Лекция 6
Реконструкция типов в просто типизированном лямбда-исчислении, комбинаторы

1 Лекция 6 Реконструкция типов в просто типизированном лямбда-исчислении, комбинаторы

1.1 Алгоритм вывода типов

Пусть есть: $? \vdash A\ :?$, хотим найти пару $\langle \text{контекст}, \text{тип} \rangle$

Алгоритм:

1. Рекурсия по структуре формулы

   Построить по формуле A пару $\langle E, \tau \rangle$, где

   $E-$набор уравнений, $\tau-$тип A

2. Решение уравнения, получения подстановки S и из решения E и $S\ (\tau)$ получение ответа

Т.е. необохимо свести вывод типа к алгоритму унификации.

Пункт 1.1. Рассмотрим 3 случая

Обозначение $\rightarrow$ – алгебраический тип

1. $A \equiv x \implies \langle \{\}, \alpha_A \rangle$, где $\{\}-$пустой конекст, α_A-новая переменная нигде не встречавшаяся до этого в формуле

2. $A \equiv P\,Q \implies \langle E_P \cup E_Q \cup \{\tau_P = \rightarrow\ (\tau_Q\ \alpha_A)\}, \alpha_A \rangle$, где α_A-новая переменная

3. $A \equiv \lambda x.P \implies \langle E_P, \alpha_x\ \rightarrow\ \tau_P \rangle$

Пункт 1.2. Алгоритм унификации

Рассмотрим $E-$набор уравнений, запишем все уравнения в алгебраическом виде т.е. $\alpha\ \rightarrow\ \beta \Leftrightarrow \rightarrow\ \alpha\,\beta$, затем применяем алгоритм унификации.

Лемма 1.1. Рассмотрим терм M и пару $\langle E_M, \tau_M \rangle$, Если $\Gamma \vdash M : \rho$, то существует:

1. $S-$решение E_M тогда $\Gamma = \{x : S(\alpha_x) \mid x \in FV(M)\}$, $FV-$множество свободных переменных в терме M, α_x- переменная полученная при разборе терма M

   $\rho\ =\ S(\tau_M)$

2. Если $S-$ решение E_M, то $\Gamma \vdash M : \rho$,

   Доказательство. индукция по структуре терма M

   (a) Если $M \equiv x$, то так как решение существует, то существует и $S(\alpha_x)$, что: $\Gamma,\ x : S(\alpha_x)\ \vdash\ x : S(\alpha_x)$

   (b) Если $M \equiv \lambda x.\, P$, то по индукции уже известен тип P, контекст Γ и тип x, тогда:

   $$\frac{\Gamma,\ x:\ S(\alpha_x)\ \vdash\ P:\ S(\alpha_P)}{\Gamma\ \vdash\ \lambda x.\, P : S(\alpha_x)\ \to\ S(\alpha_P)}$$

   (c) Если $M \equiv P\, Q$, то по индукции:

   $$\frac{\Gamma\ \vdash\ P:\ S(\alpha_P) \equiv \tau_1\ \to\ \tau_2 \qquad\qquad \Gamma\ \vdash\ Q : S(\alpha_Q) \equiv \tau_1}{\Gamma\ \vdash\ P\, Q : \tau_2}$$

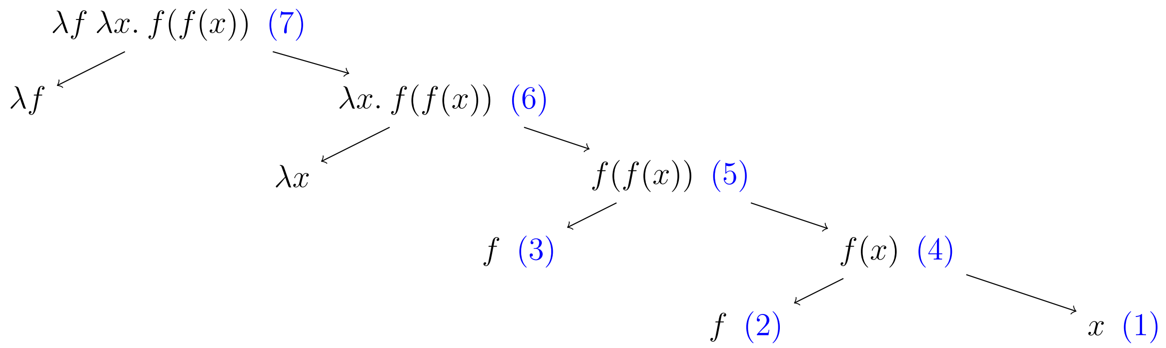
$\square$

$\langle \Gamma, \rho \rangle$ – основная пара для терма M, если

1. $\Gamma \vdash M : \tau$
2. Если $\Gamma' \vdash M : \tau'$, то сущесвтует $S:\ S(\Gamma)\ \subset\ \Gamma'$

Пример.

Рассмотрим терм: $\lambda f\ \lambda x.\ f(f(x))$, построим и пронумеруем его дерево разбора:



1. $E_1 = \langle \{\},\ \alpha_x \rangle$
2. $E_2 = \langle \{\},\ \alpha_f \rangle$
3. $E_3 = \langle \{\},\ \alpha_f \rangle$
4. $E_4 = \langle \{\alpha_f = \to (\alpha_x\ \alpha_1)\}, \alpha_1 \rangle$
5. $E_5 = \left\langle \begin{Bmatrix} \alpha_f = \to (\alpha_x\ \alpha_1) \\ \alpha_f = \to (\alpha_1\ \alpha_2) \end{Bmatrix},\ \alpha_2 \right\rangle$
6. $E_6 = \left\langle \begin{Bmatrix} \alpha_f = \to (\alpha_x\ \alpha_1) \\ \alpha_f = \to (\alpha_1\ \alpha_2) \end{Bmatrix},\ \alpha_x \to \alpha_2 \right\rangle$
7. $E_7 = \left\langle \begin{Bmatrix} \alpha_f = \to (\alpha_x \alpha_1) \\ \alpha_f = \to (\alpha_1 \alpha_2) \end{Bmatrix},\ \alpha_f \to (\alpha_x \to \alpha_2) \right\rangle$

$E = \begin{Bmatrix} \alpha_f = \to\ (\alpha_x\, \alpha_1) \\ \alpha_f = \to\ (\alpha_1\, \alpha_2) \end{Bmatrix}$, решим полученную систему:

1. Решим сисетму:

   (a)
   $$\begin{cases} \alpha_f = \to (\alpha_x\, \alpha_1) \\ \alpha_f = \to (\alpha_1\, \alpha_2) \end{cases}$$

   (b)
   $$\{ \to (\alpha_1\, \alpha_2) = \to (\alpha_x\, \alpha_1)$$

   (c)
   $$\begin{cases} \alpha_1 = \alpha_x \\ \alpha_2 = \alpha_1 \end{cases}$$

   (d)
   $$\begin{cases} \alpha_1 = \alpha_x \\ \alpha_2 = \alpha_x \end{cases}$$

2. Получим
   $$S = \begin{cases} \alpha_f = \to (\alpha_x\, \alpha_1) \\ \alpha_1 = \alpha_x \\ \alpha_2 = \alpha_x \end{cases}$$

3. $\Gamma = \{\}$, так как в заданной формуле нет свободных переменных

4. тип терма $\lambda\, f \lambda\, x.f(f(x))$ является результат подстановки
   $S(\to\ \alpha_f\ (\alpha_x \to \alpha_2))$, получаем $\tau = (\alpha_x \to \alpha_x) \to (\alpha_x \to \alpha_x)$

1.2 Сильная и слабая нормализации

Определение 1.1. Если существует последовательность редукций, приводящая терм M в нормальную форму, то $M-$слабо нормализуем. (Т.е. при редуцировании терма M мы можем не прийти в н.ф.)

Определение 1.2. Если не существует бесконечной последовательности редукций терма M, то терм $M-$ сильно нормализуем.

Утверждение 1.1.

1. $KI\Omega-$ слабо нормализуема

   Пример.

   Перепишем $KI\Omega$ как $((\lambda x\, \lambda y.\, x)(\lambda x.\, x))(((\lambda x.\, x\, x)(\lambda x.\, x\, x)))$, очевидно, что этот терм можно средуцировать двумя разными способами:

   (a) Сначала редуцируем красную скобку
      i. $((\lambda x\, \lambda y.\, x)(\lambda x.\, x))(((\lambda x.\, x\, x)(\lambda x.\, x\, x)))$
      ii. $((\lambda y.\, (\lambda x.\, x)))(((\lambda x.\, x\, x)(\lambda x.\, x\, x)))$

iii. $(\lambda x.\, x)$

Видно, что в этом случае количество шагов конечно.

(b) Редуцируем синюю скобку. Очевидно, что комбинатор Ω не имеет нормальной формы, тогда понятно, что в этом случае терм $KI\Omega$ никогда не средуцируется в нормальную форму.

2. $\Omega-$ не нормализуема

3. $II-$ сильно нормализуема

Лемма 1.2. Сильная нормализация влечет слабую.

1.3 Выразимость комбинаторов

Утверждение 1.2. Любое λ выражение можно записать с помощью комбинаторов S и K, где

$$S = \lambda x\, \lambda y\, \lambda z.\, (x\, z)\, (y\, z)\ :\ (a \to b \to c) \to (a \to b) \to a \to c$$
$$K = \lambda x\, \lambda y.\, x\ : a \to b \to a$$

Утверждение 1.3. Комбинаторы S и K являются аксиомами в ИИВ

Утверждение 1.4. Соотношение комбинаторов с λ исчислением:

1. $T(x) = x$
2. $T(P\, Q) = T(P)\, T(Q)$
3. $T(\lambda\, x.P) = K(T(P)),\ \ x \notin FV(P)$
4. $T(\lambda\, x.x) = I$
5. $T(\lambda\, x\lambda\, y.P) = T(\lambda\, x.\, T(\lambda\, y.P))$
6. $T(\lambda\, x.P\, Q) = S\ \ T(\lambda\, x.P)\, T(\lambda\, x.Q)$

Утверждение 1.5. Связь комбинаторов с ИИВ

Утверждение 1.6. Альтернативный базис:

1. $B = \lambda x\, \lambda y\, \lambda z.\, x\, (y\, z)\ :\ (a \to b) \to (c \to a) \to c \to b$
2. $C = \lambda x\, \lambda y\, \lambda z.\, ((x\, z)\, y)\ :\ (a \to b \to c) \to b \to a \to c$
3. $W = \lambda x\, \lambda y.\, ((x\, y)\, y)\ :\ (a \to a \to b) \to a \to b$