

Типы в языках программирования

Лекция 11. λ -куб и чистые системы типов (PTS)

Денис Николаевич Москвин

СПбАУ РАН

24.05.2018

- 1 Системы λ -куба: формализм
- 2 Чистые системы типов
- 3 Логический куб

- 1 Системы λ -куба: формализм
- 2 Чистые системы типов
- 3 Логический куб

- Мы не различаем заранее типы и термы.
- Множество *псевдовыражений* определяется индуктивно:

$$\mathcal{T} = V \mid C \mid \mathcal{T}\mathcal{T} \mid \lambda V : \mathcal{T}. \mathcal{T} \mid \Pi V : \mathcal{T}. \mathcal{T},$$

где V — бесконечный набор переменных, а $C = \{*, \square\}$.

- Константы $*$ и \square называют *сортами*.
- В дальнейшем переменные s, s_1, s_2 пробегают множество $\{*, \square\}$.

- *Высказывание* имеет вид $M : A$, где $M, A \in \mathcal{T}$.
- *Псевдо-контекст* — это конечное, линейно упорядоченное множество высказываний, с различными переменными в качестве субъекта.
- $\langle \rangle$ обозначает пустой контекст.
- Если $\Gamma = \langle x_1 : A_1, \dots, x_n : A_n \rangle$, то $\Gamma, y : B = \langle x_1 : A_1, \dots, x_n : A_n, y : B \rangle$.
- *Правила типизации* аксиоматизируют нотацию

$$\Gamma \vdash A : B,$$

утверждающую, что $A : B$ может быть выведено из псевдо-контекста Γ .

- В таком случае A и B называются (*допустимыми*) *выражениями*, а Γ — (*допустимым*) *контекстом*.

(аксиома)

$$\langle \rangle \vdash * : \square$$

(начальное правило)

$$\frac{\Gamma \vdash A : s}{\Gamma, x : A \vdash x : A}, \quad x \notin \Gamma$$

(правило ослабления)

$$\frac{\Gamma \vdash A : B \quad \Gamma \vdash C : s}{\Gamma, x : C \vdash A : B}, \quad x \notin \Gamma$$

(правило применения)

$$\frac{\Gamma \vdash M : (\Pi x : A. B) \quad \Gamma \vdash N : A}{\Gamma \vdash MN : B[x := N]}$$

(правило абстракции)

$$\frac{\Gamma, x : A \vdash M : B \quad \Gamma \vdash (\Pi x : A. B) : s}{\Gamma \vdash \lambda x^A. M : (\Pi x : A. B)}$$

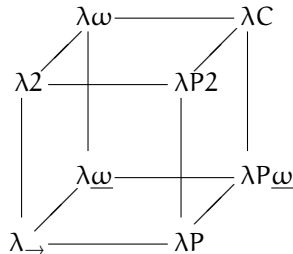
(правило преобразования)

$$\frac{\Gamma \vdash A : B \quad \Gamma \vdash B' : s \quad B =_{\beta} B'}{\Gamma \vdash A : B'}$$

$$\text{правило } (s_1, s_2) \quad \frac{\Gamma \vdash A : s_1 \quad \Gamma, x : A \vdash B : s_2}{\Gamma \vdash (\Pi x : A . B) : s_2}$$

Восемь систем λ -куба определяются конкретным подмножеством правил.

	подмножество правил			
$\lambda \rightarrow$	(*, *)			
$\lambda 2$	(*, *)	(\square , *)		
λP	(*, *)		(*, \square)	
$\lambda P 2$	(*, *)	(\square , *)	(*, \square)	
$\lambda \underline{\omega}$	(*, *)			(\square , \square)
$\lambda \omega$	(*, *)	(\square , *)		(\square , \square)
$\lambda P \underline{\omega}$	(*, *)		(*, \square)	(\square , \square)
$\lambda P \omega$	(*, *)	(\square , *)	(*, \square)	(\square , \square)



- 1 Системы λ -куба: формализм
- 2 Чистые системы типов
- 3 Логический куб

Чистые системы типов (Pure Type Systems) задают абстрактную инфраструктуру, позволяющую унифицировано описывать конкретные системы типов:

- интерпретация «высказывания-как-типы» приобретает простую форму;
- легко сравнивать свойства разных систем;
- многие свойства доказываются для целых групп систем.

Обобщения систем λ -куба, формирующие системы PTS:

- количество сортов становится произвольным (на кубе их два $*$ и \square);
- набор аксиом тоже может быть расширен (на кубе аксиома одна $*: \square$);
- сорт всего Π -типа может отличаться от сорта «возвращаемого типа».

- Множество *(пред)выражений*

$$\Lambda := V \mid C \mid \Lambda \Lambda \mid \lambda V:\Lambda. \Lambda \mid \Pi V:\Lambda. \Lambda$$

где V — множество переменных, а C — констант.

- Высказывания $M:\Lambda$, объявления $x:\Lambda$, (пред)контексты Γ — как на λ -кубе.

Спецификация конкретной PTS задаётся тройкой $S = (\mathcal{S}, \mathcal{A}, \mathcal{R})$

- \mathcal{S} — подмножество C , его элементы называют **сортами**
- \mathcal{A} — множество **аксиом** вида $c:s$, причём $c \in C$ и $s \in \mathcal{S}$
- \mathcal{R} — множество **правил** вида (s_1, s_2, s_3) , причём $s_1, s_2, s_3 \in \mathcal{S}$

V — объединение *непересекающихся* подмножеств

$$V = \bigcup_{s \in \mathcal{S}} V_s, \quad V_{s_1} \cap V_{s_2} = \emptyset, \quad V_s = \{ {}^s x, {}^s y, {}^s z, \dots \}$$

На λ -кубе, например, для выполнения последнего требования мы использовали «греко-латинскую» систему: ${}^*x, {}^*y, {}^*z, \square\alpha, \square\beta, \square\gamma$.

Аксиомы и правила для $\Gamma \vdash M : A$ (1)

Нотация присваивания типов $\Gamma \vdash M : A$ задаётся так

Аксиомы

$$\frac{}{\vdash c : s}, \text{ если } (c : s) \in \mathcal{A}$$

Начальное правило

$$\frac{\Gamma \vdash A : s}{\Gamma, x : A \vdash x : A}, \text{ если } x \equiv^s x \notin \Gamma$$

Правило ослабления

$$\frac{\Gamma \vdash M : A \quad \Gamma \vdash B : s}{\Gamma, x : B \vdash M : A}, \text{ если } x \equiv^s x \notin \Gamma$$

Здесь $s \in \mathcal{S}$, $c \in \mathcal{C}$, $x \in \mathcal{V}$ и $A, B, M \in \mathcal{L}$.

(продолжение далее...)

Аксиомы и правила для $\Gamma \vdash M : A$ (2)

Правило произведения

$$\frac{\Gamma \vdash A : s_1 \quad \Gamma, x:A \vdash B : s_2}{\Gamma \vdash (\Pi x:A. B) : s_3},$$

где $(s_1, s_2, s_3) \in \mathcal{R}$

Правило применения

$$\frac{\Gamma \vdash M : (\Pi x:A. B) \quad \Gamma \vdash N : A}{\Gamma \vdash MN : B[x := N]}$$

Правило абстракции

$$\frac{\Gamma, x:A \vdash M : B \quad \Gamma \vdash (\Pi x:A. B) : s}{\Gamma \vdash (\lambda x:A. M) : (\Pi x:A. B)}$$

Здесь $s, s_1, s_2, s_3 \in \mathcal{S}$, $x \in V$ и $A, B, M, N \in \mathcal{L}$.

(продолжение далее...)

Правило преобразования
$$\frac{\Gamma \vdash A : B \quad \Gamma \vdash B' : s \quad B =_{\beta} B'}{\Gamma \vdash A : B'}$$

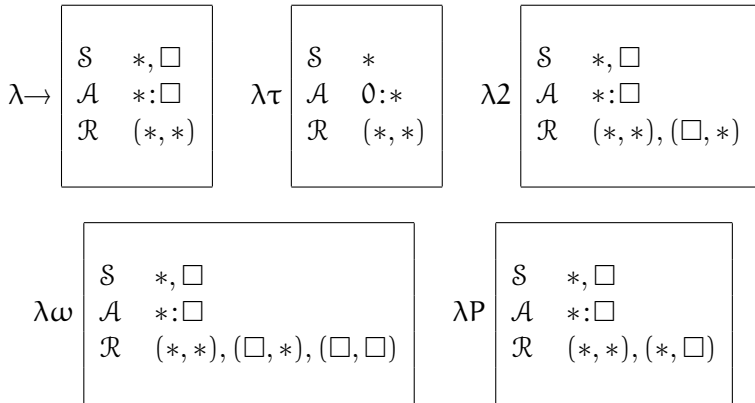
Здесь $s \in \mathcal{S}$ и $A, B, B' \in \mathcal{L}$.

Посылка $B =_{\beta} B'$ может быть неразрешима; её можно заменить на

$$B \rightarrow_{\beta} B' \vee B' \rightarrow_{\beta} B$$

Примеры PTS (1)

Принято обозначение $(s_1, s_2) \equiv (s_1, s_2, s_2)$.



$\lambda P\omega$

\mathcal{S}	$*, \square$
\mathcal{A}	$*:\square$
\mathcal{R}	$(*,*), (*, \square), (\square, *), (\square, \square)$

$\lambda C'$

\mathcal{S}	$*^t, *^p, \square$
\mathcal{A}	$*^t:\square, *^p:\square$
\mathcal{R}	\mathcal{S}^2

λECC

\mathcal{S}	$*, \{\square_i\}_{i \in \mathbb{N}^+}$
\mathcal{A}	$*:\square_i, \square_i:\square_{i+1}$
\mathcal{R}	$(*,*), (*, \square_i), (\square_i, *), (\square_i, \square_j, \square_{\max(i,j)})$

λ^*	\mathcal{S}	$*$
	\mathcal{A}	$*:*$
	\mathcal{R}	$(*,*)$

$\lambda\sqcup$	\mathcal{S}	$*, \square, \Delta$
	\mathcal{A}	$*:\square, \square:\Delta$
	\mathcal{R}	$(*,*), (\square, *), (\square, \square), (\Delta, \square), (\Delta, *)$

- Эти системы «неконсистентны» в логическом смысле: для них все типы являются обитаемыми (парадокс Жирара).

$$\lambda^* \begin{array}{l} \mathcal{S} \quad * \\ \mathcal{A} \quad *:* \\ \mathcal{R} \quad (*, *) \end{array}$$

$$\lambda\mathcal{U} \begin{array}{l} \mathcal{S} \quad *, \square, \Delta \\ \mathcal{A} \quad *:\square, \square:\Delta \\ \mathcal{R} \quad (*, *), (\square, *), (\square, \square), (\Delta, \square), (\Delta, *) \end{array}$$

- Эти системы «неконсистентны» в логическом смысле: для них все типы являются обитаемыми (парадокс Жирара).
- Если отбросить $(\Delta, *)$, получим $\lambda\mathcal{U}^-$, которая тоже неконсистентна.

$$\lambda^* \begin{array}{l} \mathcal{S} \quad * \\ \mathcal{A} \quad *:* \\ \mathcal{R} \quad (*, *) \end{array}$$

$$\lambda\mathcal{U} \begin{array}{l} \mathcal{S} \quad *, \square, \Delta \\ \mathcal{A} \quad *:\square, \square:\Delta \\ \mathcal{R} \quad (*, *), (\square, *), (\square, \square), (\Delta, \square), (\Delta, *) \end{array}$$

- Эти системы «неконсистентны» в логическом смысле: для них все типы являются обитаемыми (парадокс Жирара).
- Если отбросить $(\Delta, *)$, получим $\lambda\mathcal{U}^-$, которая тоже неконсистентна.
- А если еще отбросить (Δ, \square) , получим замечательную ЛНОЛ.

- Γ называется *(допустимым) контекстом*, если

$$\exists A, B \in \mathcal{L}. \Gamma \vdash A : B$$

- $A \in \mathcal{L}$ называется *(допустимым) выражением*, если

$$\exists \Gamma, B \in \mathcal{L}. \Gamma \vdash A : B \vee \Gamma \vdash B : A$$

- $\Gamma \vdash A : B : C$ обозначает, что одновременно выполняются утверждения $\Gamma \vdash A : B$ и $\Gamma \vdash B : C$.

- Пусть Γ — предконтекст, а A — предвыражение.
- A называется Γ -*термом*, если

$$\exists B \in \Lambda. \Gamma \vdash A : B \vee \Gamma \vdash B : A$$

- A называется Γ -*типом* (сорта s), если

$$\exists s \in \mathcal{S}. \Gamma \vdash A : s$$

- A называется Γ -*элементом* (типа B сорта s), если

$$\exists B \in \Lambda \exists s \in \mathcal{S}. \Gamma \vdash A : B : s$$

Лемма подстановки для PTS

Пусть

$$\Gamma, x:A, \Delta \vdash M : B$$

и

$$\Gamma \vdash N : A$$

тогда

$$\Gamma, \Delta[x := N] \vdash M[x := N] : B[x := N]$$

Лемма thinning для PTS

Пусть Γ и Δ — допустимые контексты, причём $\Gamma \subseteq \Delta$, тогда

$$\Gamma \vdash M : A \Rightarrow \Delta \vdash M : A$$

Лемма генерации (1)

Для $\Gamma \vdash P : Q$ по известной структуре выражения P

$$\Lambda := C \mid V \mid \Lambda \Lambda \mid \lambda V : \Lambda. \Lambda \mid \Pi V : \Lambda. \Lambda$$

эта лемма представляет свойства Γ и Q .

Лемма генерации для PTS

$$\Gamma \vdash c : Q \Rightarrow \exists s \in \mathcal{S}. \\ Q =_{\beta} s \wedge (c : s) \in \mathcal{A}$$

$$\Gamma \vdash x : Q \Rightarrow \exists s \in \mathcal{S}. \exists B =_{\beta} Q. \\ \Gamma \vdash B : s \wedge (x : B) \in \Gamma \wedge x \equiv^s x$$

...

(продолжение далее...)

Лемма генерации для PTS (продолжение)

...

$$\Gamma \vdash (\Pi x:A. B) : Q \Rightarrow \exists (s_1, s_2, s_3) \in \mathcal{R}. \\ \Gamma \vdash A : s_1 \wedge \Gamma, x:A \vdash B : s_2 \wedge Q =_{\beta} s_3$$

$$\Gamma \vdash (\lambda x:A. M) : Q \Rightarrow \exists s \in \mathcal{S}. \exists B. \\ \Gamma \vdash (\Pi x:A. B) : s \wedge \Gamma, x:A \vdash M : B \wedge Q =_{\beta} \Pi x:A. B$$

$$\Gamma \vdash (MN) : Q \Rightarrow \exists A, B. \\ \Gamma \vdash M : (\Pi x:A. B) \wedge \Gamma \vdash N : A \wedge Q =_{\beta} B[x := N]$$

Следствия леммы генерации (1)



$$\Gamma \vdash M : A \Rightarrow \exists s \in \mathcal{S}. \\ A \equiv s \vee \Gamma \vdash A : s$$



$$\Gamma \vdash M : (\Pi x : B_1. B_2) \Rightarrow \exists s_1, s_2 \in \mathcal{S}. \\ \Gamma \vdash B_1 : s_1 \wedge \Gamma, x : B_1 \vdash B_2 : s_2$$

- Если A является Γ -термом, то A — это сорт, или Γ -тип или Γ -элемент.
- Если A допустим и B — его подтерм, то B допустим.

Следствия леммы генерации (2)

- Классы сортов, Γ -типов и Γ -элементов могут пересекаться.
- Например,

$$\alpha : * \vdash (\lambda x : \alpha. x) : (\alpha \rightarrow \alpha) : *$$

$$\alpha : * \vdash (\alpha \rightarrow \alpha) : * : \square$$

Γ -терм $\alpha \rightarrow \alpha$ выступает в роли и Γ -типа и Γ -элемента.

- Есть здесь ещё смешение ролей?

Теорема о редукции субъекта для PTS

$$\Gamma \vdash M : A \wedge M \rightarrow_{\beta} M' \Rightarrow \Gamma \vdash M' : A$$

Следствия

- $\Gamma \vdash M : B \wedge B \rightarrow_{\beta} B' \Rightarrow \Gamma \vdash M : B'$
- Если A является Γ -термом и $A \rightarrow_{\beta} A'$, то A' тоже является Γ -термом.

Лемма конденсации для PTS (Condensing, Strengthening)

$$\Gamma, x:A, \Delta \vdash M : B \wedge x \notin FV(\Delta) \cup FV(M) \cup FV(B) \\ \Rightarrow \Gamma, \Delta \vdash M : B$$

Определение

PTS называется *функциональной* или *единосортной* (singly sorted), если

- 1 $(c:s_1), (c:s_2) \in \mathcal{A} \Rightarrow s_1 \equiv s_2$;
- 2 $(s_1, s_2, s_3), (s_1, s_2, s_4) \in \mathcal{R} \Rightarrow s_3 \equiv s_4$.

Все рассматриваемые нами ранее системы функциональны.

Теорема о единственности типа

Для функциональной PTS

$$\Gamma \vdash M : A \wedge \Gamma \vdash M : A' \Rightarrow A =_{\beta} A'$$

Степень терма на λ -кубе

Имеется классификация предтермов, полезная для анализа допустимых термов в системах λ -куба. Задаётся отображение $\# : \Lambda \rightarrow \{0, 1, 2, 3\}$:

$$\#(\square) = 3$$

$$\#(*) = 2$$

$$\#(\square x) = 1$$

$$\#(*x) = 0$$

$$\#(\lambda x:A. B) = \#(\Pi x:A. B) = \#B$$

$$\#(M N) = \#M$$

Для $M \in \Lambda$ значение $\#(M)$ называют *степенью* M .

Утверждение

Для всех систем λ -куба $\Gamma \vdash M : A \Rightarrow \#(M) + 1 = \#(A)$.

Определение

PTS называется *сильно нормализуемой*, если все её допустимые термы сильно нормализуемы, то есть

$$\Gamma \vdash M : A \Rightarrow SN(M) \wedge SN(A)$$

Теорема (о нормализуемости систем λ -куба)

Все системы λ -куба сильно нормализуемы.

Теорема (о разрешимости TCP и TSP для нормализуемой PTS)

Если PTS с конечным числом сортов сильно или слабо нормализуема, то TCP и TSP разрешимы.

Утверждение

Пусть λS — это PTS, расширяющая $\lambda 2$. Тогда

$$\vdash_{\lambda S} M : \perp \Rightarrow M \text{ не имеет NF}$$

Доказательство: по лемме генерации для $\vdash M : \Pi\alpha:*. \alpha$ единственный *нормальный* случай — это $\vdash \lambda\alpha:*. V : \Pi\alpha:*. \alpha$. Тогда $\alpha:* \vdash V : \alpha$, что допустимо, только если $V \equiv \alpha$. Отсюда $* \equiv \alpha$, противоречие (S не пересекается с V). ■

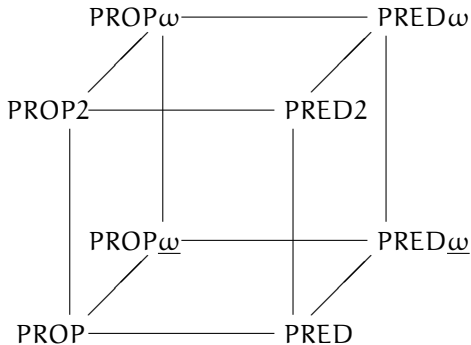
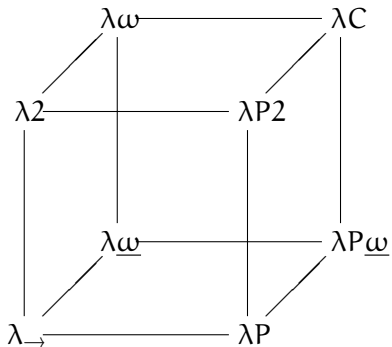
- То есть $\perp = \Pi\alpha:*. \alpha$ может быть населён только термами, не имеющими нормальной формы. Отсюда следует, что если система нормализуема, то \perp не населён.

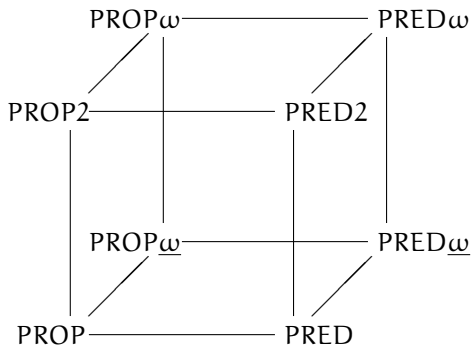
Гипотеза (1993)

Каждая слабо нормализуемая PTS сильно нормализуема.

Это — открытая проблема.

- 1 Системы λ -куба: формализм
- 2 Чистые системы типов
- 3 Логический куб





PROP proposition logic; PROP2 second-order proposition logic; $PROP_{\omega}$ weakly higher-order proposition logic; $PROP_{\omega}$ higher-order proposition logic; PRED predicate logic; PRED2 second-order predicate logic; $PRED_{\omega}$ weakly higher-order predicate logic; $PRED_{\omega}$ higher-order predicate logic.

- Системы на логическом кубе задают минимальные многосортные логики.
- Многосортные — в предикатных логиках формулы строят из атомарных разных сортов, то есть в сигнатуру входят несколько «базовых типов».
- Минимальные — только одна связка (импликация) и один квантор (общности).
- Однако, начиная со PROP2, мы можем выразить \perp , \wedge , \vee и \neg .
- Начиная с PRED2, можем выразить квантор \exists и равенство Лейбница $=_L$:

$$\begin{aligned}\exists x:S. A &\equiv \Pi \gamma:*^P. (\Pi x:S. A \rightarrow \gamma) \rightarrow \gamma \\ x =_L y &\equiv \Pi P:S \rightarrow *^P. Px \rightarrow Py\end{aligned}$$

- Пусть A — высказывание некоторой логики L .
- Пусть $[[A]]$ — интерпретация A (тип) в некоторой системе типов.
- Основные метатеоретические проблемы:
 - **корректность** (soundness):
Если A — доказуемо в L , то $[[A]]$ — обитаем.
 - **полнота** (completeness):
Если $[[A]]$ — обитаем, то A — доказуемо в L .
- **Корректность.**
Имеет место для всех вершин λ -куба.
- **Полнота.**
 - Имеет место для левой грани куба.
 - При некоторых оговорках выполняется для λP .
 - Не имеет места для $\lambda P\omega$.

- Идея доказательств такая: построить PTS максимально точно описывающую логическую систему, и доказывать полноту и корректность в два этапа.
- Например, для PRED строим эквивалентную PTS λ PRED

λP	\mathcal{S}	$*, \square$		$\lambda PRED$	\mathcal{S}	$*^P, *^S, *^f, \square^P, \square^S$
	\mathcal{A}	$*:\square$			\mathcal{A}	$*^P:\square^P, *^S:\square^S$
	\mathcal{R}	$(*,*), (*, \square)$			\mathcal{R}	$(*^P, *^P), (*^S, *^P), (*^S, \square^P),$ $(*^S, *^S, *^f), (*^S, *^f, *^f)$

а потом доказываем «равномощность» λ PRED и λP , используя подходящее «стирающее отображение».