

# Функциональное программирование

## Лекция 2. Рекурсия и редукция

Денис Николаевич Москвин

Кафедра математических и информационных технологий  
Санкт-Петербургского академического университета

17.02.2012

- 1 Теорема о неподвижной точке
- 2 Редексы и нормальная форма
- 3 Теорема Чёрча-Россера
- 4 Стратегии редукции

- 1 Теорема о неподвижной точке
- 2 Редексы и нормальная форма
- 3 Теорема Чёрча-Россера
- 4 Стратегии редукции

Схема  $\beta$ -преобразования  $(\lambda x. M) N = M[x := N]$  даёт возможность решать простейшие уравнения на термы.

## Пример

Найти  $F$ , такой что  $\forall M, N, L \lambda \vdash F M N L = M L (N L)$ .

$$F M N L = M L (N L)$$

$$F M N = \lambda z. M z (N z)$$

$$F M = \lambda y. \lambda z. M z (y z)$$

$$F = \lambda x y z. x z (y z)$$

А если уравнение рекурсивное, например,  $F M = M F$ ?

Схема  $\beta$ -преобразования  $(\lambda x. M) N = M[x := N]$  даёт возможность решать простейшие уравнения на термы.

## Пример

Найти  $F$ , такой что  $\forall M, N, L \lambda \vdash F M N L = M L (N L)$ .

$$F M N L = M L (N L)$$

$$F M N = \lambda z. M z (N z)$$

$$F M = \lambda y. \lambda z. M z (y z)$$

$$F = \lambda x y z. x z (y z)$$

А если уравнение рекурсивное, например,  $F M = M F$ ?

Оказывается, имеется универсальный способ решения!

# Теоремы о неподвижной точке

## Теорема

Для любого  $\lambda$ -терма  $F$  существует неподвижная точка:

$$\forall F \in \Lambda \quad \exists X \in \Lambda \quad \lambda \vdash FX = X$$

## Доказательство

Введем  $W \equiv \lambda x. F(x x)$  и  $X \equiv WW$ . Тогда

$$X \equiv WW \equiv (\lambda x. F(x x)) W = F(WW) \equiv FX \quad \blacksquare$$

## Теорема

Существует комбинатор неподвижной точки  $Y$ , такой что

$$\forall F \quad F(YF) = YF.$$

## Доказательство

Введём  $Y \equiv \lambda f. (\lambda x. f(x x))(\lambda x. f(x x))$ . Имеем  $YF \equiv$

$$(\lambda x. F(x x))(\lambda x. F(x x)) = F(\underbrace{((\lambda x. F(x x))(\lambda x. F(x x)))}_{YF}) \equiv F(YF) \quad \blacksquare$$

Y-комбинатор позволяет ввести рекурсию в  $\lambda$ -исчисление.

## Пример

Факториал рекурсивно:

$$\text{fac} = \lambda n. \text{iif} (\text{iszro } n) 1 (\text{mult } n (\text{fac} (\text{pred } n)))$$

Переписываем в виде

$$\text{fac} = \underbrace{(\lambda f n. \text{iif} (\text{iszro } n) 1 (\text{mult } n (f (\text{pred } n))))}_{\text{fac}' } \text{fac}$$

Отсюда видно, что  $\text{fac}$  — неподвижная точка для вспомогательной функции  $\text{fac}'$ :

$$\text{fac} = Y \text{fac}'$$

Как работает  $\text{fac} \equiv Y \text{ fac}'$ ?

## Пример

```
fac 3 = (Y fac') 3
      = fac' (Y fac') 3
      = iif (iszro 3) 1 (mult 3 ((Y fac') (pred 3)))
      = mult 3 ((Y fac') 2)
      = mult 3 (fac' (Y fac') 2)
      = mult 3 (mult 2 ((Y fac') 1))
      = mult 3 (mult 2 (mult 1 ((Y fac') 0)))
      = mult 3 (mult 2 (mult 1 1))
      = 6
```



- 1 Теорема о неподвижной точке
- 2 Редексы и нормальная форма
- 3 Теорема Чёрча-Россера
- 4 Стратегии редукции

Мы строили  $\lambda$ -исчисление как теорию о равенстве термов.

А как можно было бы доказать неравенство?

Мы строили  $\lambda$ -исчисление как теорию о равенстве термов.

А как можно было бы доказать неравенство?

## Пример

$$\mathbf{K I} \equiv (\lambda x y. x) (\lambda z. z) = \lambda y z. z$$

$$\mathbf{I I K}_* \equiv (\lambda x. x) \mathbf{I K}_* = \mathbf{I K}_* \equiv (\lambda x. x) (\lambda y z. z) = \lambda y z. z$$

Видно, что процесс носит односторонний характер: термы при конверсиях «упрощаются». Для исследования подобного вычислительного аспекта вводят понятие **редукции**:

- $\mathbf{K I} \rightarrow_{\beta} \mathbf{K}_*$  — редуцируется за один шаг;
- $\mathbf{I I K}_* \rightarrow_{\beta} \mathbf{K}_*$  — редуцируется;
- $\mathbf{K I} =_{\beta} \mathbf{I I K}_*$  — конвертируемо (равно).

## Определение

Терм вида  $(\lambda x. M) N$  называется  $\beta$ -редексом.

## Определение

Терм  $M[x := N]$  называется *сокращением* редекса  $(\lambda x. M) N$ .

## Пример

Терм  $I (K I)$  содержит два редекса

$$(\lambda x. x) ((\lambda y z. y) (\lambda p. p))$$
$$(\lambda x. x) ((\lambda y z. y) (\lambda p. p))$$

Может ли сокращение увеличить число редексов?

## Определение

Бинарное отношение  $\mathcal{R}$  над  $\Lambda$  называют **совместимым** (с операциями  $\lambda$ -исчисления), если для любых  $M, N, Z \in \Lambda$ :

$$\begin{aligned}M \mathcal{R} N &\Rightarrow (ZM) \mathcal{R} (ZN), \\ &\quad (MZ) \mathcal{R} (NZ), \\ &\quad (\lambda x. M) \mathcal{R} (\lambda x. N).\end{aligned}$$

## Определение

Совместимое отношение эквивалентности называют отношением **конгруэнтности** над  $\Lambda$ .

## Определение

Совместимое, рефлексивное и транзитивное отношение называют отношением **редукции** над  $\Lambda$ .

## Определение

Бинарное отношение  $\beta$ -редукции за один шаг  $\rightarrow_\beta$  над  $\Lambda$ :

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

$$M \rightarrow_\beta N \Rightarrow ZM \rightarrow_\beta ZN$$

$$M \rightarrow_\beta N \Rightarrow MZ \rightarrow_\beta NZ$$

$$M \rightarrow_\beta N \Rightarrow \lambda x. M \rightarrow_\beta \lambda x. N$$

По определению  $\rightarrow_\beta$  является совместимым (с операциями  $\lambda$ -исчисления).

## Пример: редуцируем терм $I (K I)$

$$(\lambda x. x) ((\lambda y z. y) (\lambda p. p)) \rightarrow_\beta (\lambda y z. y) (\lambda p. p) \rightarrow_\beta \lambda z p. p$$

$$(\lambda x. x) ((\lambda y z. y) (\lambda p. p)) \rightarrow_\beta (\lambda x. x) (\lambda z p. p) \rightarrow_\beta \lambda z p. p$$

## Определение

Бинарное отношение  $\beta$ -редукции  $\rightarrow_\beta$  над  $\Lambda$  (индуктивно):

$$(a) \quad M \rightarrow_\beta M$$

$$(b) \quad M \rightarrow_\beta N \Rightarrow M \rightarrow_\beta N$$

$$(c) \quad M \rightarrow_\beta N, N \rightarrow_\beta L \Rightarrow M \rightarrow_\beta L$$

Отношение  $\rightarrow_\beta$  является **транзитивным рефлексивным** замыканием  $\rightarrow_\beta$  и, следовательно, отношением редукции.

## Примеры

$$(\lambda x. x) ((\lambda y z. y) (\lambda p. p)) \rightarrow_\beta (\lambda x. x) ((\lambda y z. y) (\lambda p. p))$$

$$(\lambda x. x) ((\lambda y z. y) (\lambda p. p)) \rightarrow_\beta (\lambda y z. y) (\lambda p. p)$$

$$(\lambda x. x) ((\lambda y z. y) (\lambda p. p)) \rightarrow_\beta \lambda z p. p$$

# Отношение конвертируемости $=_{\beta}$

## Определение

Бинарное отношение  $=_{\beta}$  над  $\Lambda$  (индуктивно):

$$(a) \quad M \rightarrow_{\beta} N \Rightarrow M =_{\beta} N$$

$$(b) \quad M =_{\beta} N \Rightarrow N =_{\beta} M$$

$$(c) \quad M =_{\beta} N, N =_{\beta} L \Rightarrow M =_{\beta} L$$

Отношение  $=_{\beta}$  является отношением конгруэнтности.

## Утверждение

$$M =_{\beta} N \Leftrightarrow \lambda \vdash M = N.$$

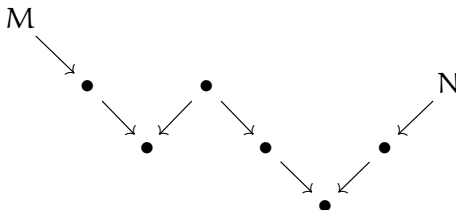
## Доказательство

Индукция по определениям. ■

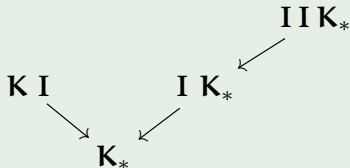


# Отношение конвертируемости $=_{\beta}$ (интуитивно)

Интуитивно: два терма  $M$  и  $N$  связаны отношением  $=_{\beta}$ , если есть связывающая их цепочка  $\rightarrow_{\beta}$ -стрелок:



Пример.  $K I =_{\beta} I I K_*$



## Определение

$\lambda$ -терм  $M$  **находится** в  $\beta$ -нормальной форме ( $\beta$ -NF), если в нем нет подтермов, являющихся  $\beta$ -редексами.

## Определение

$\lambda$ -терм  $M$  **имеет**  $\beta$ -нормальную форму, если для некоторого  $N$  выполняется  $M =_{\beta} N$  и  $N$  находится в  $\beta$ -NF.

## Примеры

- Терм  $\lambda x y. y (\lambda z. x)$  находится в  $\beta$ -нормальной форме.
- Терм  $(\lambda x. x x) y$  не находится в  $\beta$ -нормальной форме, но имеет в качестве  $\beta$ -nf терм  $y y$ .

# Нормальная форма (2)

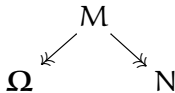
## Утверждение

Не все термы имеют  $\beta$ -нормальную форму.

## Пример

$$\begin{aligned}\Omega &\equiv \omega \omega \\ &\equiv (\lambda x. x x) (\lambda x. x x) \\ &\rightarrow_{\beta} (\lambda x. x x) (\lambda x. x x) \\ &\rightarrow_{\beta} \dots\end{aligned}$$

Это пока не доказательство! Может быть существует терм  $N$  в  $\beta$ -NF, такой что  $\Omega =_{\beta} N$ , например, так



# Нормальная форма (3)

Бывают термы, «удлинняющиеся» при редукции.

## Пример

$$\begin{aligned}\Omega_3 &\equiv \omega_3 \omega_3 \\ &\equiv (\lambda x. x x x) (\lambda x. x x x) \\ &\rightarrow_{\beta} (\lambda x. x x x) (\lambda x. x x x) (\lambda x. x x x) \\ &\rightarrow_{\beta} (\lambda x. x x x) (\lambda x. x x x) (\lambda x. x x x) (\lambda x. x x x) \\ &\rightarrow_{\beta} \dots\end{aligned}$$

С какой скоростью будет расти  $\Omega_4 \equiv \omega_4 \omega_4$ ?

# Нормальная форма (4)

Не все последовательности редукций приводят к  $\beta$ -NF.

## Пример

$$\begin{aligned} \mathbf{KI}\Omega &\equiv \mathbf{KI}((\lambda x. x x) (\lambda x. x x)) \\ &\rightarrow_{\beta} \mathbf{KI}((\lambda x. x x) (\lambda x. x x)) \\ &\rightarrow_{\beta} \dots \end{aligned}$$

$$\begin{aligned} \mathbf{KI}\Omega &\equiv (\lambda x y. x) \mathbf{I} \Omega \\ &\rightarrow_{\beta} (\lambda y. \mathbf{I}) \Omega \\ &\rightarrow_{\beta} \mathbf{I} \end{aligned}$$

(синим отмечен сокращаемый редекс)

# Редукционные графы (1)

## Определение

**Редукционный граф** терма  $M \in \Lambda$  (обозначаемый  $G_\beta(M)$ ) — это ориентированный мультиграф с вершинами в  $\{N \mid M \rightarrow_\beta N\}$  и дугами  $\rightarrow_\beta$ .

$$G_\beta(\mathbf{I}(\mathbf{I}x)) = \bullet \begin{array}{c} \curvearrowright \\ \longrightarrow \\ \curvearrowleft \end{array} \bullet \longrightarrow \bullet \quad G_\beta(\mathbf{\Omega}) = \bullet \begin{array}{c} \curvearrowright \\ \longrightarrow \\ \curvearrowleft \end{array} \bullet$$

$$G_\beta((\lambda x. \mathbf{I}) \mathbf{\Omega}) = \bullet \begin{array}{c} \curvearrowright \\ \longrightarrow \\ \curvearrowleft \end{array} \bullet \quad G_\beta(\mathbf{KI} \mathbf{\Omega}) = \bullet \begin{array}{c} \curvearrowright \\ \longrightarrow \\ \curvearrowleft \end{array} \bullet \longrightarrow \bullet \begin{array}{c} \curvearrowright \\ \longrightarrow \\ \curvearrowleft \end{array} \bullet \longrightarrow \bullet$$

$$G_\beta(\mathbf{\Omega}_3) = ??? \quad G_\beta((\lambda x. \mathbf{I}) \mathbf{\Omega}_3) = ???$$



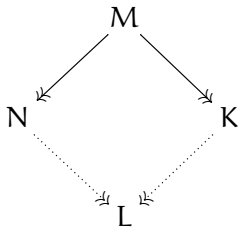
- 1 Теорема о неподвижной точке
- 2 Редексы и нормальная форма
- 3 Теорема Чёрча-Россера**
- 4 Стратегии редукции



## Теорема [Чёрч-Россер]

Если  $M \rightarrow_{\beta} N$ ,  $M \rightarrow_{\beta} K$ , то существует  $L$ , такой что  $N \rightarrow_{\beta} L$  и  $K \rightarrow_{\beta} L$ .

- Иначе говоря,  $\beta$ -редукция обладает *свойством ромба*:



- Иногда используют термин *конфлюентность*.

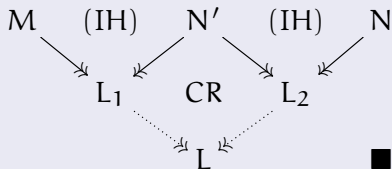
# Следствия теоремы Чёрча-Россера (1)

Теорема о существовании общего редукта

Если  $M =_{\beta} N$ , то существует  $L$ , такой что,  $M \rightarrow_{\beta} L$  и  $N \rightarrow_{\beta} L$ .

Доказательство (индукция по генерации  $=_{\beta}$ )

- $M =_{\beta} N$ , поскольку  $M \rightarrow_{\beta} N$ . Возьмем  $L \equiv N$ .
- $M =_{\beta} N$ , поскольку  $N =_{\beta} M$ . По гипотезе индукции имеется общий  $\beta$ -редукт  $L_1$  для  $N, M$ . Возьмем  $L \equiv L_1$ .
- $M =_{\beta} N$ , поскольку  $M =_{\beta} N'$ ,  $N' =_{\beta} N$ . Тогда



# Следствия теоремы Чёрча-Россера (2)

## Теорема [Редуцируемость к NF]

Если  $M$  имеет  $N$  в качестве  $\beta$ -NF, то  $M \rightarrow_{\beta} N$ .

Теперь мы можем доказать отсутствие NF у  $\Omega$ . Иначе выполнялось бы

$$\Omega \rightarrow_{\beta} N, \quad N \text{ является } \beta\text{-NF.}$$

Но  $\Omega$  редуцируется лишь к себе и не является  $\beta$ -NF.

## Теорема [Единственность NF]

$\lambda$ -терм имеет не более одной  $\beta$ -NF.

Теперь мы можем доказывать «неравенства», например

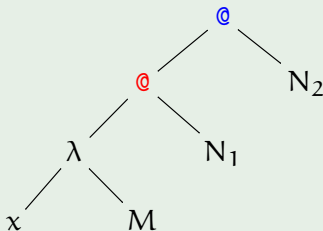
$$\lambda \not\sim \text{tru} = \text{fls}.$$

Иначе было бы  $\text{tru} =_{\beta} \text{fls}$ , но это две разные NF, что противоречит единственности.

- 1 Теорема о неподвижной точке
- 2 Редексы и нормальная форма
- 3 Теорема Чёрча-Россера
- 4 Стратегии редукции

- Как мы можем редуцировать терм?
  - Переменная:  $v$  — редукция завершена.
  - Абстракция:  $\lambda x. M$  — редуцируем  $M$ .
  - Аппликация:  $M N$ . Все варианты отсюда.
- Разбираем аппликацию до не-аппликации (обычно влево)
  - $(\dots ((v N_1) N_2) \dots N_k)$  — редуцируем отдельно все  $N_i$  (обычно слева направо).
  - $(\dots (((\lambda x. M) N_1) N_2) \dots N_k)$ . Все варианты отсюда.
- **Нормальная стратегия:** сокращаем редекс  $(\lambda x. M) N_1$ .
- **Аппликативная стратегия:** редуцируем отдельно все  $N_i$  (обычно слева направо) до нормальной формы  $N'_i$ , затем сокращаем редекс  $(\lambda x. M) N'_1$ .

Пример: дерево для терма  $((\lambda x. M) N_1) N_2$



- Узлы  $@$  задают аппликацию, «узлы»  $\lambda$  — абстракцию.
- Узлы  $@$  могут задавать редекс ( $@$ ) или нет ( $@$ ).
- В первом случае при поиске редекса — кандидата на сокращение есть три варианта (нашли, влево, вправо), во втором — два (влево, вправо).

## Теорема

Лямбда-терм может иметь одну из двух форм:

$$\lambda \vec{x}. y \vec{N} \equiv \lambda x_1 \dots x_n. y N_1 \dots N_k, \quad n \geq 0, k \geq 0$$

$$\lambda \vec{x}. (\lambda z. M) \vec{N} \equiv \lambda x_1 \dots x_n. (\lambda z. M) N_1 \dots N_k, \quad n \geq 0, k > 0$$

## Определение

Первая форма называется *головной нормальной формой* (HNF). Переменная  $y$  называется *головной переменной*, а редекс  $(\lambda z. M) N_1$  — *головным редексом*.

## Определение

*Слабая головная нормальная форма* (WHNF) — это HNF или лямбда-абстракция, то есть не редекс на верхнем уровне.

Синтаксические категории:

- Нормальные формы  $NF ::= \lambda x. NF \mid NANF$
- Нормальные формы (не абстракции)  $NANF ::= v \mid NANF NF$
- Не абстракции  $NA = v \mid PQ$

Операционная семантика *нормальной* стратегии:

$$\frac{NA \rightarrow NA'}{NA N \rightarrow NA' N} \quad (\text{Аппл1})$$

$$\frac{N \rightarrow N'}{NANF N \rightarrow NANF N'} \quad (\text{Аппл2})$$

$$\frac{M \rightarrow M'}{\lambda x. M \rightarrow \lambda x. M} \quad (\text{Абстр})$$

$$(\lambda x. M) N \rightarrow M[x := N] \quad (\text{Редук})$$

Нормальная стратегия всегда сокращает самый левый внешний редекс (leftmost outermost).



# Операционная семантика аппликативной стратегии

Синтаксические категории:

- Нормальные формы  $NF ::= \lambda x. NF \mid NANF$
- Нормальные формы (не абстракции)  $NANF ::= v \mid NANF NF$
- Не абстракции  $NA = v \mid PQ$

Операционная семантика *аппликативной* стратегии:

$$\frac{NA \rightarrow NA'}{NA N \rightarrow NA' N} \quad (\text{Аппл1})$$

$$\frac{N \rightarrow N'}{NANF N \rightarrow NANF N'} \quad (\text{Аппл2})$$

$$\frac{M \rightarrow M'}{\lambda x. M \rightarrow \lambda x. M} \quad (\text{Абстр})$$

$$(\lambda x. M) NF \rightarrow M[x := NF] \quad (\text{Редук})$$

$$\frac{N \rightarrow N'}{(\lambda x. M) N \rightarrow (\lambda x. M) N'} \quad (\text{Аппл3})$$

Аппликативная стратегия сокращает самый левый внутренний редекс (leftmost innermost).

## Аппликативная vs нормальная стратегии

$$KI\Omega \equiv KI((\lambda x. x x) (\lambda x. x x)) \rightarrow_{\beta} \dots$$

$$KI\Omega \equiv (\lambda x y. x) I \Omega \rightarrow_{\beta} I$$

## Теорема о нормализации

Если терм  $M$  имеет нормальную форму, то последовательное сокращение самого левого внешнего редекса приводит к этой нормальной форме.

- То есть **нормальная стратегия** гарантированно нормализует нормализуемое.
- Можем доказывать отсутствие NF. Например,  $K \Omega I$ .

Недостаток нормальной стратегии — возможная неэффективность, достоинство — не считает ничего «лишнего».

## Нормальная vs аппликативная стратегии

- Пусть  $N$  — «большой» терм

$$(\lambda x. F x (G x) x) N \rightarrow_{\beta} F N (G N) N$$

В процессе дальнейших редукций редексы в  $N$  придётся сокращать три раза.

- Зато в

$$(\lambda x y. y) N \rightarrow_{\beta} \lambda y. y$$

нормальная стратегия не вычисляет  $N$  ни разу.

- Аппликативная стратегия в обоих примерах вычислит  $N$  один раз.

- Аппликативная стратегия похожа на стратегию вычислений («энергичную», eager) большинства языков программирования. Сначала вычисляются аргументы, затем происходит применение функции.
- Нормальная стратегия похожа на способ вычисления в «ленивых» (lazy) языках (Haskell, Clean).
- Для решения проблем с эффективностью в «ленивых» языках используют *механизм разделения* (через вычисления в контекстах или через редукцию на графах).

- Нет необходимости всегда доводить редукцию до NF. На практике часто ограничиваются WHNF.
- Это позволяет избежать захвата переменной в замкнутом терме (почему?)
- При наличии констант (в расширенных системах) понятие WHNF (и HNF) дополняют частично применёнными константными функциями, например

**and true**

поскольку его можно записать в  $\eta$ -эквивалентном WHNF-виде

$\lambda x. \text{and true } x$

- **Механизм вызова** — термин, применяемый при исследовании высокоуровневых языков программирования.
- В функциональных языках:
  - «вызов по значению» — аппликативный порядок редукций до WHNF;
  - «вызов по имени» — нормальный порядок редукций до WHNF;
  - «вызов по необходимости» — «вызов по имени» плюс разделение.