"] \arrow[d, "F(f)"'] & \mathbb{N} \arrow[d, "f"] \\ 1 + T \arrow[r, "\beta"] & T \end{tikzcd}$$

$$\forall\, T, (1 + T) \to T \Rightarrow \mathbb{N} \to T$$

Индуктивный тип

«Идеальный» терм удаления выглядит следующим образом (но не исключает противоречивые термы как в примере ранее — фактически, это чуть изменённый Y-комбинатор):

$$\Pi(T : \mathbf{Type}).(step : \mu F \to T).F(\mu f) \to T$$

step — рекурсивный вызов

Термы зависимого удаления (индукции) также имеют теоретико-категориальную семантику:

1. Fumex C., Ghani N., Johann P. Indexed induction and coinduction, fibrationally // Algebra and Coalgebra in Computer Science. Springer, 2011. P. 176–191.
2. Ghani N. et al. Fibred Data Types // Proceedings of the 2013 28th Annual ACM/IEEE Symposium on Logic in Computer Science. IEEE Computer Society, 2013. P. 243–252.
3. Препринты: http://arxiv.org/find/cs/1/au:+Ghani_N/0/1/0/all/0/1

Индуктивный тип

Второе определение

Введём понятие «строго положительного» вхождения переменной X в терм τ.

- любое вхождение в терм без зависимых произведений (τ=X × X, $\tau = \Sigma X.\#2$) строго положительное, если есть зависимые произведения — смотрим на структуру.
- для ΠA. B, если X не встречается в A, вхождения строго положительные
- ΠX. B — отрицательное вхождение X.
- ΠA. B — если X встречается в A в отрицательных позициях, то его вхождение в τ положительно, но не строго положительно.

Т.е. X не должен стоять «слева» от стрелки вообще (строго положительное) или в самой левой части стрелки (положительное)

Примеры строго положительных вхождений: $X \times X$, $\mathbb{N} \to X$, $X \times (\mathbb{N} \to X)$
Пример нестрого положительного вхождения: $(X \to \mathbb{N}) \to \mathbb{N}$
Примеры отрицательных вхождений: $X \to \mathbb{N}$, $(X \times \mathbb{N}) \to X$, $F(X) \to X$, $(\mathbb{N} \to X) \to X$ (?)

Индуктивный тип

Второе определение

μF — набор *конструкторов* — типов, в которых переменная итерации X может встречаться только в строго положительных позициях.

Строгая положительность позволяет определить терм удаления в «идеальном» виде и не получить при этом возможность парадоксов.

Можно описать в той же теоретико-категориальной семантике, но нужно использовать нетривиальное понятие *полиномиальных функторов* (требует определения равенства).

1. Paulin-Mohring C. Inductive Definitions in the System Coq - Rules and Properties // TLCA ’93 Proceedings of the International Conference on Typed Lambda Calculi and Applications. Springer-Verlag London, UK, 1993. P. 328–345.

Индуктивный тип

Определение по неподвижным точкам

Индуктивный тип — наименьшая неподвижная точка $F^i(\bot)$.

$$\oslash \to F(\oslash) \to F(F(\oslash)) \to \ldots \to F^{\infty}(\oslash)$$

Определение в стиле Мендлера

В явном виде подходит только для System F_ω (т.к. требует Type : Type).

$$\mu F \equiv \Pi(G : \mathbf{Type}).\Pi(X : \mathbf{Type}).\Pi(step : X \to G).F{\cdot}X \to G$$

Сигнатура типа задаёт итерацию на нём. Без Type : Type можно ввести μF как терм для произвольного F : **Type → Type**, но разрешить удалять только с помощью терма Мендлера.

1. Constable R.L., Mendler N.P. Recursive Definitions in Type Theory // Logics of Programs, Conference, Brooklyn College, June 17--19, 1985, Proceedings / ed. Parikh R. 1985. Vol. 193. P. 61–78.
2. Abel A., Matthes R., Uustalu T. Iteration and coiteration schemes for higher-order and nested datatypes // Theoretical Computer Science. 2005. Vol. 333, № 1-2. P. 3–66.
3. Ahn K.Y., Sheard T. A hierarchy of mendler style recursion combinators // ACM SIGPLAN Notices. 2011. Vol. 46, № 9. P. 234–246.

Определение с аннотацией размера

- вводим ординалы ord
- рассматриваем индуктивный тип в определении наименьшей неподвижной точки; терм введения увеличивает размер ($S : \mathbb{N}^{(k)} \to \mathbb{N}^{(k+1)}$)
- 1.Abel A. Towards Generic Programming with Sized Types // Mathematics of Program Construction / ed. Uustalu T. Springer Berlin Heidelberg, 2006. P. 10–28.
2.Abel A. Implementing a normalizer using sized heterogeneous types. // J. Funct. Program. 2009. № 510996. P. 1–24.

Простые расширения

Взаимно индуктивные типы

Несколько типов $A_1, \ldots, A_k$, определения которых могут ссылаться друг на друга.

Пример (CPDT) — типы списков с контролируемой чётностью длины:

$$\mu(\mathrm{EvenList} = \#1 + (\mathbb{N} \times \mathrm{OddList})$$
$$\mathrm{OddList} = \mathrm{N} \times \mathrm{EvenList})$$

Индексированные индуктивные типы

(индуктивные предикаты — $\Pi A.\mathbf{Type}$ для некоторого A)

Чётность ($T : \Pi\mathbb{N}.\mathbf{Type}$) — в форме набора конструкторов:

$$\mu_{\Pi\mathbb{N}.\mathbf{Type}}\mathrm{Even} =$$

$$\mathrm{even}_z\mathrm{ero} : \mathrm{Even}(0)$$

$$\mathrm{even}_p\mathrm{lus}_2 : \Pi(k : \mathbb{N}).\ \mathrm{Even}(k) \to \mathrm{Even}(S(S(k)))$$

Коиндуктивные типы

Коиндуктивный тип — наибольшая неподвижная точка $F^i(\#1)$, νF.

Пример

поток (потенциально бесконечная последовательность)
$\#1, \mathbb{N} \times \#1, \mathbb{N} \times \mathbb{N} \times \#1, \ldots, \mathbb{N} \times \mathbb{N} \times \ldots \times \mathbb{N} \times \ldots$

операции:

- удаление — взять элемент с начала потока $\nu F \to F(\nu F)$
- введение — сгенерировать поток по некоторым начальным данным

Коиндуктивные типы

Теоретико-категориальная семантика — дуальна индуктивным типам. Дуализм в теории категорий — все морфизмы разворачиваются. Вместо инициального объекта в категории F-алгебр — терминальный объект в категории F-коалгебр.

В стиле Мендлера:
generate : $\Pi(G : \mathbf{Type}).\Pi(X : \mathbf{Type}).\Pi(\mathrm{step} : G \to X).G \to F(X)$

На уровне программ:

- индуктивные типы — обеспечивают завершимость (termination) структурно-рекурсивных функций
- коиндуктивные типы — обеспечивают продуктивность (productivity) структурно-рекурсивных генераторов (следующий уровень детализации объекта может быть получен за конечное число шагов)

Нетривиальные расширения

Индуктивно-рекурсивные и индуктивно-индуктивные типы

Одновременно определяется индуктивный тип $\mathrm{U} : \mathbf{Type}$ и рекурсивная функция $\mathrm{T} : \mathrm{U} \to \mathbf{Type}$.

1. Forsberg F.N., Setzer A. A finite axiomatisation of inductive-inductive definitions. 2012.
2. Ghani N. et al. Fibred Data Types // Proceedings of the 2013 28th Annual ACM/IEEE Symposium on Logic in Computer Science. IEEE Computer Society, 2013. P. 243–252.

Индуктивные типы над произвольным видом

Можно ли (и с какими ограничениями) определить индуктивные/коиндуктивные типы $\mu\mathrm{F}$, где $\mathrm{F} : \mathrm{A} \to \mathrm{A}$ над произвольным типом A? Ограниченным типом A?

Тип идентичности (равенства)

Для произвольного типа A — индексированный индуктивный тип $\mathrm{Id} : \Pi A.\ A.\ \mathbf{Type}$

$(\mathrm{Id}(a, b) \equiv a = b)$.

Реализует равенство Лейбница.

Единственный конструктор — рефлексивность: $\mathrm{refl}_A : \Pi(a : A).\ \mathrm{Id}(a, a)$.

Терм зависимого удаления — конвертация по равенству:

$$J(a, b : A, \varepsilon : a =_A b, C : \Pi(x, y : A).\ (\delta : x =_A y).\ \mathbf{Type}, x : C(a, a, \mathrm{refl}_A(a))) : C(a, b, \varepsilon)$$

Задачи со звёздочкой

Задача 8.1* Записать термы зависимого удаления и правила редукции для списка, формул алгебры логики и противоречивого типа.

Задача 8.2** С помощью термов зависимого удаления реализовать (в исчислении конструкций) интерпретатор формул, который позволяет получить из окружения переменных $\mathrm{Env} : \mathbb{N} \to \#2$ и формулы Formula результат вычисления формулы с такими переменными

- дополнительная * — Coq

Задача 8.3** Используя дуальность индуктивных и коиндуктивных типов, записать правило удаления коиндуктивного типа, задаваемого функтором F в стиле Мендлера.