1 Лекция 9 (30.10.2018)

1.1

Def. Ранг типа

$R(x)$ — все типа ранга x.

- $R(0)$ — все типы без кванторов
- $R(x+1) = R(x) \mid R(x) \to R(x+1) \mid \forall \alpha . R(x+1)$

Enddef.

Например:

- $\alpha \in R(0)$
- $\forall \alpha . \alpha \in R(1)$
- $(\forall \alpha . \alpha) \to (\forall b . b) \in R(2)$
- $((\forall \alpha . \alpha) \to (\forall b . b)) \to b \in R(3)$

Тут видно, если если выражение слева от знака имликации имеет ранг n, то все выражение будет иметь ранг $\geq (n+1)$.

Утверждение: Пусть x — выражение только с поверхностными кванторами, тогда $x \in R(1)$.

Def. Типовая система

$\sigma ::= \forall \alpha_1 . \forall \alpha_2 . \dots . \forall \alpha_n . \tau$, где $\tau \in R(0)$ и, следовательно, $\sigma \in R(1)$.

Enddef.

Def. Частный случай (спциализация) типовой схемы

$\sigma_1 \sqsubseteq \sigma_2$ — типовая схема

σ_2 — частный случай (специализация) σ_1, если

1. $\sigma_1 = \forall \alpha_1 . \forall \alpha_2 . \dots . \forall \alpha_n . \tau_1$
2. $\sigma_2 = \forall \beta_1 . \forall \beta_2 . \dots . \forall \beta_n . \tau_1 [\alpha_i := S(\alpha_i)]$
3. $\forall i . \beta_i \in FV(\tau_1)$

Enddef.

$M_1 : \forall \alpha . \alpha \to \alpha$

$M : \forall \beta_1 . \forall \beta_2 : (\beta_1 \to \beta_2) \to (\beta_1 \to \beta_2)$

Вполне возможно, что в ходе замены, все типы будут уточнены (α уточниться как $\beta_1 \to \beta_2$.

1.2 Хиндли-Милнер

1. Все типы только с поверхностными кванторами ($R(1)$)
2. $\overline{HM} ::= p \mid \overline{HM}\ \overline{HM} \mid \lambda p . \overline{HM} \mid let = \overline{HM}\ in\ \overline{HM}$

Докажем: $\dfrac{\Gamma \vdash \phi[p := \Theta]}{\Gamma \vdash \exists p . \phi}$

- $\exists p.\phi = \forall b.(\forall p.(\phi \to b)) \to b$
- $\phi \to \perp \equiv \forall b.(\phi \to b)$
- $$\cfrac{\cfrac{\cfrac{\cfrac{\Gamma, \forall p.(\phi \to b) \vdash \forall p.(\phi \to b)}{\Gamma, \forall p.(\phi \to b) \vdash \phi[p := \Theta] \to b}}{\Gamma, \forall p.(\phi \to b) \vdash b}}{\Gamma \vdash (\forall p.(\phi \to b)) \to b}}{\Gamma \vdash \forall b.(\forall p.(\phi \to b)) \to b}$$

Соглашение:

- σ — типовая схема
- τ — простой тип

1. $\dfrac{}{\Gamma, x : \sigma \vdash x : \sigma}$
2. $\dfrac{\Gamma \vdash e_1 : \tau \to \tau' \qquad \Gamma \vdash e_1 : \tau}{\Gamma \vdash e_0\ e_1 : \tau'}$
3. $\dfrac{\Gamma, x : \tau \vdash e : \tau'}{\Gamma \vdash \lambda x.e : \tau \to \tau'}$
4. $\dfrac{\Gamma \vdash e_0 : \sigma \qquad \Gamma, x : \sigma \vdash e_1 : \tau}{\Gamma \vdash let\ x = e_0\ in\ e_1 : \tau}$, $let\ x = a\ in\ b \equiv (\lambda x.b)\ a$
5. $\dfrac{\Gamma \vdash e : \sigma' \qquad \sigma' \sqsubseteq \sigma}{\Gamma \vdash e : \sigma}$
6. $\dfrac{\Gamma \vdash e : \sigma}{\Gamma \vdash e : \forall \alpha.\sigma}\ \alpha \notin FV(\Gamma)$

1.3 Алгоритм вывода типов в системе Хиндли-Милнера W

На вход подаются $\Gamma,\ M$, на выходе наиболее общая пара (S, τ)

1. $M = x : \tau,\ x \in \Gamma$ (иначе ошибка)
   - Выбросить все кванторы из τ
   - Переименовать все свободные переменные в свежие
     Например: $\forall \alpha_1.\phi \Rightarrow \phi[\alpha_1 := \beta_1]$, где β_1 — свежая переменная

   $(\emptyset, \Gamma(x))$
2. $M = \lambda n.e$
   - τ — новая типовая переменная
   - $\Gamma' = \Gamma \setminus \{n :_\}$ (т.е. Γ без переменной n)
   - $\Gamma'' = \Gamma' \cup n : \tau$
3. $M = P\ Q$

- τ — новая типовая переменная
- $(S_1, \tau_1) = W(\Gamma, P)$
- $(S_2, \tau_2) = W(S_1(\Gamma), Q)$
- S_3 — Унификация $(S_2(\tau_1), \tau_2 \to \tau)$ $(S_3 \circ S_2 \circ S_3, S_3(\tau))$

4. *let* $x = P$ *in* Q

- $(S, \tau) = W(\Gamma, P)$
- $\Gamma' = \Gamma$ без x
- $\Gamma'' = \Gamma' \cup \{x : \forall \alpha_1 \ldots \alpha_k . \tau_1\}$, где $\alpha_1 \ldots \alpha_k$ свободные переменные в τ_1
- $(S_2, \tau_2) = W(S_1(\Gamma''), Q)$
- $(S_1 \circ S_2), \tau_2)$

Надеемся, что логика второго порядка противоречива.
Введем явный Y-комбинатор
$Y f =_\beta f(Y\ f)$
$Y : \forall \alpha . (\alpha \to \alpha) \to alpha$ — аксиома

```
type intList = Nil | Cons of int * intList;;

let rec length l = match l with
  | Nil -> 0
  | Cons (x, s) -> 1 + length s;;

let my_list = Cons(1, Cons (2, Cons (3, Nil)));;

print_int (length my_list);;  (* output: 3 *)
```

$Nil = inLeft O = \lambda a . \lambda b . a\ O$
$Cons = inRight p = \lambda a . \lambda b . b\ p$
$\lambda a . \lambda b . a\ O : \forall \gamma . (\alpha \to \gamma) \to (\beta \to \gamma) \to \gamma$
$\lambda a . \lambda b . b\ O : \forall \gamma . (\alpha \to \gamma) \to (\beta \to \gamma) \to \gamma$
$\delta = \forall \gamma . (\alpha \to \gamma) \to (\beta \to \gamma) \to \gamma$
$\lambda a . \lambda b . b\ (\lambda a . \lambda b . a\ O) : \forall \alpha . (\alpha \to \gamma) \to (\delta \to \gamma) \to \gamma$
Научимся задавать рекурсивные типы.

1. Эквирекурсивный

   ```
   list = Nil | Cons a * list
   ```

   $\alpha = f(\alpha)$ — уравнение с неподвижной точкой. Пусть $\nu \alpha . f(\alpha) = f(\nu \alpha . f(\alpha))$. Пояснение: Y — для выражений, а для типов — ν.

   ```
   class Enum <extends Enum<E>>
   ```

2. Изорекурсивный

```
struct list {
    list x;
}
x.x.x.x
```

$* : list* \to list$ — разыменовывание

$Roll : Nil|Cons(a * list) \to list$

$Unroll : list \to Nil|Cons(a * list)$

Общий тип (введение в типовую систему):

- $roll : f(\alpha) \to \alpha$
- $unroll : \alpha \to f(\alpha)$

Пример из Си:

- $* : T* \to T$
- $\& : T \to T*$
- $T = \alpha$
- $T* = f(\alpha)$

Зависимые типы и логика 1-ого порядка

$sprintf : string \to smth \to string$

$sprintf"\%d" : int \to string$

$sprintf"\%f" : float \to string$

тип sprintf определяется первым аргументом.

2 Лекция 10 (06.11.2019)

2.1 Обобщенные типовые системы

- Сорта: {*, □}
  - Выражение "$A : *$"означает, что A — тип. И тогда, если на метаязыке мы хотим сказать "Если A тип, то и $A \to A$ тоже тип то формально это выглядит как $A : * \vdash (A \to A) : *$
  - □ - это абстракция над сортом для типов.
- $T ::= x \mid c \mid T\ T \mid \lambda x : T.\ T\ |\Pi x : T.\ T$
- Аксиома:
  - $\dfrac{}{\vdash *.\square}$
- Правила вывода:
  1. $\dfrac{\Gamma \vdash A : S}{\Gamma, x : A \vdash x : A}\ x \notin \Gamma$

2. $\dfrac{\Gamma \vdash A : B \qquad \Gamma C : S}{\Gamma, x : C \vdash A : B}$ — правило ослабление (примерно как $\alpha \to \beta \to \alpha$ в И.В.)

3. $\dfrac{\Gamma \vdash A : B \qquad \Gamma \vdash B' : S \qquad B =_\beta B'}{\Gamma \vdash A : B'}$ — правило конверсии

4. $\dfrac{\Gamma \vdash F : (\Pi x : A.B) \qquad \Gamma \vdash a : A}{\Gamma \vdash (F\ a) : B[x := a]}$ — правило применения

- Семейства правила (generic-правила)

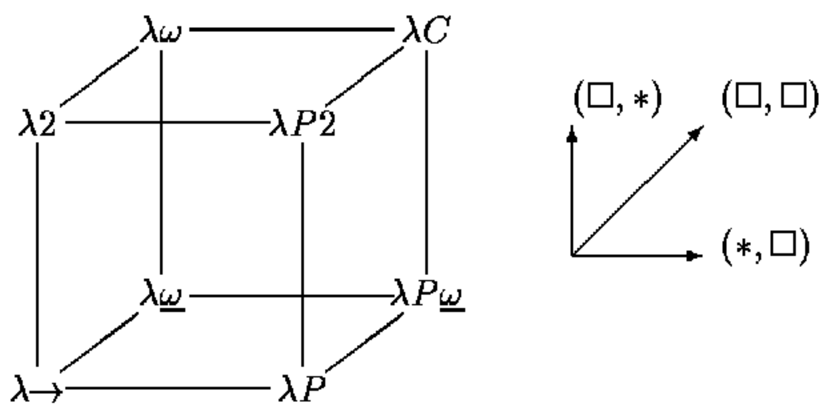
  Пусть $(s_1, s_2) \in S \subseteq \{*, \Box\}^2$.

  1. П-правило: $\dfrac{\Gamma \vdash A : s_1 \qquad \Gamma, x : A \vdash B : s_2}{\Gamma \vdash (\Pi x : A.B) : s_2}$
  2. λ-правило: $\dfrac{\Gamma \vdash A : s_1 \qquad \Gamma, x : A \vdash b : B \qquad \Gamma, x : A \vdash B : s_2}{\Gamma \vdash (\lambda x : A.b) : (\Pi x : A.B)}$

Например:

- $5 : int : * : \Box$
- $[] : * \to * : \Box$
- $\Lambda M.List < M >: * \to * \Box$

2.2 λ-куб



Th
Обобщенная типовая система сильно нормализуема

Примеры:

- $\lambda\omega$:

$$\vdash (\lambda\alpha : *.\alpha \to \alpha)(* \to *) : \Box$$

1. $\dfrac{\vdash * : \Box \qquad \dfrac{\vdash *.\Box}{a : * \vdash *.\Box}}{\vdash (* \to *) : \Box}$

2. $$\vdash * : \Box \quad \frac{\dfrac{\alpha : * \vdash \alpha : * \qquad \alpha : *, x : * \vdash \alpha : *}{\alpha : * \vdash \alpha \to \alpha : x} \qquad \dfrac{\vdash * : \Box \qquad \dfrac{\vdash *.\Box}{a : * \vdash *.\Box}}{\vdash (* \to *) : \Box}}{\vdash (\lambda\alpha : *.\alpha \to \alpha) : * \to *}$$

Notes:

- $(\lambda x.x) : (A \to A)$ - implicit typing (Curry style)
- $I_A = \lambda x : A.x$ - explicit typing (Church style)