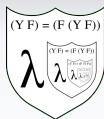


Соответствие Карри–Ховарда для программистов

А. Н. Непейвода
ИПС им. А.К. Айламазяна РАН,
МГТУ им. Н.Э. Баумана

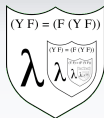
ru-STEP, 28 января 2022



Неформально о λ -исчислении

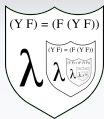
Пусть F , X — термы. $F X$ — операция применения терма F (функции) к терму X (данным).

Пусть M — терм, возможно содержащий x . Тогда абстракция $\lambda x.M$ обозначает анонимную (неименованную) функцию от x : $x \rightarrow M[x]$.



Полуформально о типизации ФВП

- 1 Если $M[x]$ имеет тип σ в контексте $x : \tau$, тогда естественно, что $\lambda x.M$ имеет тип $\tau \rightarrow \sigma$;
- 2 Если $(M N)$ имеет тип σ , а N имеет тип τ , тогда естественно, что M имеет тип $\sigma \rightarrow \tau$.



Полуформально о типизации ФВП

- 1 Если $M[x]$ имеет тип σ в контексте $x : \tau$, тогда естественно, что $\lambda x.M$ имеет тип $\tau \rightarrow \sigma$;
- 2 Если $(M N)$ имеет тип σ , а N имеет тип τ , тогда естественно, что M имеет тип $\sigma \rightarrow \tau$.

Логическая спецификация

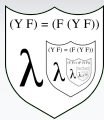
$$\frac{\Gamma, x : \tau \vdash M : \sigma}{\Gamma \vdash \lambda x.M : \tau \rightarrow \sigma}$$

$$\frac{\Gamma \vdash M : \tau \rightarrow \sigma, \Gamma \vdash N : \tau}{\Gamma \vdash (M N) : \sigma}$$



Проблема типизации ФВП

Рассмотрим терм $\lambda x.(x\ x)$. Какой у него тип?



Проблема типизации ФВП

Рассмотрим терм $\lambda x.(x\ x)$. Какой у него тип?

- Пусть тип аргумента применения (то есть x) — это τ , а тип результата применения (то есть $(x\ x)$) — это σ . Тогда $\tau = \tau \rightarrow \sigma$.

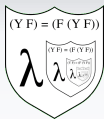


Проблема типизации ФВП

Рассмотрим терм $\lambda x.(x\ x)$. Какой у него тип?

- Пусть тип аргумента применения (то есть x) — это τ ,
а тип результата применения (то есть $(x\ x)$) — это σ .
Тогда $\tau = \tau \rightarrow \sigma$.

Уравнение $\tau = \tau \rightarrow \sigma$ не имеет неподвижной точки,
отличной от $\perp$.



Проблема типизации ФВП

Рассмотрим терм $\lambda x.(x\ x)$. Какой у него тип?

- Пусть тип аргумента применения (то есть x) — это τ ,
а тип результата применения (то есть $(x\ x)$) — это σ .
Тогда $\tau = \tau \rightarrow \sigma$.

Уравнение $\tau = \tau \rightarrow \sigma$ не имеет неподвижной точки,
отличной от $\perp$.

Зацикливается не только вычисление типа: см.
 $(\lambda x.(x\ x)) (\lambda x.(x\ x))$. Однако в некоторых случаях всё
успешно вычисляется: напр. $(\lambda x.(x\ x)) (\lambda x.(\lambda y.(y\ x)))$.



Элементарное соответствие Карри–Ховарда

Вспомним простые правила типизации λ -термов:

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



Элементарное соответствие Карри–Ховарда

Вспомним простые правила типизации λ -термов:

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

...а теперь забудем про термы и посмотрим только на типы.

$$\frac{\Gamma, \tau \vdash \sigma}{\Gamma \vdash \tau \rightarrow \sigma} \quad \text{(правило введения импликации)}$$

$$\frac{\Gamma \vdash \tau \rightarrow \sigma, \Gamma \vdash \tau}{\Gamma \vdash \sigma} \quad \begin{array}{l} \text{(правило удаления импликации} \\ \text{aka } \textit{modus ponens}) \end{array}$$



Древесная форма естественного вывода

Правила вывода для $\Rightarrow$ в минимальной логике

Применению и абстракции соответствуют следующие правила вывода (modus ponens и правило дедукции):

$$\begin{array}{lcl} & & * A \\ & & | B \\ () : & \frac{A \quad A \Rightarrow B}{B} & \lambda. : \frac{}{A \Rightarrow B} \end{array}$$

Тип населён $\Leftrightarrow$ существует λ -терм, имеющий такой тип.
Населённые типы в типизированном λ -исчислении — в точности теоремы минимальной импликативной логики.



Пример разбора задачи на соответствие Карри–Ховарда

Покажем, что тип $((A \Rightarrow ((A \Rightarrow B) \Rightarrow B)) \Rightarrow C) \Rightarrow C$ населён.

Поскольку это — функциональный тип, то внешним конструктором его терма будет абстракция (соответствует правилу дедукции).

$*(A \Rightarrow ((A \Rightarrow B) \Rightarrow B)) \Rightarrow C$ (тип терма x)

(Здесь нужно придумать, как построить терм типа C , имея только x)

C

$((A \Rightarrow ((A \Rightarrow B) \Rightarrow B)) \Rightarrow C) \Rightarrow C$ (тип терма $\lambda x. \dots$)

Поскольку x — это функциональный терм, принимающий аргументом функцию типа $\tau = A \Rightarrow ((A \Rightarrow B) \Rightarrow B)$, нужно попробовать построить терм, имеющий тип τ . Для этого опять понадобится абстракция (даже две).



Пример разбора задачи на соответствие Карри–Ховарда

* $(A \Rightarrow ((A \Rightarrow B) \Rightarrow B)) \Rightarrow C$ (тип терма x)

* A (тип терма y)

Тут опять не хватает шагов: нужно построить терм типа $(A \Rightarrow B) \Rightarrow B$, имея y .

Так как это функция, добавляем ещё абстракцию.

$(A \Rightarrow B) \Rightarrow B$

$A \Rightarrow ((A \Rightarrow B) \Rightarrow B)$ (тип терма $\lambda y. \dots$)

C (тип терма $x (\lambda y. \dots)$)

$((A \Rightarrow ((A \Rightarrow B) \Rightarrow B)) \Rightarrow C) \Rightarrow C$ (тип терма $\lambda x. x (\lambda y. \dots)$)



Пример разбора задачи на соответствие Карри–Ховарда

* $(A \Rightarrow ((A \Rightarrow B) \Rightarrow B)) \Rightarrow C$ (тип терма x)

* A (тип терма y)

* $A \Rightarrow B$ (тип терма z)

Как получить терм типа B из y и z ?

B

$(A \Rightarrow B) \Rightarrow B$ (тип терма $\lambda z. \dots$)

$A \Rightarrow ((A \Rightarrow B) \Rightarrow B)$ (тип терма $\lambda y. (\lambda z. \dots)$)

C (тип терма $x (\lambda y. (\lambda z. \dots))$)

$((A \Rightarrow ((A \Rightarrow B) \Rightarrow B)) \Rightarrow C) \Rightarrow C$ (тип терма $\lambda x. x (\lambda y. \lambda z. \dots)$)



Пример разбора задачи на соответствие Карри–Ховарда

$*(A \Rightarrow ((A \Rightarrow B) \Rightarrow B)) \Rightarrow C$ (тип терма x)

$*A$ (тип терма y)

$*A \Rightarrow B$ (тип терма z)

B (тип терма $z y$)

$(A \Rightarrow B) \Rightarrow B$ (тип терма $\lambda z.(z y)$)

$A \Rightarrow ((A \Rightarrow B) \Rightarrow B)$ (тип терма $\lambda y.(\lambda z.(z y))$)

C (тип терма $x (\lambda y.(\lambda z.(z y)))$)

$((A \Rightarrow ((A \Rightarrow B) \Rightarrow B)) \Rightarrow C) \Rightarrow C$ (тип терма $\lambda x.x (\lambda y.\lambda z.(z y))$)



Определение монады

```
class Applicative M => Monad M where  
  return :: a -> M a  
  (>>=) :: M a -> (a -> M b) -> M b
```

Дополнительные монадические операторы:

```
join :: M M a -> M a  
fmap :: (a -> b) -> M a -> M b
```



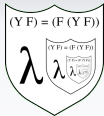
Соответствие Карри–Ховарда

Propositional Lax Logic — монадическая логика с единственным модальным оператором $\bigcirc$.

Правила вывода (+ стандартные правила естественного вывода в интуиционистской логике):

- $\frac{A}{\bigcirc A}$ (Return)
- $\frac{A \Rightarrow \bigcirc B}{\bigcirc A \Rightarrow \bigcirc B}$ (Bind)

Сохраняет возможность извлечь монадический оператор из вывода типов! Обратим внимание на инвертированный порядок аргументов.



Пример разбора задачи на монадическое соответствие

Построим оператор `join` из операторов `>>=` и `return`.

* $\circ \circ A$ (тип терма x)

(Тут нужно придумать, как объединить контейнеры, оборачивающие x)

$\circ A$

$\circ \circ A \Rightarrow \circ A$ (тип терма $\lambda x. \dots$)

Единственный оператор, умеющий заглядывать внутрь контейнера — это `bind`. Он требует два аргумента: контейнерный типа $M\ a$ и монадическую функцию типа $a \rightarrow M\ b$. Поскольку нужно заглянуть внутрь только одной из двух контейнерных оболочек, в роли типа a выступает также монада $M\ a'$. Значит, нам нужна стрелка типа $M\ a' \rightarrow M\ a'$. Это функция `id`, вывод которой мы уже умеем строить.



Пример разбора задачи на монадическое соответствие

* $\circ \circ A$ (тип терма x)

* $\circ A$ (тип терма y)

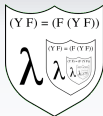
| $\circ A$ (просто возвращаем сам y)

$\circ A \Rightarrow \circ A$ (тип терма $\lambda y.y$)

(Аргументы *bind*-оператора построены,
осталось применить их в правильном порядке)

$\circ A$

$\circ \circ A \Rightarrow \circ A$ (тип терма $\lambda x. \dots$)



Пример разбора задачи на монадическое соответствие

* $\circ \circ A$ (тип терма x)

* $\circ A$ (тип терма y)

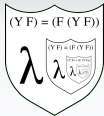
| $\circ A$ (просто возвращаем сам y)

$\circ A \Rightarrow \circ A$ (тип терма $\lambda y.y$)

(Аргументы *bind*-оператора построены,
осталось применить их в правильном порядке)

$\circ A$ (тип терма $x \gg = (\lambda y.y)$)

$\circ \circ A \Rightarrow \circ A$ (тип терма $\lambda x.(x \gg = (\lambda y.y))$)



Пример разбора задачи на монадическое соответствие

* $\circ \circ A$ (тип терма x)

* $\circ A$ (тип терма y)

| $\circ A$ (просто возвращаем сам y)

$\circ A \Rightarrow \circ A$ (тип терма $\lambda y.y$)

(Аргументы *bind*-оператора построены,
осталось применить их в правильном порядке)

$\circ A$ (тип терма $x \gg = (\lambda y.y)$)

$\circ \circ A \Rightarrow \circ A$ (тип терма $\lambda x.(x \gg = (\lambda y.y))$)

С помощью построения вывода в модальной логике мы просто решили уравнение в ФВП, получив ответ:

`join x = x >>= id.`



Выводы

- Двухстороннее применение соответствия Карри–Ховарда облегчает обучение как по логике, так и по теоретическим основам функциональных языков.