Типовая система Хиндли-Милнера. Экзистенгциальные типы

25 декабря 2018 г.

1 Ранг типа

Введем определение. Под рангом типа мы будем понимать число, получаемое по следующим правилам:

$$Rn(\tau) = \begin{cases} 0, \qquad \tau \equiv \alpha \\ Rn(\phi), \qquad \tau \equiv \sigma \to \phi \text{если} \sigma \text{не содержит } \forall \\ max(Rn(\sigma)+1, Rn(\phi)), \qquad \tau \equiv \sigma \to \phi \text{если} \sigma \text{содержит } \forall \\ max(Rn(\phi), 1),, \qquad \tau \equiv \forall \alpha.\phi \end{cases}$$

Примеры:
1. $\alpha \in$Rn(0)
2. $\forall \alpha.\alpha \in$Rn(1)
3. $(\forall \alpha.\alpha) \to (\forall \beta.\beta) \in$Rn(2), так как каждый тип вида $\forall \alpha.\alpha \in$Rn(1) то по третьему правилу весь тип $\in$Rn(2)

Теперь рассмотрим тип с поверхностными кванторами, то есть любой тип вида $\forall.\tau$ где в τ отсутсвуют кванторы. Очевидно, что любой такой тип $\in$Rn(1). Действительно, тип внутри кватнора точно имеет ранг 0. Навешивание одного или нескольких кванторов всеобщности увеличт его ранг на единицу.

2 Типовая схема

Возьмем только типы с поверхностными кванторами(из Rn(1)). Также можно формулу превратить в формулу с поверхностными кванторами.
Например:
$\beta \to \forall \alpha.\alpha \equiv \forall \alpha.(\beta \to \alpha)$

Введем определение, типовой схемой назовем выражение вида:

$\sigma \equiv \forall\alpha_1.\forall\alpha_2.....\forall\alpha_n.t$ где t∈Rn(0)

Также будем считать что $\sigma_1 <= \sigma_2$
(σ_2 является спецификацией σ_1) если:

$\sigma_2 \equiv \forall\alpha_1...\forall\alpha.\tau_1$
$\sigma_1 \equiv \forall\beta_1...\beta_n.\tau_1[\alpha_1 := \Theta_1]...[\alpha_n := \Theta_n]$
Например: $\forall\beta_1.\forall\beta_2.(\beta_1 \to \beta_2) \to (\beta_1 \to \beta_2)$
является спецификацией $\forall\alpha.\alpha \to \alpha$

3 Экзистенциальные типы

Экзистенциальные типы это способ инкапсуляции данных. Предположим что у нас есть стек с хранилищем типа α у которого определены следующие операции:

empty: α
push: $\alpha\&\nu \to \alpha$
pop: $\alpha \to \alpha\&\nu$

Тогда очевидно что тип stack≡ $\alpha\&(\alpha\&\nu \to \alpha)\&(\alpha \to \alpha\&\nu)$
Но что если мы реализовали хранилище как-то по особенному, не меняя типов операций. Мы хотим скрыть данные о реализации, в частности о типе α. Вместо деталей просто скажем что существует интерфейс, удовлетворяющий такому типу:
$\exists\alpha.\alpha\&(\alpha\&\nu \to \alpha)\&(\alpha \to \alpha\&\nu)$

Предположим, что мы захотим создать стек в котором лежвт целые числа. Рассмотрим, как тогда будет выглядеть тип созданного стека:
stack≡ $\forall\nu\exists\alpha.\alpha\&(\alpha\&\nu \to \alpha)\&(\alpha \to \alpha\&\nu)$
По аналогии с правилом удаления квантора существования, можно определить правила вывода для выражений абстрактных типов:

1 - я скрывает реализацию интерфейса 2 - я

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

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

Поскольку выводимые формулы выглядят слишком громоздко, перепишем их, вспомнив, что:
$\exists\alpha.\beta \equiv \forall\beta.(\forall\alpha.\sigma \to \beta) \to \beta$

Тогда:
pack M, θ **to** $\exists\alpha.\varphi = \Lambda\beta.\lambda x^{\forall\alpha.\varphi\to\beta}.x\theta M$
abstype α **with** $x : \varphi$ **in** M **is** $N : \psi = M\psi(\Lambda\alpha.\lambda x^{\varphi}.N)$

4 Типовая система Хиндли Милнера

Начнем с определения типа. Тип в системе Хиндли-Милнера:

Монотип - выражение в грамматике вида $\tau ::= \alpha|\tau \to \tau|(\tau)$
Политип - выражение в грамматике вида $\sigma ::= \tau|\forall\alpha.\sigma$

Поэтому типы вида $\alpha \to \forall\beta.\beta$ - некорректны в системе ХМ

Грамматика в системе Хиндли-Милнера имеет вид:

$$\Lambda ::= x|\lambda x.\Lambda|\Lambda\Lambda|(\Lambda)|\text{let x} = \Lambda \text{ in } \Lambda$$

Обозначим контекст Γ без типа x как Γ_x
В новой системе получаем следуюущие правила вывода:

1. Тавтология $\dfrac{}{\Gamma \vdash x.\phi}$

2. Уточнение $\dfrac{\Gamma \vdash e : \sigma}{\Gamma \vdash e : \sigma'}$

3. Обобщение $\dfrac{\Gamma \vdash e : \sigma}{\Gamma \vdash e : \forall\alpha.\sigma}$

4. Абстракция $\dfrac{\Gamma_x, x : \tau' \vdash e : \tau}{\Gamma \vdash \lambda x.e : \tau' \to \tau}$

5. Применение $\dfrac{\Gamma \vdash e : \tau' \to \tau \qquad \Gamma \vdash e' : \tau'}{\Gamma ee' : \tau}$

6. Let $\dfrac{\Gamma \vdash e : \sigma \qquad \Gamma_x, x : \sigma \vdash e' : \tau}{\Gamma \vdash \textbf{ let } x = e \textbf{ in } e' : \tau}$

Хотя в системе Хиндли-Милнера (как и во всех рассматриваемых нами типовых системах) нельзя типизировать $\mathcal{Y}$-комбинатор, можно добавить его, расширив язык. Давайте определим его как $\mathcal{Y}f = f\,(\mathcal{Y}f)$. Какой у него должен быть тип? Пусть $\mathcal{Y}Y$ принимает f типа α, и возвращает нечто типа β, то есть $\mathcal{Y} : \alpha \to \beta$. Функция f должна принимать то же, что возвращает $\mathcal{Y}$, так как результат $\mathcal{Y}$ передаётся в f, и возвращать она должна то же, что возвращает $\mathcal{Y}$, так как тип выражений с обоих сторон равенства должен быть одинаковый, то есть $f : \beta \to \beta$ Кроме того, α и тип f это одно и то же, $\alpha = \beta \to \beta$. После подстановки и заключения свободной переменной под квантор получаем $\mathcal{Y} : \forall\beta.(\beta \to \beta) \to \beta$.

Через такой $\mathcal{Y}$ можно определять рекурсивные функции, и они будут типизироваться.

В ХМ при рекурсии вы должны знать тип вызываемой функции(Пример с БинЛистом)