

Лекция 4.

Экзистенциальные типы

Типовая система Хиндли-Милнера

## Абстрактные типы данных

Стек  $\alpha$  из значений типа  $v$ : контейнер, соответствующий интерфейсу

метод	тип	комментарий
<code>empty</code>	$\alpha$	(конструктор)
<code>push</code>	$v \rightarrow \alpha \rightarrow \alpha$	
<code>pop</code>	$\alpha \rightarrow \alpha \&v$	

Возможны разные реализации интерфейса.

**Замечание:** Мы понимаем АДТ как набор функций, без собственных данных. Напомним, что `a.method(...)` — другая запись для `method(a, ...)`.

## Пример определения и применения АД

```
abstype stack with
  empty : stack
  push : int * stack -> stack
  pop : stack -> stack * int
is pack Maybe Int,
  empty = None
  push (n,s) = Some n
  pop s = case s with None -> 0 | Some v -> v
in
  stack::pop(stack::push(12,stack::empty))
```

## Экзистенциальные типы

Экзистенциальный тип — тип, соответствующий квантору существования в смысле изоморфизма Карри-Ховарда. Соответствует абстрактному типу данных.

$$\frac{\Gamma \vdash \varphi[\alpha := \theta]}{\Gamma \vdash \exists \alpha. \varphi} \qquad \frac{\Gamma \vdash \exists \alpha. \varphi \quad \Gamma, \varphi \vdash \psi}{\Gamma \vdash \psi}$$

АТД имеет интерфейс  $\varphi$ , тип АТД  $\alpha$  реализуется типом  $\theta$ , а сам интерфейс — термом  $M$ :

$$\frac{\Gamma \vdash M : \varphi[\alpha := \theta]}{\Gamma \vdash (\text{pack } M, \theta \text{ to } \exists \alpha. \varphi) : \exists \alpha. \varphi}$$

... и если вычисление  $N : \psi$  работает при условии наличия какой-то реализации АТД  $x : \varphi$  в контексте, то нам достаточно АТД  $P : \exists \alpha. \varphi$  для получения результата:

$$\frac{\Gamma \vdash P : \exists \alpha. \varphi \quad \Gamma, x : \varphi \vdash N : \psi}{\Gamma \vdash \text{abstype } \alpha \text{ with } x : \varphi \text{ is } P \text{ in } N : \psi} (\alpha \notin FV(\Gamma, \psi))$$

## Стек в $F$

$$\frac{\Gamma \vdash M : \varphi[\alpha := \theta]}{\Gamma \vdash (\text{pack } M, \theta \text{ to } \exists \alpha. \varphi) : \exists \alpha. \varphi} \quad \frac{\Gamma \vdash P : \exists \alpha. \varphi \quad \Gamma, x : \varphi \vdash N : \psi}{\Gamma \vdash \text{abstype } \alpha \text{ with } x : \varphi \text{ is } P \text{ in } N : \psi}$$

Интерфейс стека (возьмём  $v$  как чёрчевский нумерал):

$$\varphi := (\underbrace{\alpha}_{\text{empty}} \& \underbrace{(v \& \alpha \rightarrow \alpha)}_{\text{push}}) \& \underbrace{(\alpha \rightarrow \alpha \& v)}_{\text{pop}}$$

Какое-нибудь вычисление — скажем,  $\text{pop}(\text{push}(12, \text{empty}))$ :

$$x : \varphi \vdash \underbrace{\pi_R((\pi_R x)((\pi_R(\pi_L x))\langle 12, \pi_L(\pi_L x) \rangle))}_{N} : v$$

И простая реализация, для  $\theta := (\gamma \rightarrow \gamma) \vee v$  — это `Maybe Int`:

$$\vdash \langle \langle (\text{In}_L \lambda x. x), \lambda n. \text{In}_R (\pi_L n) \rangle, \lambda n. \text{case } (\lambda x. 0) (\lambda x. x) n \rangle : \varphi[\alpha := \theta]$$

## Раскрываем $\exists$ через $\forall$

Напомним, что  $\exists\alpha.\varphi := \forall\beta.(\forall\alpha.\varphi \rightarrow \beta) \rightarrow \beta$ .

$$\frac{\Gamma \vdash M : \varphi[\alpha := \theta]}{\Gamma \vdash (\text{pack } M, \theta \text{ to } \exists\alpha.\varphi) : \exists\alpha.\varphi}$$

Перепишем это правило только через базовые конструкции системы  $F$ :

$$\frac{\Gamma \vdash M : \varphi[\alpha := \theta]}{\Gamma \vdash \Lambda\beta.\lambda e^{\forall\alpha.\varphi \rightarrow \beta}.(e \ \theta) \ M : \forall\beta.(\forall\alpha.\varphi \rightarrow \beta) \rightarrow \beta}$$

«Пусть есть вычисление  $e$ , использующее АТД  $\alpha$  с интерфейсом  $\varphi$ , возвращающее  $\beta$ . Тогда, имея конкретный тип реализации АТД  $\theta$  и саму реализацию АТД  $M : \varphi[\alpha := \theta]$ , то с помощью вычисления  $e$  возможно вычислить результат и вернуть значение типа  $\beta$ ».

Сравните с case для алгебраического типа и вспомните действия редактора связей (линкера).

## Раскроем abstype

$$\frac{\Gamma \vdash P : \exists \alpha. \varphi \quad \Gamma, x : \varphi \vdash N : \psi}{\Gamma \vdash \text{abstype } \alpha \text{ with } x : \varphi \text{ is } P \text{ in } N : \psi}$$

Перепишем это правило через базовые конструкции системы  $F$ :

$$\frac{\Gamma \vdash P : \forall \beta. (\forall \alpha. \varphi \rightarrow \beta) \rightarrow \beta \quad \Gamma, x : \varphi \vdash N : \psi}{\Gamma \vdash (P \ \psi) \ (\Lambda \alpha. \lambda x^\varphi. N) : \psi}$$

Вспомним  $\text{rask}$ :

$$\frac{\Gamma \vdash M : \varphi[\alpha := \theta]}{\Gamma \vdash \Lambda \beta. \lambda e^{\forall \alpha. \varphi \rightarrow \beta}. (e \ \theta) \ M : \forall \beta. (\forall \alpha. \varphi \rightarrow \beta) \rightarrow \beta}$$

Результат:

$$\begin{aligned} & ((\Lambda \beta. \lambda e^{\forall \alpha. \varphi \rightarrow \beta}. (e \ \theta) \ M) \ \psi) \ (\Lambda \alpha. \lambda x^\varphi. N) \rightarrow_\beta \\ & (\lambda e^{\forall \alpha. \varphi \rightarrow \psi}. (e \ \theta) \ M) \ (\Lambda \alpha. \lambda x^\varphi. N) \rightarrow_\beta \\ & (\Lambda \alpha. \lambda x^\varphi. N) \ \theta \ M \rightarrow_\beta \\ & (\lambda x^{\varphi[\alpha := \theta]}. N[\alpha := \theta]) \ M \rightarrow_\beta N[\alpha := \theta][x := M] \end{aligned}$$

## Пример реализации на Хаскеле

```
{-# LANGUAGE RankNTypes #-}
data AbstractStack = AS (forall b . (forall a .
    ( a, Integer -> a -> a, a -> (a, Integer) )
    -> b) -> b)

abstype :: AbstractStack -> Integer
abstype stack =
  case stack of
    AS r -> r x where
      x (empty, push, pop) =
        let (stk, v) = pop (push 12 $ push 5 empty) in
        let (stk2, v2) = pop stk in
        v + v2

packedStack :: AbstractStack
packedStack = AS (\t -> t ( [], \i -> \l -> i:l, \ (i:l) -> (l,i) ) )

main = do print (abstype packedStack)
```



## Общие свойства системы $F$

В системе  $F$  (в варианте по Чёрчу, так и в варианте по Карри) имеют место теорема Чёрча-Россера и сильная нормализация.

Разрешимость задач типизации системы  $F$ :

	По Чёрчу	По Карри
$\Gamma \vdash M : \sigma$	да	нет
$\Gamma \vdash M : ?$	да	нет
$\Gamma \vdash ? : \sigma$	нет	нет
$? \vdash M : \sigma$	нет	нет
$? \vdash M : ?$	нет	нет

## Ранг типа

Напомним, что  $\exists\alpha.\varphi := \forall\beta.(\forall\alpha.\varphi \rightarrow \beta) \rightarrow \beta$ .

### Определение

Функция «ранг типа»  $rk \subseteq T \times \mathbb{N}_0$ .  $rk(\sigma) = [mrk(\sigma), +\infty) \cap \mathbb{N}_0$ , где  $mrk$ :

$$mrk(\tau) = \begin{cases} 0, & \tau \text{ без кванторов} \\ \max(mrk(\sigma), 1), & \tau = \forall x.\sigma \\ \max(mrk(\sigma_1) + 1, mrk(\sigma_2)), & \tau = \sigma_1 \rightarrow \sigma_2, \tau \text{ имеет кванторы} \end{cases}$$

### Лемма

Если  $rk(\sigma, 1)$ , то для формулы  $\sigma$  найдётся эквивалентная формула с поверхностными кванторами.

### Пример

$0 \notin rk(\forall\alpha.\gamma \rightarrow \beta)$ ;  $1 \notin rk((\forall\alpha.\gamma \rightarrow \beta) \rightarrow f) = \{2, 3, \dots\}$

$1 \notin rk(\exists\alpha.\gamma) = rk(\forall\beta.(\forall\alpha.\gamma \rightarrow \beta) \rightarrow \beta) = \{2, 3, \dots\}$

$1 \in rk(\forall\alpha.\delta \rightarrow \forall\beta.\delta \rightarrow \forall\gamma.\delta)$

# Типовая система Хиндли-Милнера: язык

## Определение

Тип ( $\tau$ ) и типовая схема:

$$\tau ::= \alpha \mid (\tau \rightarrow \tau) \quad \sigma ::= \forall x. \sigma \mid \tau$$

Пред-лямбда-терм (типизация по Карри)

$$H ::= x \mid (H \ H) \mid (\lambda x. H) \mid (\text{let } x = H \text{ in } H)$$

Редукция для let:

$$\text{let } x = E_1 \text{ in } E_2 \rightarrow_{\beta} E_2[x := E_1]$$

## Пример

$$\text{let } \text{Inc} = \lambda n. \lambda f. \lambda x. n \ f \ (f \ x) \text{ in } \text{Inc}(\text{Inc } \bar{0}) \rightarrow_{\beta} \bar{2}$$

# Типовая система Хиндли-Милнера: специализация

## Определение

Пусть  $\sigma_1 = \forall \alpha_1. \forall \alpha_2. \dots \forall \alpha_n. \tau_1$ . Тогда  $\sigma_2$  — частный случай или специализация  $\sigma_1$  (обозначается как  $\sigma_1 \sqsubseteq \sigma_2$ ), если

$$\sigma_2 = \forall \beta_1. \forall \beta_2. \dots \forall \beta_m. \tau_1[\alpha_1 := S(\alpha_1), \dots, \alpha_n := S(\alpha_n)] \text{ и } \beta_i \notin FV(\forall \alpha_1. \forall \alpha_2. \dots \forall \alpha_n. \tau_1)$$

## Пример

$$\forall \alpha. \alpha \rightarrow \alpha \sqsubseteq \forall \beta_1. \forall \beta_2. (\beta_1 \rightarrow \beta_2) \rightarrow (\beta_1 \rightarrow \beta_2)$$

## Типовая система Хиндли-Милнера: правила вывода

$$\begin{array}{c}
 \overline{\Gamma, x : \sigma \vdash x : \sigma} \quad x \notin FV(\Gamma) \qquad \frac{\Gamma \vdash E_0 : \tau \rightarrow \tau' \quad \Gamma \vdash E_1 : \tau}{\Gamma \vdash E_0 E_1 : \tau'} \qquad \frac{\Gamma, x : \tau \vdash E : \tau'}{\Gamma \vdash \lambda x. E : \tau \rightarrow \tau'} \\
 \\
 \frac{\Gamma \vdash E_0 : \sigma \quad \Gamma, x : \sigma \vdash E_1 : \tau}{\Gamma \vdash \text{let } x = E_0 \text{ in } E_1 : \tau} \qquad \frac{\Gamma \vdash E : \sigma' \quad \sigma' \sqsubseteq \sigma}{\Gamma \vdash E : \sigma} \qquad \frac{\Gamma \vdash E : \sigma}{\Gamma \vdash E : \forall \alpha. \sigma} \quad \alpha \notin FV(\Gamma)
 \end{array}$$

### Пример

$$\begin{array}{c}
 \overline{x : \alpha \vdash x : \alpha} \\
 \hline
 \vdash \lambda x. x : \alpha \rightarrow \alpha \\
 \hline
 \vdash \lambda x. x : \forall \alpha. \alpha \rightarrow \alpha
 \end{array}$$
  

$$\frac{\frac{\overline{\text{id} : \forall \alpha. \alpha \rightarrow \alpha \vdash \text{id} : \forall \alpha. \alpha \rightarrow \alpha}}{\text{id} : \forall \alpha. \alpha \rightarrow \alpha \vdash \text{id} : \text{int} \rightarrow \text{int}}} S(\alpha) = \text{int} \quad \text{id} : \forall \alpha. \alpha \rightarrow \alpha \vdash 0 : \text{int}}{\text{id} : \forall \alpha. \alpha \rightarrow \alpha \vdash \text{id } 0 : \text{int}}$$

Отсюда: `let id = λx.x in ⟨id 0, id «a»⟩ : int&string`

## Алгоритм реконструкции типа $W$

На вход подаются  $\Gamma$ ,  $M$ , на выходе наиболее общая пара:  $\langle S, \tau \rangle = W(\Gamma, M)$

1.  $M = x$ ,  $x : \tau \in \Gamma$  (иначе ошибка)

- ▶  $\tau' \dashv \tau$  без кванторов, все свободные переменные переименованы в свежие.

возвращаем  $\langle \emptyset, \tau' \rangle$ ; например,  $W(\{x : \forall \alpha. \varphi, y : \beta\}, x) = \langle \emptyset, \varphi[\alpha := \gamma] \rangle$

2.  $M = \lambda n. E$

- ▶  $\Gamma' = \{x : \sigma \mid x : \sigma \in \Gamma, x \neq n\} \cup \{n : \alpha\}$ ,  $\alpha$  — свежая типовая переменная
- ▶  $\langle S, \tau \rangle = W(\Gamma', E)$

возвращаем  $\langle S, S(\alpha) \rightarrow \tau \rangle$

3.  $M = P \ Q$

- ▶  $\langle S_1, \tau_1 \rangle = W(\Gamma, P)$ ;  $\langle S_2, \tau_2 \rangle = W(S_1(\Gamma), Q)$
- ▶  $S_3 = \mathcal{U}[S_2(\tau_1), \tau_2 \rightarrow \alpha]$ ,  $\alpha$  — свежая

возвращаем  $\langle S_3 \circ S_2 \circ S_1, S_3(\alpha) \rangle$

4.  $M = (\text{let } n = P \text{ in } Q)$

- ▶  $\langle S_1, \tau_1 \rangle = W(\Gamma, P)$
- ▶  $\Gamma' = \{x : \sigma \mid x : \sigma \in \Gamma, x \neq n\} \cup \{n : \forall \alpha_1 \dots \alpha_k. \tau_1\}$ , где  $\alpha_1 \dots \alpha_k$  — все свободные переменные  $\tau_1$
- ▶  $\langle S_2, \tau_2 \rangle = W(S_1(\Gamma'), Q)$

возвращаем  $\langle S_2 \circ S_1, \tau_2 \rangle$

## Рекурсия в НМ: делаем НМ тьюринг-полной

1. Рекурсия для термов.  $Y$ -комбинатор. Добавим специальное правило вывода:

$$\overline{Y : \forall \alpha. (\alpha \rightarrow \alpha) \rightarrow \alpha}$$

2. Рекурсия для типов. Рассмотрим список

$$\text{Nil} = \text{In}_L 0 \quad \text{Cons } e \, l = \text{In}_R \langle e, l \rangle \quad \text{List} : ?$$

Заметим, что при попытке выписать уравнение для типа мы получим рекурсию:

$$\tau = \text{Int} \vee \langle \text{Int}, \tau \rangle$$

Рекурсивный тип надо добавить явно:

$$\tau = \mu \alpha. \text{Int} \vee \langle \text{Int}, \alpha \rangle$$

Мю-оператор — это  $Y$ -комбинатор для типов. Как его добавить в типовую систему?

## Эквирекursивные и изорекursивные типы: $\mu\alpha.\sigma(\alpha)$

- ▶ Эквирекursивные типы. Считаем, что  $\alpha = \sigma(\alpha)$ . Например, в Java:

```
public abstract class Enum<E extends Enum<E>>
    implements Constable, Comparable<E>, Serializable
{ ... }
```

Уравнение (частный случай):  $E = Enum(E)$ , или  $E = \mu\varepsilon.Enum(\varepsilon)$ .

- ▶ Изорекursивные типы.  $\alpha \neq \sigma(\alpha)$ , но есть изоморфизм:

$$\text{roll} : \sigma(\alpha) \rightarrow \alpha \quad \text{unroll} : \alpha \rightarrow \sigma(\alpha)$$

Например, для struct List { List\* next; int value; }:

Комп.	В C++	Пример
roll	взятие ссылки	List a; a.next = NULL; return len(&a)
unroll	разыменование	len (List* a) { return (*a).next ? ... : 0 }



## Разрешимость задачи реконструкции типа в разных вариантах $F$

Ранг типов	Собственное название	Разрешимость
0	$\lambda \rightarrow$	разрешимо (лекция 2)
1	$HM$	разрешимо (алгоритм $W$ )
2		разрешимо
$\geq 3$		неразрешимо