Applied Type Theory

Лекция 2: λ-исчисление, редукции и рекурсия

Воронов Михаил Сергеевич

ВМК МГУ

Осень 2025

План лекции

1. Нормальные формы и стратегии редукции
2. Чёрч–Россер и конфлюэнтность
3. Неподвижные точки и рекурсия: комбинаторы Y и Z

План лекции

1. Нормальные формы и стратегии редукции
2. Чёрч–Россер и конфлюэнтность
3. Неподвижные точки и рекурсия: комбинаторы Y и Z

Нормальная форма: определения и интуиция

Определение (Нормальная форма (NF))

Терм находится в NF, если в нём нет ни одного β-редекса.

Интуиция

Нормальная форма — это «результат вычисления» терма; если она существует, вычисление можно считать завершённым.

- **Критерий завершения:** достижение NF означает, что редуцировать больше нечего.
- **Уникальность результата:** в конфлюэнтных системах NF единственна, порядок редукций не важен.
- **Выбор стратегии:** знание о NF позволяет говорить о нормализующих стратегиях (normal order).
- **Доказательства терминации:** свойства SN/WN и наличие NF — центральны для типизированных систем.

Вопрос: у всех ли термов есть NF?

- Как вы думаете, любой ли терм $M \in \Lambda$ можно довести до NF?
- Если нет — приведите идею контрпримера, что может привести к бесконечной редукции?

Не все термы имеют NF: контрпримеры

- $\Omega \equiv (\lambda x.\, xx)(\lambda x.\, xx)$ — бесконечное самоприменение
- $\Omega\, I$ — головной редекс снова ведёт к Ω
- Для многих «строгих» F терм $Y\, F$ не имеет NF из-за бесконечной редукции, в нестрогих контекстах возможны частные случаи, где значение достигается
- $(\lambda x.\, x\, x\, x)\, (\lambda x.\, x\, x\, x)$ — терм, синтаксический размер которого увеличивается при редукции

Общий принцип

Если в терме есть «самоприменение» или циклическая зависимость, редукция может не завершиться.

Сильная нормализация (SN) и сходимость

Определение (Сильная нормализация)

Терм M *сильно нормализуем* (SN), если не существует бесконечной β-редукции $M = M_0 \to_\beta M_1 \to_\beta M_2 \to_\beta \cdots$.

Замечания

- (эквивалентное определение) SN $\Rightarrow$ любая стратегия редукции завершается (возможны разные длины, но нет бесконечных цепочек).
- В нетипизированном λ-исчислении термы могут *не* иметь SN (напр., Ω).
- В просто типизированном $\lambda_\to$ все термы SN. Это пролог к следующей лекции (прогресс/сохранение, терминация).

Слабая нормализация (WN)

Определение (Слабая нормализация)

Терм M *слабо нормализуем* (WN), если существует конечная β-редукция $M \twoheadrightarrow_\beta N$, где N — NF.

Замечания

- Для WN достаточно существования *хотя бы одной* завершающейся последовательности, другие стратегии могут зациклиться.
- SN $\subseteq$ WN, но не наоборот: $(\lambda x.\, 1)\, \Omega$ — WN, но не SN.

Головная редукция и головные контексты

Определение (Головной контекст)

Головные контексты заданы грамматикой: $H ::= [\,] \mid H\, M$.

Определение (Слабая головная редукция)

$M \to_h N$, если $M \equiv H[(\lambda x.\, P)\, Q]$ и $N \equiv H\big[P[x := Q]\big]$ для некоторого головного контекста H. Редукция не заходит под λ и не спускается в аргументы.

Определение (Сильная головная редукция)

Для полной (сильной) головной редукции разрешают спуск под внешние λ:
$H ::= [\,] \mid H\, M \mid \lambda x.\, H$.

Примеры: головная редукция

Пример

Пример редукции вдоль головного пути (используем контекст $H = [\,]\,y$):

$$\begin{aligned} ((\lambda x.\, x\, x)\,(\lambda z.\, z))\, y &\to_h ((\lambda z.\, z)\,(\lambda z.\, z))\, y \\ &\to_h (\lambda z.\, z)\, y \\ &\to_h y \end{aligned}$$

- **Под λ:** в слабой head-редукции не заходим под абстракцию.

$$\lambda x.\, ((\lambda y.\, y\, y)\,(\lambda y.\, y\, y)) \quad \not\to_h \quad \text{(head-шаг отсутствует)}$$

- **Редекс только в аргументе:** не спускаемся в аргументы до свёртки головы.

$$x\,((\lambda z.\, z\, z)\,(\lambda z.\, z\, z)) \quad \not\to_h \quad \text{(нет головного редекса)}$$

Контраст: если головной редекс есть, редуцируем только его, игнорируя редексы внутри аргументов до свёртки головы.

Нормальные формы: определения и интуиция

Определение (Головная нормальная форма (HNF))

Имеет вид $\lambda x_1 \ldots x_n.\, y\, M_1 \ldots M_k$ (где y — переменная). Нет редексов на *головной оси*. **Интуиция:** известна «голова» терма и внешние абстракции; редексы могут оставаться в аргументах.

Определение (Слабая головная НФ (WHNF))

Либо λ-абстракция $\lambda x.\, M$, либо $y\, M_1 \ldots M_k$ *после удаления внешних* λ*-связок*; по *головному пути* β-редексов нет. **Интуиция:** вычислено ровно столько, чтобы узнать форму значения (что стоит «в голове»); внутри аргументов и в λ-блоке редукция не происходит.

Иерархия

NF $\subseteq$ HNF $\subseteq$ WHNF, но не наоборот.

HNF vs WHNF?

- **WHNF**: достаточно, чтобы исходный терм был $\lambda x.\, M$ или имел вид $y\, M_1 \ldots M_k$; внутрь аргументов не заходим.
- **HNF**: снимаем все внешние λ и требуем, чтобы в голове стояла *переменная* y, т.е. форма $\lambda x_1 \ldots x_n.\, y\, M_1 \ldots M_k$.

Зачем это нужно?

- В ленивых ЯП достаточно WHNF для сопоставления с образцом; знать «что в голове» уже довольно.
- HNF сильнее: гарантирует отсутствие редексов на головном пути, полезно в теоремах о стандартизации и нормализации.

Примеры классификации

Терм	NF	HNF	WHNF
$I \equiv \lambda x.\, x$	$\checkmark$	$\checkmark$	$\checkmark$
SKK (до редукции)	$\times$	$\times$	$\times$
$x\,(I\,y)$	$\times$	$\checkmark$	$\checkmark$
$\lambda x.\, y\,(I\,x)$	$\times$	$\checkmark$	$\checkmark$

Оговорка

В таблице S, K понимаются как макро-развёртывания в λ-термы, поэтому $S\,K\,K$ до редукции не является ни HNF, ни WHNF.

Классифицируйте термы по типам нормальных форм (NF/HNF/WHNF):

1. $x\,(I\,y)$
2. $\lambda x.\,y\,(I\,x)$
3. $(\lambda x.\,x)\,((\lambda y.\,y)\,t)$
4. $(\lambda x.\,M)\,N$

Классификация термов: ответы

1. $x\,(I\,y)$ — HNF/WHNF, не NF (редекс внутри $I\,y$).
2. $\lambda x.\,y\,(I\,x)$ — HNF/WHNF, не NF (редекс внутри $I\,x$).
3. $(\lambda x.\,x)\,((\lambda y.\,y)\,t)$ — не HNF/WHNF (головной β-редекс); редуцируется к $((\lambda y.\,y)\,t) \to_\beta t$, после чего WHNF/HNF зависят от t.
4. $(\lambda x.\,M)\,N$ — не HNF/WHNF (головной β-редекс); форма после шага зависит от M, N.

Стратегии редукции: определения

Определение (Normal order)

Всегда редуцируем *самый левый внешний* редекс.

Определение (Call-by-name)

Как normal order, но *без* редукций под λ.

Определение (Call-by-value)

Сначала вычисляем аргументы до значений (λ-абстракций), затем выполняем β-шаг.

Определение (Call-by-need)

Call-by-name с мемоизацией значений (ленивость с разделением результатов).

Типы вызовов: пример square

$$\text{square}(n) \equiv n * n$$

Call-by-name

$$\begin{aligned}\text{square}(1+1) &\to (1+1)*(1+1)\\ &\to 2*(1+1)\\ &\to 2*2\\ &\to 4\end{aligned}$$

Call-by-value

$$\begin{aligned}\text{square}(1+1) &\to \text{square}(2)\\ &\to 2*2\\ &\to 4\end{aligned}$$

Call-by-need

$$\begin{aligned}\text{square}(1+1) &\to \text{let } x = 1+1 \text{ in } x\\ &\to \text{let } x = 2 \text{ in } x*x\\ &\to 2*2\\ &\to 4\end{aligned}$$

Замечание

Пример со square иллюстрирует поведение CBN/CBV/need с «операциями» как чёрными ящиками (δ-редукции), а не чистую β-редукцию.

Ключевые отличия в стратегиях редукции

Стратегия	**Под λ**	**Арг-ты до знач.**	**Мемоизация**	**Свойства**
Normal order	да	нет	нет	Нормализующая (если есть NF)
Call-by-name (CBN)	нет	нет	нет	Ленивость без сохранения
Call-by-value (CBV)	нет	да	нет	Строгая; зацикливает Y
Call-by-need	нет	нет	да	Ленивость с разделением (Haskell)

Нормальный порядок — нормализующая стратегия для β: $(\lambda x.\,1)\,\Omega$

Normal order

$$(\lambda x.\,1)\,\Omega \to_\beta 1$$

Внешний левейший редекс сворачивается, аргумент не вычисляется.

Call-by-value (Applicative)

$$(\lambda x.\,1)\,\Omega \text{ требует вычислить } \Omega$$

$$\Omega \to_\beta \Omega \;\to_\beta\; \cdots \text{ (зацикливание)}$$

Сначала вычисляется аргумент до значения, получаем бесконечную редукцию.

Вывод

Normal order — нормализующая стратегия: если NF существует, она будет найдена, а аппликативная стратегия может зациклиться.

Какая стратегия сойдётся?

1. $(\lambda x.\, 1)\,\Omega$.
2. $(\lambda f.\, f\, I)\,(\lambda y.\, \Omega)$.
3. $(\lambda x.\, x\, x)(\lambda x.\, x\, x)$.

Какая стратегия сойдётся?

1. $(\lambda x.\, 1)\, \Omega$ — сойдётся при CBN/normal order, нет при CBV.
2. $(\lambda f.\, f\, I)\, (\lambda y.\, \Omega)$ — **не сойдётся ни при одной стратегии**:
   $(\lambda f.\, f\, I)\, (\lambda y.\, \Omega) \;\to_\beta\; (\lambda y.\, \Omega)\, I \;\to_\beta\; \Omega.$
3. $(\lambda x.\, x\, x)(\lambda x.\, x\, x)$ — не сойдётся ни при одной стратегии.

План лекции

1. Нормальные формы и стратегии редукции
2. Чёрч–Россер и конфлюэнтность
3. Неподвижные точки и рекурсия: комбинаторы Y и Z

Теорема Чёрча–Россера (конфлюэнтность)

Теорема

Если $M \twoheadrightarrow_\beta N$ *и* $M \twoheadrightarrow_\beta L$, *то существует* $K \in \Lambda$ *такое, что*

$$N \twoheadrightarrow_\beta K \quad \text{и} \quad L \twoheadrightarrow_\beta K.$$

$$\begin{tikzcd}
 & M \arrow[dl, "\twoheadrightarrow_\beta"'] \arrow[dr, "\twoheadrightarrow_\beta"] & \\
N \arrow[dr, "\twoheadrightarrow_\beta"'] & & L \arrow[dl, "\twoheadrightarrow_\beta"] \\
 & K &
\end{tikzcd}$$

Идея доказательства: параллельная редукция $\Rightarrow_\beta$ и «ромб»

Определение. Параллельная редукция $\Rightarrow_\beta$ задаётся индукцией:

$$x \Rightarrow_\beta x$$

$$\lambda x.M \Rightarrow_\beta \lambda x.N \quad \text{если } M \Rightarrow_\beta N$$

$$M_1\, M_2 \Rightarrow_\beta N_1\, N_2 \quad \text{если } M_i \Rightarrow_\beta N_i$$

$$(\lambda x.M)\, N \Rightarrow_\beta M'[x := N'] \quad \text{если } M \Rightarrow_\beta M',\ N \Rightarrow_\beta N' \text{ (подстановка без захвата; при необходи}$$

Лемма (ромб для $\Rightarrow_\beta$)

Если $M \Rightarrow_\beta N$ и $M \Rightarrow_\beta L$, то существует K такое, что $N \Rightarrow_\beta K$ и $L \Rightarrow_\beta K$.

Связь с $\to_\beta$: каждый шаг моделируется параллельным; далее переносим «ромб» на $\twoheadrightarrow_\beta$.

Существование общего редукта

Лемма о существовании общего редукта

Для любых $M_1, M_2 \in \Lambda$, если $M_1 =_\beta M_2$, то существует общий редукт L такой, что $M_1 \twoheadrightarrow_\beta L$ и $M_2 \twoheadrightarrow_\beta L$.

Доказательство.

Доказательство индукцией по способам генерации эквивалентности; возникшие «ромбы» спрямляются с помощью теоремы Чёрча–Россера. □

Редуцируемость к нормальной форме

Лемма о редуцируемости к β-NF

Если терм $M \in \Lambda$ имеет N в качестве β-NF, то M можно свести к ней: $M \twoheadrightarrow_\beta N$.

Доказательство.

Пусть $M =_\beta N$, где N находится в β-NF. По лемме о существовании общего редукта существует терм L такой, что $M \twoheadrightarrow_\beta L$ и $N \twoheadrightarrow_\beta L$. Так как в N отсутствуют редексы, имеем $N \equiv L$. Следовательно, $M \twoheadrightarrow_\beta N$. $\square$

Единственность нормальной формы

Следствие

Если NF *терма* M *существует, то она единственна (с точностью до* α*-эквивалентности).*

Доказательство.

От противного: пусть M имеет N_1 и N_2 в качестве β-NF. Тогда $N_1 =_\beta M =_\beta N_2$. По теореме Чёрча–Россера существует $L \in \Lambda$ такое, что $N_1 \twoheadrightarrow_\beta L$ и $N_2 \twoheadrightarrow_\beta L$. По лемме о редуцируемости к β-NF получаем $N_1 \equiv L \equiv N_2$. $\square$

Общий редукт и редуцируемость к NF

- **Семантически безопасные переписывания:** замена подтермов на β-эквивалентные сохраняет результат нормализации (сходимость к общему редукту).
- **Корректность оптимизаций:** β-свёртки/раскрытия и inlining не меняют NF, если она существует.
- **Эквивалентность программ:** если $M =_\beta N$, при нормализации они сходятся к одному результату; полезно для тестов/рефакторинга.
- **Инструменты:** тактики переписывания в Coq/Agda по β-равенству безопасны для смысла термов.

Теорема о стандартизации

Теорема

Если $M \to_\beta^* N$, *то существует стандартная (leftmost-outermost) редукция из* M *в* N.

Идея.

Перестановками коммутирующих шагов двигаем внешние левейшие свёртки вперёд. Индукция по длине редукции и структуре терма. □

Почему важна стандартизация?

- **Движок переписываний:** достаточно реализовать leftmost-outermost шаги, чтобы не терять достижимость целевого терма.
- **Нормализуемость normal order:** если есть NF, стратегия её найдёт — практическая база для «ленивых» интерпретаторов.
- **Детерминированные эвристики:** при доказательствах/оптимизациях можно фиксировать порядок свёрток без риска «пропустить» NF.
- **Отладка и трассировка:** стандартные редукции дают воспроизводимые траектории, полезно для объяснимости.

Нормальный порядок — нормализующая стратегия для β

Теорема

Если у терма M существует NF, *то стратегия нормального порядка её найдёт.*

Идея.

Следует из теоремы о стандартизации. Нормальный порядок не вычисляет аргументы преждевременно, сохраняя сходимость к NF. $\square$

Ещё следствия Чёрча–Россера

- **Продолжимость до** NF**:** если $M \twoheadrightarrow_\beta N$ и N — NF, то *любая* цепочка редукций от M может быть продолжена до N.
- HNF/WHNF**:** в общем случае HNF и WHNF не обязаны быть уникальными; уникальна именно полная β-нормальная форма (если существует).

План лекции

Неподвижная точка функции

Определение

Терм X называется *неподвижной точкой* терма F, если $F\,X =_\beta X$.

Интуиция

В обычном анализе это точка пересечения графиков $y = f(x)$ и $y = x$. В λ-исчислении неподвижные точки позволяют определять рекурсию без явного именования.

Теорема о неподвижной точке

Теорема

Для любого терма $F \in \Lambda$ существует неподвижная точка:

$$\forall F \in \Lambda \ \exists X \in \Lambda : \ F\,X =_\beta X$$

Доказательство.

Возьмём $W \equiv \lambda x.\,F\,(x\,x)$ и $X \equiv W\,W$. Тогда

$$X \equiv W\,W \ \to_\beta \ F\,(W\,W) = F\,X.$$

$\square$

- Ключевая идея: самоприменение создаёт «петлю» рекурсии.
- Следствие: в λ-исчислении любая рекурсия выражается *анонимно*.

Равномерная теорема о неподвижной точке

Теорема

Существует терм $Y \in \Lambda$ такой, что для любого $F \in \Lambda$ $\ Y\,F$ — неподвижная точка F:

$$\exists Y \in \Lambda\ \ \forall F \in \Lambda :\ \ F\,(Y\,F) =_\beta Y\,F$$

Доказательство.

Пусть $Y \equiv \lambda f.\,(\lambda x.\,f(x\,x))\,(\lambda x.\,f(x\,x))$. Тогда при любом F:
$Y\,F \twoheadrightarrow_\beta F((\lambda x.\,F(x\,x))\,(\lambda x.\,F(x\,x))) \equiv F\,(Y\,F)$. $\square$

Комбинатор Карри Y

Определение (Комбинатор Карри Y)

$$Y \equiv \lambda f.\, (\lambda x.\, f(x\,x))\, (\lambda x.\, f(x\,x))$$

Основное свойство

$Y\,F \twoheadrightarrow_\beta F(\,Y\,F\,)$ и $Y\,F =_\beta F(\,Y\,F\,)$

Важный нюанс

Редукция $YF \twoheadrightarrow_\beta F(YF)$ однонаправленна. Обратное верно лишь как эквивалентность $=_\beta$.
При CBV Y зацикливается, поэтому используют Z.

CBV-совместимый вариант: Z-комбинатор

При стратегии CBV простое раскрытие комбинатора Y приводит к зацикливанию. Требуется модифицированный вариант:

$$Z \equiv \lambda f.\,(\lambda x.\,f\,(\lambda v.\,x\,x\,v))\,(\lambda x.\,f\,(\lambda v.\,x\,x\,v))$$

Ключевое свойство (CBV)

При CBV: $\quad Z\,F \;\to_\beta\; F\,(\lambda v.\,Z\,F\,v)$ (с точностью до α-экв.).

Шаги CBV.

$$\begin{aligned}
Z\,F &= (\lambda f.\,(\lambda x.\,F(\lambda v.\,x\,x\,v))\,(\lambda x.\,F(\lambda v.\,x\,x\,v)))\,F \\
&\to_\beta (\lambda x.\,F(\lambda v.\,x\,x\,v))\,(\lambda x.\,F(\lambda v.\,x\,x\,v)) \\
&\to_\beta F\big(\lambda v.\,(\lambda x.\,F(\lambda v.\,x\,x\,v))\,(\lambda x.\,F(\lambda v.\,x\,x\,v))\,v\big) \\
&= F\big(\lambda v.\,Z\,F\,v\big).
\end{aligned}$$

Почему Y зацикливает при CBV

При CBV аргумент редуцируется до значения перед подстановкой. В терме $Y\,F$ внутренняя структура заставляет вычислять самоприменение до подстановки:

$$Y \equiv \lambda f.\,(\lambda x.\,f(x\,x))\,(\lambda x.\,f(x\,x))$$

Применяя CBV к $Y\,F$, нужно сначала привести к значению аргумент $(\lambda x.\,F(x\,x))$, что ведёт к бесконечному раскрытию самоприменения. Поэтому используют Z.

Как писать рекурсию через Y

Цель: определить факториал

$$fac \equiv \lambda n.\, \mathrm{if}\ (\mathrm{iszero}\ n)\ 1\ (\mathrm{mult}\ n\ (fac\ (\mathrm{pred}\ n)))$$

Решение через Y

1. Заменяем рекурсивное имя fac на параметр f:

$$F \equiv \lambda f.\, \lambda n.\, \mathrm{if}\ (\mathrm{iszero}\ n)\ 1\ (\mathrm{mult}\ n\ (f\ (\mathrm{pred}\ n)))$$

2. Тогда $fac \equiv Y\ F$

Демонстрация: редукции $\mathit{fac}\,3$

Пример

$$
\begin{aligned}
\mathit{fac}\,3 &\equiv (YF)\,3 && (1)\\
&\twoheadrightarrow_\beta F(YF)\,3 && (2)\\
&\equiv (\lambda n.\,\text{if (iszero } n\text{) 1 (mult } n \text{ ((}YF\text{) (pred } n\text{))))}\,3 && (3)\\
&\to_\beta \text{if (iszero 3) 1 (mult 3 ((}YF\text{)(pred 3)))} && (4)\\
&\to_\beta \text{mult 3 ((}YF\text{) 2)} \quad \text{(рекурсивный вызов)} && (5)
\end{aligned}
$$

Ключевое наблюдение

Комбинатор Y раскрывается однократно, далее рекурсия происходит через параметр f.

Вопрос: YF и направленность редукции

Почему $YF \not\twoheadrightarrow_\beta F(YF) \twoheadrightarrow_\beta YF$ в обе стороны, несмотря на то что $YF =_\beta F(YF)$? Приведите идею, объясняющую асимметрию направленной редукции.

Термовые уравнения

Схема β-редукции $(\lambda x.\, M)\, N \to_\beta M[x := N]$ позволяет решать простые уравнения на термы. Например, найти F такое, что $F\, M\, N\, L = M\, L\, (N\, L)$ для всех M, N, L:

- $F\, M\, N = \lambda l.\, M\, l\, (N\, l)$
- $F\, M = \lambda n\, l.\, M\, l\, (n\, l)$
- $F = \lambda m\, n\, l.\, m\, l\, (n\, l)$

Ограничение

Такой метод не работает для *рекурсивных* уравнений вида $X = F\, X$. Здесь требуется комбинатор неподвижной точки.

Решение рекурсивных уравнений через комбинатор Y

Цель: решить уравнение на терм $X = F\,X$.

Идея

Полагаем $X \equiv Y\,F$, где $Y \equiv \lambda f.\,(\lambda x.\,f(x\,x))\,(\lambda x.\,f(x\,x))$. Тогда

$$Y\,F \twoheadrightarrow_\beta F\big((\lambda x.\,F(x\,x))\,(\lambda x.\,F(x\,x))\big) \equiv F(Y\,F),$$

то есть $F(Y\,F) =_\beta Y\,F$, и X действительно неподвижная точка F.

Шаблон

Для рекурсивной функции строим $F \equiv \lambda f.\,\ldots f\,\ldots$ и полагаем $\mathrm{name} \equiv Y\,F$.

При CBV используется Z-комбинатор.