Функциональное программирование Лекция 3. Просто типизированное лямбда-исчисление

Денис Николаевич Москвин

СПбАУ

22.09.2015

План лекции

1. Понятие типа
2. Просто типизированное λ-исчисление
3. Формализм систем $\lambda_{\to}$
4. Свойства $\lambda_{\to}$

План лекции

1. Понятие типа
2. Просто типизированное λ-исчисление
3. Формализм систем $\lambda_{\to}$
4. Свойства $\lambda_{\to}$

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 — (WH)NF.
- Типы — *синтаксические* конструкции, приписываемые термам по определённым правилам:

$$M : \sigma$$

Для чего нужны типы?

- Типы дают частичную спецификацию

$$f : \mathbb{N} \to \mathbb{N} \qquad g : (\forall n : \mathbb{N}.\, \exists m : \mathbb{N}.\, m > n)$$

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

$$M : \sigma \;\wedge\; M \twoheadrightarrow v \;\Rightarrow\; v : \sigma$$

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

Какие бывают системы типов?

Возможны классификации систем типам по разным аспектам:

- Статические (static) vs динамические (dinamic);
- 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 $\mathbf{I} \equiv \lambda x.\,x$ может быть приписан тип $\alpha \to \alpha$

$$\mathbf{I} : \alpha \to \alpha$$

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

Пример (на некотором условном языке)

$$\mathtt{sin} : \mathtt{Double} \to \mathtt{Double}$$

$$\mathtt{length} : \mathtt{Array} \to \mathtt{Int}$$

Системы Карри и Чёрча

При типизации λ-исчислении 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.

Два взгляда на системы типов

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***.

План лекции

Просто типизированное λ-исчисление

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

Определение

Множество типов $\mathbb{T}$ системы $\lambda_{\to}$ определяется индуктивно:

$$\alpha, \beta, \ldots \in \mathbb{T} \qquad \text{(переменные типа)}$$

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

- В абстрактном синтаксисе:

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

  Здесь $\mathbb{V} = \{\alpha, \beta, \ldots\}$ — множество типовых переменных.
- Соглашение: α, β, γ используем для типовых переменных, а σ, τ, ρ — для произвольных типов.

Соглашения и примеры

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

$$\sigma_1 \to \sigma_2 \to \ldots \to \sigma_n \;\equiv\; (\sigma_1 \to (\sigma_2 \to \ldots \to (\sigma_{n-1} \to \sigma_n) \ldots))$$

Примеры типов

$$\begin{aligned}
(\alpha \to \beta) &\equiv \alpha \to \beta \\
(\alpha \to (\beta \to \gamma)) &\equiv \alpha \to \beta \to \gamma \\
((\alpha \to \beta) \to \gamma) &\equiv (\alpha \to \beta) \to \gamma \\
((\alpha \to \beta) \to ((\beta \to \gamma) \to (\alpha \to \gamma))) &\equiv (\alpha \to \beta) \to (\beta \to \gamma) \to \alpha \to \gamma \\
((\alpha \to \beta) \to (((\alpha \to \beta) \to \beta) \to \beta)) &\equiv (\alpha \to \beta) \to ((\alpha \to \beta) \to \beta) \to \beta
\end{aligned}$$

Всякий тип в $\lambda_\to$ может быть записан в виде

$$\sigma_1 \to \sigma_2 \to \ldots \to \sigma_n \to \alpha$$

- Если терм ***переменная*** — как угодно:
  $x\!:\!\alpha,\ y\!:\!\alpha \to \beta,\ z\!:\!(\alpha\to\beta)\to((\alpha\to\beta)\to\beta)\to\beta.$
- Если терм ***аппликация*** $M\,N$, то
  - M должно быть функцией, то есть иметь стрелочный тип $M\!:\!\sigma\to\tau$;
  - 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 $x\!:\!\alpha,\ y\!:\!\alpha \to \beta$ имеем $(y\,x)\!:\!\beta$.
Добавив $z\!:\!\beta \to \gamma$, 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 $(\lambda x.\,M):\sigma\to\tau$, причём тип аргумента должен быть $x:\sigma$, а тип тела абстракции — $M:\tau$.

Пример «типизации»

Для $x:\alpha$ имеем $(\lambda x.\,x):\alpha\to\alpha$.

- А надо ли здесь отдельно указывать, что $x:\alpha$?
  - Если не указать, то допустимо и $(\lambda x.\,x):\beta\to\beta$ и даже $(\lambda x.\,x):(\alpha\to\beta)\to(\alpha\to\beta)$ — стиль Карри.
  - Если указать $(\lambda x:\alpha.\,x):\alpha\to\alpha$, 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 $M:\alpha, N:\beta, P:\gamma, Q:\rho$

$$F:\alpha \to (\beta \to (\gamma \to \delta))$$
$$(F\,M):\beta \to (\gamma \to \delta)$$
$$((F\,M)\,N):\gamma \to \delta$$
$$(((F\,M)\,N)\,P):\delta$$

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

Зелёные скобки опускаются.

План лекции

Предтермы системы $\lambda_{\to}$ а ля Карри

Определение

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

$$\begin{array}{rcl} x \in V & \Rightarrow & x \in \Lambda \\ M, N \in \Lambda & \Rightarrow & (M\,N) \in \Lambda \\ M \in \Lambda, x \in V & \Rightarrow & (\lambda x.\, M) \in \Lambda \end{array}$$

- В абстрактном синтаксисе

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

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

Предермы системы $\lambda_{\to}$ а ля Чёрч

Определение

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

$$\begin{array}{rcl} x \in V & \Rightarrow & x \in \Lambda_{\mathbb{T}} \\ M, N \in \Lambda_{\mathbb{T}} & \Rightarrow & (M\,N) \in \Lambda_{\mathbb{T}} \\ M \in \Lambda_{\mathbb{T}}, x \in V, \sigma \in \mathbb{T} & \Rightarrow & (\lambda x\!:\!\sigma.\,M) \in \Lambda_{\mathbb{T}} \end{array}$$

- В абстрактном синтаксисе

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

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

Примеры предтермов

Система $\lambda_\rightarrow$ а ля Карри:

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

Система $\lambda_\rightarrow$ а ля Чёрч:

$$\lambda x{:}\alpha.\, \lambda y{:}\beta.\, x \;\equiv\; \lambda x^{\alpha}\, y^{\beta}.\, x$$
$$\lambda x{:}\alpha.\, \lambda y{:}\alpha.\, x \;\equiv\; \lambda x^{\alpha}\, y^{\alpha}.\, x$$
$$\lambda f{:}\alpha.\, \lambda g{:}\beta.\, \lambda x{:}\gamma.\, f\,(g\, x) \;\equiv\; \lambda f^{\alpha}\, g^{\beta}\, x^{\gamma}.\, f\,(g\, x)$$
$$\lambda x{:}\alpha.\, x\, x \;\equiv\; \lambda x^{\alpha}.\, x\, x$$

Утверждение о типизации

Определение

Утверждение (о типизации) в $\lambda_{\to}$ «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 $\lambda_{\to}$ «а ля Чёрч» надо лишь заменить Λ на $\Lambda_{\mathbb{T}}$.

Примеры утверждений о типизации

Система в стиле Карри	Система в стиле Чёрча
$(\lambda x.\, x) : \alpha \to \alpha$	$(\lambda x^{\alpha}.\, x) : \alpha \to \alpha$
$(\lambda x.\, x) : (\alpha \to \beta) \to \alpha \to \beta$	$(\lambda x^{\alpha \to \beta}.\, x) : (\alpha \to \beta) \to \alpha \to \beta$
$(\lambda x\, y.\, x) : \alpha \to \beta \to \alpha$	$(\lambda x^{\alpha}\, y^{\beta}.\, x) : \alpha \to \beta \to \alpha$

Объявления

Определение

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

Примеры объявлений

$$x : \alpha$$
$$y : \beta$$
$$f : \alpha \to \beta$$
$$g : (\alpha \to \beta) \to \gamma$$

Контексты

Определение

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

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

Контекст иногда называют *базисом* или *окружением*.

- Фигурные скобки множества иногда опускают:

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

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

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

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

Правила типизации $\lambda_{\rightarrow}$ «а ля Карри»

Определение

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

$$\Gamma \vdash M{:}\tau$$

если его вывод может быть произведен по правилам:

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

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

Типизация $\lambda_{\to}$ «а ля Карри» в виде правил вывода

(аксиома)	$\Gamma \vdash x : \sigma$, если $(x : \sigma) \in \Gamma$
(удаление $\to$)	$\dfrac{\Gamma \vdash M : \sigma \to \tau \quad \Gamma \vdash N : \sigma}{\Gamma \vdash (M\,N) : \tau}$
(введение $\to$)	$\dfrac{\Gamma, x : \sigma \vdash M : \tau}{\Gamma \vdash (\lambda x.\, M) : \sigma \to \tau}$

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

$$\dfrac{\dfrac{x : \alpha, y : \beta \vdash x : \alpha}{x : \alpha \vdash (\lambda y.\, x) : \beta \to \alpha}}{\vdash (\lambda x\, y.\, x) : \alpha \to \beta \to \alpha}$$

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

Типизация $\lambda_{\to}$ «а ля Чёрч» в виде правил вывода

(аксиома)	$\Gamma \vdash x : \sigma$, если $(x : \sigma) \in \Gamma$
(удаление $\to$)	$\dfrac{\Gamma \vdash M : \sigma \to \tau \quad \Gamma \vdash N : \sigma}{\Gamma \vdash (M\,N) : \tau}$
(введение $\to$)	$\dfrac{\Gamma, x : \sigma \vdash M : \tau}{\Gamma \vdash (\lambda x : \sigma.\, M) : \sigma \to \tau}$

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

$$\dfrac{\dfrac{x : \alpha, y : \beta \vdash x : \alpha}{x : \alpha \vdash (\lambda y : \beta.\, x) : \beta \to \alpha}}{\vdash (\lambda x : \alpha.\, \lambda y : \beta.\, x) : \alpha \to \beta \to \alpha}$$

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

План лекции

Технические леммы

Лемма об инверсии (лемма генерации)

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

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

Лемма сужения

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

- 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 $\tau \to \sigma$.
- По лемме о контекстах $x \in \mathrm{dom}(\Gamma)$ и должен иметь там *единственное* связывание по определению контекста. То есть $\tau = \tau \to \sigma$ — тип является подвыражением себя, чего не может быть, поскольку (*и пока*) типы конечны.

$$x : \tau \nvdash (x\,x) : \sigma, \quad \nvdash \boldsymbol{\omega} : \sigma, \quad \nvdash \boldsymbol{\Omega} : \sigma, \quad \nvdash \mathbf{Y} : \sigma.$$

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

Лемма подстановки типа для $\lambda_{\to}$

Определение

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

Пример

$$[\alpha := \gamma \to \gamma](\alpha \to \beta \to \alpha) \;=\; (\gamma \to \gamma) \to \beta \to \gamma \to \gamma$$

Лемма подстановки типа

$\Gamma \vdash M : \sigma \;\Rightarrow\; [\alpha := \tau]\Gamma \vdash M : [\alpha := \tau]\sigma$. ($\lambda_{\to}$ а ля Карри)
$\Gamma \vdash M : \sigma \;\Rightarrow\; [\alpha := \tau]\Gamma \vdash [\alpha := \tau]M : [\alpha := \tau]\sigma$. ($\lambda_{\to}$ а ля Чёрч)

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

$$x : \alpha \;\vdash\; (\lambda y^{\alpha} z^{\beta} . x) : \alpha \to \beta \to \alpha \quad \Rightarrow$$
$$x : \gamma \to \gamma \;\vdash\; (\lambda y^{\gamma \to \gamma} z^{\beta} . x) : (\gamma \to \gamma) \to \beta \to \gamma \to \gamma$$

Лемма подстановки *терма* для $\lambda_{\to}$

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

$$x\!:\!\gamma \to \gamma \ \vdash \ (\lambda y^{\beta}.\,x)\!:\!\beta \to \gamma \to \gamma$$

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

$$\vdash \ (\lambda y^{\beta}\, p^{\gamma}.\,p)\!:\!\beta \to \gamma \to \gamma$$

Теорема о редукции субъекта

Пусть $M \twoheadrightarrow_{\beta} N$. 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 $\lambda_{\to}$ термов замкнуто относительно редукции.

Единственность типа в $\lambda_{\to}$

Теорема о единственности типа для $\lambda_{\to}$ а ля Чёрч

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

Терм в $\lambda_{\to}$ 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 $\lambda_{\to}$ а ля Чёрч.

Контрпример (для системы а ля Карри)

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

$$\vdash \ (\lambda x\,y.\,x) : \alpha \to (\delta \to \gamma \to \delta) \to \alpha$$
$$\vdash \ (\lambda x\,y.\,x) : (\gamma \to \gamma) \to \beta \to \gamma \to \gamma$$

Для систем в стиле Карри единственности типа нет.

Связь между системами Карри и Чёрча

- Можно задать стирающее отображение $|\cdot| : \Lambda_{\mathbb{T}} \to \Lambda$:

$$\begin{array}{rcl} |x| & \equiv & x \\ |M\,N| & \equiv & |M|\,|N| \\ |\lambda x\!:\!\sigma.\,M| & \equiv & \lambda x.\,|M| \end{array}$$

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

$$M \in \Lambda_{\mathbb{T}} \;\wedge\; \Gamma \vdash_{\text{Ч}} M\!:\!\sigma \;\Rightarrow\; \Gamma \vdash_{\text{К}} |M|\!:\!\sigma$$

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

$$M \in \Lambda \;\wedge\; \Gamma \vdash_{\text{К}} M\!:\!\sigma \;\Rightarrow\; \exists N \in \Lambda_{\mathbb{T}}\,[\Gamma \vdash_{\text{Ч}} N\!:\!\sigma \;\wedge\; |N| \equiv M]$$

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

$$\sigma \text{ обитаем в } \lambda\!\to\text{-Карри} \;\Leftrightarrow\; \sigma \text{ обитаем в } \lambda\!\to\text{-Чёрч}$$

Проблемы разрешимости

- 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 $\lambda_{\to}$ (и в стиле Чёрча, и в стиле Карри) все эти задачи разрешимы.
- ЗПТ выглядит проще ЗСТ, но обычно они эквивалентны: проверка $(M\,N) : \sigma?$ требует синтеза $N : ?$.

Слабая и сильная нормализация

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 $\mathbf{K\,I\,K}$ — сильно нормализуем,
терм $\mathbf{K\,I\,\Omega}$ — слабо нормализуем,
терм $\mathbf{\Omega}$ — не нормализуем.

Теорема о нормализации $\lambda_{\to}$

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 $\lambda_{\to}$

Обе системы $\lambda_{\to}$ (и Карри, и Чёрча) *сильно нормализуемы*.

То есть любой допустимый терм в $\lambda_{\to}$ всегда редуцируется к нормальной форме.