

Функциональное программирование

Лекция 3. Просто типизированное лямбда-исчисление

Денис Николаевич Москвин

СПбАУ РАН

19.09.2017

- 1 Понятие типа
- 2 Просто типизированное λ -исчисление
- 3 Формализм систем λ_{\rightarrow}
- 4 Свойства λ_{\rightarrow}

- 1 Понятие типа
- 2 Просто типизированное λ -исчисление
- 3 Формализм систем λ_{\rightarrow}
- 4 Свойства λ_{\rightarrow}

Что такое типы?

Система типов — это гибко управляемый синтаксический метод доказательства отсутствия в программе определенных видов поведения при помощи классификации выражений языка по разновидностям вычисляемых ими значений.

Бенджамин Пирс

- В λ -исчислении:
 - выражения — λ -термы;
 - вычисление — их редукция;
 - значения — $(\text{WN})\text{NF}$.
- Типы — *синтаксические* конструкции, приписываемые термам по определённым правилам:

$M:\sigma$

Для чего нужны типы?

- Типы дают частичную спецификацию

$$f:\mathbb{N} \rightarrow \mathbb{N} \quad g:(\forall n:\mathbb{N}. \exists m:\mathbb{N}. m > n)$$

- Правильно типизированные программы не могут «сломаться». Робин Милнер (1978)

$$M:\sigma \wedge M \rightarrow v \Rightarrow v:\sigma$$

- Типизированные программы всегда завершаются.
(это не всегда так :)
- Проверка типов отлавливает простые ошибки.

Какие бывают системы типов?

Возможны классификации систем типов по разным аспектам:

- Статические (static) vs динамические (dynamic);
- Явные (explicit) vs неявные (implicit);
- Сильные (strong) vs слабые (weak).

Пример слабой системы:

```
x = 5;  
y = "37";  
z = x + y;
```

- При императивном подходе типы «естественно» возникают из интерпретации различных состояний.
- При функциональном подходе типы «естественно» возникают из анализа арностей функций.

- В большинстве систем типизации тождественной функции $I \equiv \lambda x. x$ может быть приписан тип $\alpha \rightarrow \alpha$

$$I: \alpha \rightarrow \alpha$$

- Если x , являющийся аргументом функции I , имеет тип α , то значение $I x$ тоже имеет тип α .
- В общем случае $\alpha \rightarrow \beta$ является типом функции из α в β .

Пример (на некотором условном языке)

```
sin : Double → Double
```

```
length : Array → Int
```

При типизации λ -исчисления выделяют два семейства систем типов.

Системы в стиле Карри

Термы те же, что и в бестиповой теории. Каждый терм обладает множеством различных типов (пустое, одно- или многоэлементное, бесконечное).

Системы в стиле Чёрча

Термы — аннотированные версии бестиповых термов. Каждый терм имеет тип (обычно уникальный), выводимый из способа, которым терм аннотирован.

Подход программиста

Термы интерпретируются как программы, а типы — как их частичные спецификации.

- Системы в стиле Карри: неявная типизация (например, Haskell, Ocaml).
- Системы в стиле Чёрча: явная типизация (большинство типизированных языков).

Логический подход

Типы интерпретируются как высказывания, а термы — как их доказательства.

Связь между «вычислительными» и логическими системами называют *соответствием Карри-Говарда*.

- 1 Понятие типа
- 2 Просто типизированное λ -исчисление
- 3 Формализм систем λ_{\rightarrow}
- 4 Свойства λ_{\rightarrow}

Просто типизированное λ -исчисление

Самая простая система — это *просто типизированное λ -исчисление* (λ_{\rightarrow} или Simple Type Theory (STT)).

Определение

Множество типов \mathbb{T} системы λ_{\rightarrow} определяется индуктивно:

$\alpha, \beta, \dots \in \mathbb{T}$ (переменные типа)

$\sigma, \tau \in \mathbb{T} \Rightarrow (\sigma \rightarrow \tau) \in \mathbb{T}$ (типы пространства функций)

- В абстрактном синтаксисе:

$$\mathbb{T} ::= \mathbb{V} \mid \mathbb{T} \rightarrow \mathbb{T}$$

Здесь $\mathbb{V} = \{\alpha, \beta, \dots\}$ — множество типовых переменных.

- Соглашение: α, β, γ используем для типовых переменных, а σ, τ, ρ — для произвольных типов.

Стрелка *правоассоциативна*: если $\sigma_1, \dots, \sigma_n \in \mathbb{T}$, то

$$\sigma_1 \rightarrow \sigma_2 \rightarrow \dots \rightarrow \sigma_n \equiv (\sigma_1 \rightarrow (\sigma_2 \rightarrow \dots \rightarrow (\sigma_{n-1} \rightarrow \sigma_n) \dots))$$

Примеры типов

$$(\alpha \rightarrow \beta) \equiv \alpha \rightarrow \beta$$

$$(\alpha \rightarrow (\beta \rightarrow \gamma)) \equiv \alpha \rightarrow \beta \rightarrow \gamma$$

$$((\alpha \rightarrow \beta) \rightarrow \gamma) \equiv (\alpha \rightarrow \beta) \rightarrow \gamma$$

$$((\alpha \rightarrow \beta) \rightarrow ((\beta \rightarrow \gamma) \rightarrow (\alpha \rightarrow \gamma))) \equiv (\alpha \rightarrow \beta) \rightarrow (\beta \rightarrow \gamma) \rightarrow \alpha \rightarrow \gamma$$

$$((\alpha \rightarrow \beta) \rightarrow (((\alpha \rightarrow \beta) \rightarrow \beta) \rightarrow \beta)) \equiv (\alpha \rightarrow \beta) \rightarrow ((\alpha \rightarrow \beta) \rightarrow \beta) \rightarrow \beta$$

Всякий тип в λ_{\rightarrow} может быть записан в виде

$$\sigma_1 \rightarrow \sigma_2 \rightarrow \dots \rightarrow \sigma_n \rightarrow \alpha$$

Как типизировать термы? (переменные и аппликация)

- Если терм *переменная* — как угодно:
 $x:\alpha, y:\alpha \rightarrow \beta, z:(\alpha \rightarrow \beta) \rightarrow ((\alpha \rightarrow \beta) \rightarrow \beta) \rightarrow \beta.$
- Если терм *аппликация* $M N$, то
 - M должно быть функцией, то есть иметь стрелочный тип $M:\sigma \rightarrow \tau$;
 - N должно быть «подходящим» аргументом, то есть иметь тип $N:\sigma$;
 - вся аппликация должна иметь тип результата применения функции: $(M N):\tau.$

Примеры «типизаций»

Для $x:\alpha, y:\alpha \rightarrow \beta$ имеем $(y x):\beta.$

Добавив $z:\beta \rightarrow \gamma$, получим $z (y x):\gamma.$

А какие должны иметь типы x и y , чтобы $x (y x):\gamma?$

Как типизировать термы? (абстракция)

- Если терм **абстракция** $\lambda x. M$, то его тип должен быть стрелочным $(\lambda x. M): \sigma \rightarrow \tau$, причём тип аргумента должен быть $x: \sigma$, а тип тела абстракции — $M: \tau$.

Пример «типизации»

Для $x: \alpha$ имеем $(\lambda x. x): \alpha \rightarrow \alpha$.

- А надо ли здесь отдельно указывать, что $x: \alpha$?
 - Если не указать, то допустимо и $(\lambda x. x): \beta \rightarrow \beta$ и даже $(\lambda x. x): (\alpha \rightarrow \beta) \rightarrow (\alpha \rightarrow \beta)$ — **стиль Карри**.
 - Если указать $(\lambda x: \alpha. x): \alpha \rightarrow \alpha$, то тип терма определяется однозначно — **стиль Чёрча**.

Типизируйте по Чёрчу: $\lambda x:?. \lambda y:?. x (y x):?$

Как типизировать термы? (ассоциативность)

Правила ассоциативности для типов (вправо), аппликации (влево) и абстракции (вправо) хорошо согласованы друг с другом.

В предположении $M:\alpha, N:\beta, P:\gamma, Q:\rho$

$$F:\alpha \rightarrow (\beta \rightarrow (\gamma \rightarrow \delta))$$

$$(FM):\beta \rightarrow (\gamma \rightarrow \delta)$$

$$((FM)N):\gamma \rightarrow \delta$$

$$(((FM)N)P):\delta$$

$$(\lambda y:\tau. Q):\tau \rightarrow \rho$$

$$(\lambda x:\sigma. (\lambda y:\tau. Q)):\sigma \rightarrow (\tau \rightarrow \rho)$$

Зелёные скобки опускаются.

- 1 Понятие типа
- 2 Просто типизированное λ -исчисление
- 3 Формализм систем λ_{\rightarrow}**
- 4 Свойства λ_{\rightarrow}

Определение

Множество *предтермов* (или *псевдотермов*) Λ строится из переменных из $V = \{x, y, z, \dots\}$ с помощью аппликации и абстракции:

$$x \in V \Rightarrow x \in \Lambda$$

$$M, N \in \Lambda \Rightarrow (MN) \in \Lambda$$

$$M \in \Lambda, x \in V \Rightarrow (\lambda x. M) \in \Lambda$$

- В абстрактном синтаксисе

$$\Lambda ::= V \mid (\Lambda \Lambda) \mid (\lambda V. \Lambda)$$

- То есть предтермы системы в стиле Карри — это в точности термы бестипового λ -исчисления.

Определение

Множество *предтермов* $\Lambda_{\mathbb{T}}$ строится из переменных из $V = \{x, y, z, \dots\}$ с помощью аппликации и **аннотированной типами абстракции**:

$$x \in V \Rightarrow x \in \Lambda_{\mathbb{T}}$$

$$M, N \in \Lambda_{\mathbb{T}} \Rightarrow (MN) \in \Lambda_{\mathbb{T}}$$

$$M \in \Lambda_{\mathbb{T}}, x \in V, \sigma \in \mathbb{T} \Rightarrow (\lambda x : \sigma. M) \in \Lambda_{\mathbb{T}}$$

- В абстрактном синтаксисе

$$\Lambda_{\mathbb{T}} ::= V \mid (\Lambda_{\mathbb{T}} \Lambda_{\mathbb{T}}) \mid (\lambda V : \mathbb{T}. \Lambda_{\mathbb{T}})$$

- Все соглашения о скобках и ассоциативности те же, что и в системе Λ .

Система λ_{\rightarrow} а ля Карри:

$\lambda x y. x$

$\lambda f g x. f (g x)$

$\lambda x. x x$

Система λ_{\rightarrow} а ля Чёрч:

$\lambda x:\alpha. \lambda y:\beta. x \equiv \lambda x^\alpha y^\beta. x$

$\lambda x:\alpha. \lambda y:\alpha. x \equiv \lambda x^\alpha y^\alpha. x$

$\lambda f:\alpha. \lambda g:\beta. \lambda x:\gamma. f (g x) \equiv \lambda f^\alpha g^\beta x^\gamma. f (g x)$

$\lambda x:\alpha. x x \equiv \lambda x^\alpha. x x$

Определение

Утверждение (о типизации) в λ_{\rightarrow} «а ля Карри» имеет вид

$$M:\tau$$

где $M \in \Lambda$ и $\tau \in \mathbb{T}$. Тип τ иногда называют *предикатом*, а терм M — *субъектом* утверждения.

Для λ_{\rightarrow} «а ля Чёрч» надо лишь заменить Λ на $\Lambda_{\mathbb{T}}$.

Примеры утверждений о типизации

Система в стиле Карри

$$(\lambda x. x):\alpha \rightarrow \alpha$$

$$(\lambda x. x):(\alpha \rightarrow \beta) \rightarrow \alpha \rightarrow \beta$$

$$(\lambda x y. x):\alpha \rightarrow \beta \rightarrow \alpha$$

Система в стиле Чёрча

$$(\lambda x^{\alpha}. x):\alpha \rightarrow \alpha$$

$$(\lambda x^{\alpha \rightarrow \beta}. x):(\alpha \rightarrow \beta) \rightarrow \alpha \rightarrow \beta$$

$$(\lambda x^{\alpha} y^{\beta}. x):\alpha \rightarrow \beta \rightarrow \alpha$$

Определение

Объявление — это утверждение (о типизации) с термовой переменной в качестве субъекта.

Примеры объявлений

$x:\alpha$

$y:\beta$

$f:\alpha \rightarrow \beta$

$g:(\alpha \rightarrow \beta) \rightarrow \gamma$

Определение

Контекст — это множество объявлений с *различными* переменными в качестве субъекта:

$$\Gamma = \{x_1:\sigma_1, x_2:\sigma_2, \dots, x_n:\sigma_n\}$$

Контекст иногда называют *базисом* или *окружением*.

- Фигурные скобки множества иногда опускают:

$$\Gamma = x:\alpha, y:\beta, f:\alpha \rightarrow \beta, g:(\alpha \rightarrow \beta) \rightarrow \gamma$$

- Контексты можно *расширять*, добавляя объявление *новой* переменной:

$$\Delta = \Gamma, z:\alpha \rightarrow \gamma = x:\alpha, y:\beta, f:\alpha \rightarrow \beta, g:(\alpha \rightarrow \beta) \rightarrow \gamma, z:\alpha \rightarrow \gamma$$

- Контекст можно рассматривать как (частичную) функцию из множества переменных V в множество типов \mathbb{T} .

Определение

Утверждение $M : \tau$ называется **выводимым** в контексте Γ , обозначение

$$\Gamma \vdash M : \tau$$

если его вывод может быть произведен по правилам:

$$\begin{aligned} (x:\sigma) \in \Gamma &\Rightarrow \Gamma \vdash x:\sigma \\ \Gamma \vdash M:\sigma \rightarrow \tau, \Gamma \vdash N:\sigma &\Rightarrow \Gamma \vdash (MN):\tau \\ \Gamma, x:\sigma \vdash M:\tau &\Rightarrow \Gamma \vdash (\lambda x. M):\sigma \rightarrow \tau \end{aligned}$$

Если существуют Γ и τ , такие что $\Gamma \vdash M:\tau$, то предтерм M называют **(допустимым) термом**.

(аксиома) $\Gamma \vdash x:\sigma$, если $(x:\sigma) \in \Gamma$

(удаление \rightarrow)
$$\frac{\Gamma \vdash M:\sigma \rightarrow \tau \quad \Gamma \vdash N:\sigma}{\Gamma \vdash (MN):\tau}$$

(введение \rightarrow)
$$\frac{\Gamma, x:\sigma \vdash M:\tau}{\Gamma \vdash (\lambda x. M):\sigma \rightarrow \tau}$$

Пример дерева вывода типа для $\lambda x y. x$

$$\frac{\frac{x:\alpha, y:\beta \vdash x:\alpha}{x:\alpha \vdash (\lambda y. x):\beta \rightarrow \alpha}}{\vdash (\lambda x y. x):\alpha \rightarrow \beta \rightarrow \alpha}$$

То есть для любых $\alpha, \beta \in \mathbb{T}$ верно $\vdash (\lambda x y. x):\alpha \rightarrow \beta \rightarrow \alpha$.

Типизация λ_{\rightarrow} «а ля Чёрч» в виде правил вывода

(аксиома) $\Gamma \vdash x:\sigma$, если $(x:\sigma) \in \Gamma$

(удаление \rightarrow)
$$\frac{\Gamma \vdash M:\sigma \rightarrow \tau \quad \Gamma \vdash N:\sigma}{\Gamma \vdash (MN):\tau}$$

(введение \rightarrow)
$$\frac{\Gamma, x:\sigma \vdash M:\tau}{\Gamma \vdash (\lambda x:\sigma. M):\sigma \rightarrow \tau}$$

Вывод типа для $\lambda x^{\alpha} y^{\beta}. x$ проще

$$\frac{\frac{x:\alpha, y:\beta \vdash x:\alpha}{x:\alpha \vdash (\lambda y:\beta. x):\beta \rightarrow \alpha}}{\vdash (\lambda x:\alpha. \lambda y:\beta. x):\alpha \rightarrow \beta \rightarrow \alpha}$$

То есть для **каждых** $\alpha, \beta \in \mathbb{T}$ верно $\vdash (\lambda x^{\alpha} y^{\beta}. x):\alpha \rightarrow \beta \rightarrow \alpha$.

- 1 Понятие типа
- 2 Просто типизированное λ -исчисление
- 3 Формализм систем λ_{\rightarrow}
- 4 Свойства λ_{\rightarrow}

Лемма об инверсии (лемма генерации)

- $\Gamma \vdash x:\sigma \Rightarrow (x:\sigma) \in \Gamma$.
- $\Gamma \vdash (MN):\tau \Rightarrow \exists \sigma [\Gamma \vdash M:\sigma \rightarrow \tau \wedge \Gamma \vdash N:\sigma]$.
- $\Gamma \vdash (\lambda x.M):\rho \Rightarrow \exists \sigma, \tau [\Gamma, x:\sigma \vdash M:\tau \wedge \rho \equiv \sigma \rightarrow \tau]$. (λ_{\rightarrow} а ля Карри)
- $\Gamma \vdash (\lambda x:\sigma.M):\rho \Rightarrow \exists \tau [\Gamma, x:\sigma \vdash M:\tau \wedge \rho \equiv \sigma \rightarrow \tau]$. (λ_{\rightarrow} а ля Чёрч)

Лемма о типизируемости подтерма

Пусть M' — подтерм M . Тогда $\Gamma \vdash M:\sigma \Rightarrow \Gamma' \vdash M':\sigma'$ для некоторых Γ' и σ' .

То есть, если терм имеет тип, то каждый его подтерм тоже имеет тип.

Какой контекст требуется, чтобы произвести присваивание типов?

Лемма «разбавления» (Thinning)

Пусть Γ и Δ — контексты, причём $\Delta \supseteq \Gamma$. Тогда $\Gamma \vdash M:\sigma \Rightarrow \Delta \vdash M:\sigma$. *Расширение контекста не влияет на выводимость утверждения типизации.*

Лемма о свободных переменных

$\Gamma \vdash M:\sigma \Rightarrow FV(M) \subseteq \text{dom}(\Gamma)$. *Свободные переменные типизированного терма должны присутствовать в контексте.*

Лемма сужения

$\Gamma \vdash M:\sigma \Rightarrow \Gamma \upharpoonright FV(M) \vdash M:\sigma$. *Сужение контекста до множества свободных переменных терма не влияет на выводимость утверждения типизации.*

Свойства λ_{\rightarrow} : нетипизируемые предтермы

- Рассмотрим предтерм $x x$. Предположим, что это терм.
- Тогда имеются Γ и σ , такие что

$$\Gamma \vdash (x x) : \sigma$$

- По лемме об инверсии существует такой τ , что правый подтерм $x : \tau$, а левый подтерм (тоже x) имеет тип $\tau \rightarrow \sigma$.
- По лемме о контекстах $x \in \text{dom}(\Gamma)$ и должен иметь там *единственное* связывание по определению контекста. То есть $\tau = \tau \rightarrow \sigma$ — тип является подвыражением себя, чего не может быть, поскольку (*и пока*) типы конечны.

$$x : \tau \not\vdash (x x) : \sigma, \quad \not\vdash \omega : \sigma, \quad \not\vdash \Omega : \sigma, \quad \not\vdash Y : \sigma.$$

Предтермы $\omega = \lambda x. x x$, $\Omega = \omega \omega$ и $Y = \lambda f. (\lambda x. f(x x))(\lambda x. f(x x))$ не имеют типа по лемме о типизируемости подтерма.

Лемма подстановки типа для λ_{\rightarrow}

Определение

Для $\sigma, \tau \in \mathbb{T}$ *подстановку* τ вместо α в σ обозначим $[\alpha := \tau]\sigma$.

Пример

$$[\alpha := \gamma \rightarrow \gamma](\alpha \rightarrow \beta \rightarrow \alpha) = (\gamma \rightarrow \gamma) \rightarrow \beta \rightarrow \gamma \rightarrow \gamma$$

Лемма подстановки типа

$\Gamma \vdash M : \sigma \Rightarrow [\alpha := \tau]\Gamma \vdash M : [\alpha := \tau]\sigma$. (λ_{\rightarrow} а ля Карри)

$\Gamma \vdash M : \sigma \Rightarrow [\alpha := \tau]\Gamma \vdash [\alpha := \tau]M : [\alpha := \tau]\sigma$. (λ_{\rightarrow} а ля Чёрч)

Пример. Подстановка $[\alpha := \gamma \rightarrow \gamma]$:

$$\begin{aligned} x : \alpha \vdash (\lambda y^{\alpha} z^{\beta}. x) : \alpha \rightarrow \beta \rightarrow \alpha &\Rightarrow \\ x : \gamma \rightarrow \gamma \vdash (\lambda y^{\gamma \rightarrow \gamma} z^{\beta}. x) : (\gamma \rightarrow \gamma) \rightarrow \beta \rightarrow \gamma \rightarrow \gamma \end{aligned}$$

Лемма подстановки терма

Пусть $\Gamma, x:\sigma \vdash M:\tau$ и $\Gamma \vdash N:\sigma$, тогда $\Gamma \vdash M[x := N]:\tau$.

То есть, подходящая по типу *подстановка терма сохраняет тип*.

Пример

Берём терм

$$x:\gamma \rightarrow \gamma \vdash (\lambda y^{\beta}. x):\beta \rightarrow \gamma \rightarrow \gamma$$

и подставляем в него вместо свободной переменной $x:\gamma \rightarrow \gamma$ терм $\mathbf{I}_{\gamma} \equiv \lambda p^{\gamma}. p$ подходящего типа $\gamma \rightarrow \gamma$. Получаем

$$\vdash (\lambda y^{\beta} p^{\gamma}. p):\beta \rightarrow \gamma \rightarrow \gamma$$

Теорема о редукции субъекта

Пусть $M \rightarrow_{\beta} N$. Тогда $\Gamma \vdash M:\sigma \Rightarrow \Gamma \vdash N:\sigma$.

- То есть *тип терма сохраняется при β -редукциях*.
- С «вычислительной» точки зрения это одно из ключевых свойств любой системы типов.

Следствие

Множество типизируемых в λ_{\rightarrow} термов замкнуто относительно редукции.

Единственность типа в λ_{\rightarrow}

Теорема о единственности типа для λ_{\rightarrow} а ля Чёрч

Пусть $\Gamma \vdash M:\sigma$ и $\Gamma \vdash M:\tau$. Тогда $\sigma \equiv \tau$.

Терм в λ_{\rightarrow} а ля Чёрч имеет единственный тип.

Следствие

Пусть $\Gamma \vdash M:\sigma$, $\Gamma \vdash N:\tau$ и $M =_{\beta} N$. Тогда $\sigma \equiv \tau$.

Типизируемые β -конвертируемые термы имеют одинаковый тип в λ_{\rightarrow} а ля Чёрч.

Контрпример (для системы а ля Карри)

Оба типа подходят для $\mathbf{K} = \lambda x y. x$ в λ_{\rightarrow} а ля Карри:

$$\vdash (\lambda x y. x) : \alpha \rightarrow (\delta \rightarrow \gamma \rightarrow \delta) \rightarrow \alpha$$

$$\vdash (\lambda x y. x) : (\gamma \rightarrow \gamma) \rightarrow \beta \rightarrow \gamma \rightarrow \gamma$$

Для систем в стиле Карри единственности типа нет.

Связь между системами Карри и Чёрча

- Можно задать стирающее отображение $|\cdot| : \mathcal{L}_{\mathbb{T}} \rightarrow \mathcal{L}$:

$$|x| \equiv x$$

$$|MN| \equiv |M||N|$$

$$|\lambda x:\sigma. M| \equiv \lambda x. |M|$$

- Все атрибутированные типами термы из версии Чёрча $\lambda \rightarrow$ «проектируются» в термы в версии Карри:

$$M \in \mathcal{L}_{\mathbb{T}} \wedge \Gamma \vdash_{\mathbb{C}} M:\sigma \Rightarrow \Gamma \vdash_{\mathbb{K}} |M|:\sigma$$

- Термы из версии Карри $\lambda \rightarrow$ могут быть «подняты» в термы из версии Чёрча:

$$M \in \mathcal{L} \wedge \Gamma \vdash_{\mathbb{K}} M:\sigma \Rightarrow \exists N \in \mathcal{L}_{\mathbb{T}} [\Gamma \vdash_{\mathbb{C}} N:\sigma \wedge |N| \equiv M]$$

- Для произвольного типа $\sigma \in \mathbb{T}$ выполняется

$$\sigma \text{ обитаем в } \lambda \rightarrow \text{-Карри} \Leftrightarrow \sigma \text{ обитаем в } \lambda \rightarrow \text{-Чёрч}$$

- Есть ли алгоритм, который позволяют решить задачу?

$\vdash M:\sigma?$	Задача проверки типа Type Checking Problem	ЗПТ TCP
$\vdash M:?$	Задача синтеза типа Type Synthesis (or Assgnment) Problem	ЗСТ TSP, TAP
$\vdash ?:\sigma$	Задача обитаемости типа Type Inhabitation Problem	ЗОТ TIP
- Для λ_{\rightarrow} (и в стиле Чёрча, и в стиле Карри) все эти задачи разрешимы.
- ЗПТ выглядит проще ЗСТ, но в системах Карри они эквивалентны: проверка $(MN):\sigma?$ требует синтеза $N:?$.

Определение

Терм называют *слабо (weak) нормализуемым* (WN), если **существует** последовательность редукций, приводящих его к нормальной форме.

Определение

Терм называют *сильно (strong) нормализуемым* (SN), если **любая** последовательность редукций, приводит его к нормальной форме.

Примеры

Терм KIK — сильно нормализуем,
терм $KI\Omega$ — слабо нормализуем,
терм Ω — не нормализуем.

Теорема о нормализации λ_{\rightarrow}

Определение

Систему типов называют *слабо нормализуемой* если все её допустимые термы слабо нормализуемы.

Определение

Систему типов называют *сильно нормализуемой* если все её допустимые термы сильно нормализуемы.

Теорема о нормализации λ_{\rightarrow}

Обе системы λ_{\rightarrow} (и Карри, и Чёрча) *сильно нормализуемы*.

То есть любой допустимый терм в λ_{\rightarrow} всегда редуцируется к нормальной форме.