Типы в языках программирования
Лекция 4. Обитаемость простых типов

Денис Николаевич Москвин

СПбАУ РАН

15.03.2018

План лекции

1. Обитаемость типа
2. Булева модель и ее следствия
3. Населяющие машины
4. Алгоритм Бен-Йелса
5. Алгоритм Бен-Йелса: сходимость и продуктивность

План лекции

1. Обитаемость типа
2. Булева модель и ее следствия
3. Населяющие машины
4. Алгоритм Бен-Йелса
5. Алгоритм Бен-Йелса: сходимость и продуктивность

Понятие обитаемости

Задача поиска обитателей

По заданному типу σ и контексту Γ предъявить терм M, такой что

$$\Gamma \vdash M : \sigma$$

Терм M при этом называют ***обитателем типа***, если он при этом находится в нормальной форме — ***нормальным обитателем***.

Вопросы, связанные с обитаемостью заданного типа в конкретной системе типов:

- Есть ли у него обитатели?
- Сколько разных обитателей есть у него? (имеет смысл спрашивать только про нормальных обитателей)
- Можно ли алгоритмически перечислить всех обитателей?

Аппликативная структура бестипового терма

Теорема

Лямбда-терм может иметь одну из двух форм:

$$\begin{aligned} \lambda \overrightarrow{x}.\, y\, \overrightarrow{N} &\equiv \lambda x_1 \ldots x_n .\, y\, N_1 \ldots N_k \\ \lambda \overrightarrow{x}.\, (\lambda z.\, P)\, Q\, \overrightarrow{N} &\equiv \lambda x_1 \ldots x_n .\, (\lambda z.\, P)\, Q\, N_1 \ldots N_k \end{aligned}$$

Здесь $n \geqslant 0$, $k \geqslant 0$, а переменная y может совпадать с одной из x_i, и *обязана* совпадать, если терм замкнут.

Определение

- Первая форма называется ***головной нормальной формой*** (HNF).
- Переменная y называется ***головной переменной***, а редекс $(\lambda z.\, P)\, Q$ — ***головным редексом***.
- Конструкция $\lambda x_1 \ldots x_n$ называется ***абстрактором***.

η-редукция и $\beta\eta$-нормальная форма

Определение

Бинарное отношение η-***редукции за один шаг*** $\to_\eta$ над Λ строится на основе правила

$$\lambda x.\, M\, x \quad \to_\eta \quad M$$

аналогично β-редукции. Предполагается, что $x \notin FV(M)$.

Определение

λ-терм M находится в η-***нормальной форме*** (η-NF), если в нем нет подтермов, являющихся η-редексами.

Пример

- $\lambda s\, z.\, s\, z \to_\eta \lambda s.\, s.$
- $\lambda f\, x\, y.\, f\, x\, y \to_\eta \ldots$

Типизация нормальной формы

Структура типизированной β-NF

Если M находится в β-NF, и $\Gamma \vdash M : \sigma$, то он (с точностью до α-эквивалентности) имеет вид

$$\lambda x_1^{\sigma_1} \dots x_n^{\sigma_n} . y^{\tau_1 \to \dots \to \tau_k \to \rho} N_1^{\tau_1} \dots N_k^{\tau_k}$$

При этом $\sigma = \sigma_1 \to \dots \to \sigma_n \to \rho$.

- Здесь $n \geqslant 0$, $k \geqslant 0$, а переменная y может совпадать с одной из x_i, и *обязана* совпадать, если терм замкнут.
- При этом каждый N_i находится в β-NF и

$$\Gamma, x_1 : \sigma_1, \dots, x_n : \sigma_n \vdash N_i : \tau_i$$

- Тип ρ не обязан быть переменной (то есть может быть стрелкой).

План лекции

Булева модель для типов

- Переменную типа проинтерпретируем как пробегающую значения из $\mathbb{B} = \{0, 1\}$, а стрелку $x \to y$ как $1 - x + xy$.
- ***Булевой оценкой*** (valuation) назовем функцию $\rho : \mathbb{V} \to \mathbb{B}$.
- ***Интерпретация*** $[[\sigma]]_\rho$ типа σ на оценке ρ:

$$\begin{array}{lcl} [[\alpha]]_\rho & = & \rho(\alpha); \\ [[\sigma \to \tau]]_\rho & = & [[\sigma]]_\rho \to [[\tau]]_\rho. \end{array}$$

- Оценка ρ ***удовлетворяет*** типу σ ($\rho \models \sigma$), если $[[\sigma]]_\rho = 1$.
- Оценка ρ ***удовлетворяет*** контексту Γ ($\rho \models \Gamma$), если она удовлетворяет всем типам этого контекста.
- В частности, пустому контексту удовлетворяет любая оценка.

Булева модель и населенность

Утверждение

Пусть $\Gamma \vdash M : \sigma$. Тогда $\forall \rho.\ \rho \models \Gamma \Rightarrow \rho \models \sigma$.

Доказательство: индукция по дереву вывода типа. ■

Следствие 1

Если σ населен (то есть существует M, такой что $\vdash M : \sigma$), то σ — тавтология классической логики.

Обратное неверно (например, закон Пирса). Но, если ограничится системой только с одной переменной (α), то это утверждение станет верным. (Вывести из теоремы Статмана.)

Следствие 2

Никакой тип-переменная не населен.

Теорема [Statman 1982]

Если $\mathbb{V} = \{\alpha\}$ и $\sigma = \sigma_1 \to \ldots \to \sigma_n \to \alpha$ $(n \geqslant 1)$, то

$$\sigma \text{ населен} \quad \Leftrightarrow \quad \exists i.\, \sigma_i \text{ не населен.}$$

$(\Rightarrow)$. Если все аргументы населены, передадим в качестве аргументов их обитателей и тем самым населим α.
$(\Leftarrow)$. Индукция по структуре σ. Пусть σ_i не населен.
(1): $\sigma_i = \alpha$.

$$\begin{array}{l} x_1 : \sigma_1, \ldots, x_n : \sigma_n \ \vdash \ x_i : \alpha \\ \vdash \ \lambda x_1^{\sigma_1} \ldots x_n^{\sigma_n}.\, x_i : \sigma_1 \to \ldots \to \sigma_n \to \alpha \end{array}$$

(2): $\sigma_i = \tau_1 \to \ldots \to \tau_k \to \alpha$. По (контропозиции) IH для σ_i все τ_j населены: существуют N_j, такие что $\vdash N_j : \tau_j$.

$$\begin{array}{l} x_1 : \sigma_1, \ldots, x_n : \sigma_n \ \vdash \ x_i : \tau_1 \to \ldots \to \tau_k \to \alpha \\ x_1 : \sigma_1, \ldots, x_n : \sigma_n \ \vdash \ x_i\, N_1 \ldots N_k : \alpha \\ \vdash \ \lambda x_1^{\sigma_1} \ldots x_n^{\sigma_n}.\, x_i\, N_1 \ldots N_k : \sigma_1 \to \ldots \to \sigma_n \to \alpha \end{array}$$

■

Система с 1 типовой переменной: подсчет обитателей

Теорема [Ben-Yelles 1979]

Если $\mathbb{V} = \{\alpha\}$ и $\sigma = \sigma_1 \to \ldots \to \sigma_n \to \alpha$ $(n \geqslant 1)$, то

- если все $\sigma_i = \alpha$, то число нормальных обитателей σ равно n;
- если хотя бы один σ_i стрелочный, число нормальных обитателей 0 или ∞.

Система с 1 типовой переменной: подсчет обитателей

Теорема [Ben-Yelles 1979]

Если $\mathbb{V} = \{\alpha\}$ и $\sigma = \sigma_1 \to \ldots \to \sigma_n \to \alpha$ $(n \geqslant 1)$, то

- если все $\sigma_i = \alpha$, то число нормальных обитателей σ равно n;
- если хотя бы один σ_i стрелочный, число нормальных обитателей 0 или ∞.

- первое утверждение — тривиально;
- второе — окажется простым следствием свойств общего алгоритма.

План лекции

Нормальные обитатели типов

- Какое количество разных нормальных обитателей есть у типа (с точностью до α-эквивалентности)?
- Для $\alpha \to \alpha$ такой обитатель один: $\lambda x^{\alpha}.\, x$.
- А для $(\alpha \to \beta) \to \alpha \to \beta$?

Нормальные обитатели типов

- Какое количество разных нормальных обитателей есть у типа (с точностью до α-эквивалентности)?
- Для $\alpha \to \alpha$ такой обитатель один: $\lambda x^{\alpha}.\, x$.
- А для $(\alpha \to \beta) \to \alpha \to \beta$? Два:

$$\lambda f^{\alpha\to\beta}.\, f$$

$$\lambda f^{\alpha\to\beta}\, x^{\alpha}.\, f\, x$$

- Они η-эквивалентны, второй получается из первого η-экспансией.
- Какой из них лучше с точки зрения обитаемости $(\alpha \to \beta) \to \alpha \to \beta$?

Длинные нормальные обитатели типов

Определение

Замкнутый терм $M : \sigma_1 \to \ldots \to \sigma_n \to \alpha$ находится в ***длинной нормальной форме*** (LNF), если он имеет вид

$$\lambda x_1^{\sigma_1} \ldots x_n^{\sigma_n} . x_i \, M_1 \ldots M_k$$

и все M_j тоже находятся в LNF.

- Здесь $x_i : \sigma_i$, причем $\sigma_i = \tau_1 \to \ldots \to \tau_k \to \alpha$, $k \geqslant 0$ и $M_j : \tau_j$.
- Можно расширить определение на незамкнутые термы. Тогда в головной позиции может стоять $y_j : \rho_j$ из контекста $\Gamma = y_1^{\rho_1}, \ldots, y_m^{\rho_m}$, то есть терм в LNF имеет вид

$$\lambda x_1^{\sigma_1} \ldots x_n^{\sigma_n} . y_j \, M_1 \ldots M_k$$

где $\rho_j = \zeta_1 \to \ldots \to \zeta_k \to \alpha$, $k \geqslant 0$ и $M_j : \zeta_j$.

Грамматика для построения обитателей

Зададим порождающую двухуровневую грамматику со следующими правилами вывода:

$$L(\alpha;\Gamma) \implies x\,L(\sigma_1;\Gamma)\;\ldots\;L(\sigma_n;\Gamma),$$

$$\text{если } (x:\sigma_1\to\ldots\to\sigma_n\to\alpha)\in\Gamma;$$

$$L(\sigma\to\tau;\Gamma) \implies \lambda y^{\sigma}.\,L(\tau;\Gamma,y:\sigma),$$

где переменная y — свежая и $n \geqslant 0$.

Пример вывода

$$L((\alpha\to\beta)\to(\alpha\to\beta);\emptyset) \implies \lambda f^{\alpha\to\beta}.\,L(\alpha\to\beta;\{f:\alpha\to\beta\})$$

$$\implies \lambda f^{\alpha\to\beta}.\,\lambda x^{\alpha}.\,L(\beta;\{f:\alpha\to\beta,x:\alpha\})$$

$$\implies \lambda f^{\alpha\to\beta}.\,\lambda x^{\alpha}.\,f\,L(\alpha;\{f:\alpha\to\beta,x:\alpha\}) \implies \lambda f^{\alpha\to\beta}.\,\lambda x^{\alpha}.\,f\,x$$

- Введем операцию $\twoheadrightarrow$ как рефлексивное транзитивное замыкание $\Longrightarrow$.
- Тогда продукцию с предыдущего слайда можно записать так

$$L((\alpha\to\beta)\to(\alpha\to\beta);\emptyset) \twoheadrightarrow \lambda f^{\alpha\to\beta}\, x^{\alpha}.\, f\, x$$

Утверждение

Для заданных σ, Γ и M

$$L(\sigma;\Gamma) \twoheadrightarrow M \;\Leftrightarrow\; \Gamma\vdash M:\sigma \wedge M \text{ в LNF}.$$

Доказательство: по построению.

Населяющие машины

Для каждого типа σ вывод терминалов можно описать с помощью ***населяющей машины (Inhabitation Machine)*** M_σ.

Пример: $\sigma = \alpha \to \alpha \to \alpha$

$$\begin{tikzcd} \boxed{\alpha \to \alpha \to \alpha} \arrow[rr, "\lambda x^\alpha .\, \lambda y^\alpha ."] & & \boxed{\alpha} \arrow[r] \arrow[d] & x \\ & & y & \end{tikzcd}$$

$\lambda x^\alpha .\, \lambda y^\alpha . x$
$\lambda x^\alpha .\, \lambda y^\alpha . y$

Пример: $\sigma = (\alpha \to \alpha) \to \alpha \to \alpha$

$$\begin{tikzcd} \boxed{(\alpha \to \alpha) \to \alpha \to \alpha} \arrow[rr, "\lambda f^{\alpha\to\alpha} .\, \lambda x^\alpha ."] & & \boxed{\alpha} \arrow[loop above, "f"] \arrow[r] & x \end{tikzcd}$$

$\lambda f^{\alpha\to\alpha} .\, \lambda x^\alpha .\, x$
$\lambda f^{\alpha\to\alpha} .\, \lambda x^\alpha .\, f\, x$
$\lambda f^{\alpha\to\alpha} .\, \lambda x^\alpha .\, f(f\, x)$
$\ldots$

Постройте населяющие машины для типов:

$$\alpha \to ((\gamma \to \beta) \to \alpha) \to \beta \to \alpha$$

$$((\alpha \to \beta) \to \alpha) \to \alpha$$

$$(\alpha \to \alpha) \to (\alpha \to \alpha) \to \alpha \to \alpha$$

$$(\alpha \to \beta \to \gamma) \to \beta \to \alpha \to \gamma$$

$$(\alpha \to \alpha \to \alpha) \to \alpha \to \alpha$$

$$((\alpha \to \alpha) \to \alpha) \to \alpha$$

$$(((\alpha \to \alpha) \to \alpha) \to \alpha) \to \alpha \to \alpha$$

См. также Барендрегта [BDS13] 1C, 2D.

План лекции

Определение

Определим ***глубину*** β-***NF*** так ($n \geqslant 0, k > 0$):

$$\begin{aligned} \mathrm{Depth}(\lambda x_1 \ldots x_n.\, y) &= 0 \\ \mathrm{Depth}(\lambda x_1 \ldots x_n.\, y\, N_1 \ldots N_k) &= 1 + \max_{1 \leqslant j \leqslant k} \mathrm{Depth}(N_j) \end{aligned}$$

Множество всех длинных нормальных обитателей типа τ обозначим $\mathrm{Long}(\tau)$ и сконструируем семейство подмножеств:

$$\mathrm{Long}(\tau, d) = \{M \mid M \in \mathrm{Long}(\tau), \mathrm{Depth}(M) \leqslant d\}$$

Пример

$$\begin{aligned} &\mathrm{Long}((\alpha \to \alpha) \to \alpha \to \alpha, 3) = \\ &\{\lambda s\, z.\, z, \lambda s\, z.\, s\, z, \lambda s\, z.\, s\,(s\, z), \lambda s\, z.\, s\,(s\,(s\, z))\} \end{aligned}$$

Алгоритм Бен-Йелса излагается по Хиндли [Hin97] гл.8.

Метапеременные и NF-схемы

Введем множество ***метапеременных***, отличных от всех термовых переменных. Будем обозначать их символами в верхнем регистре.

Определение

NF-схема это терм в NF, который помимо обычных термовых переменных может содержать метапеременные, причем:

- метапеременные не связываются, то есть $\lambda V.\, x\, V$ запрещено.
- Метапеременные могут стоять только в правой части аппликации, то есть $\lambda x.\, V\, x$ запрещено.
- Каждая метапеременная входит в NF-схему не более одного раза, то есть $\lambda x.\, x\, V\, V$ запрещено.

Обычный терм в NF — тоже NF-схема, но ***несобственная*** (non-proper).

Аппроксимации

Теорема о поиске (Ben-Yelles 1979)

Алгоритм поиска принимает на вход тип τ и возвращает конечную или бесконечную последовательность множеств $\mathcal{A}(\tau, d)$, при этом

- каждый элемент $\mathcal{A}(\tau, d)$ — это замкнутая типизированная длинная NF-схема типа τ, причем это
    - либо собственная NF-схема глубины d;
    - либо терм глубины $d - 1$.
- Множество $\mathcal{A}(\tau, d)$ конечно.
- $\mathrm{Long}(\tau, d) \subset \mathcal{A}(\tau, 0) \cup \ldots \cup \mathcal{A}(\tau, d+1)$.
- Введем «термовое» подмножество $\mathcal{A}_{\mathrm{term}}(\tau, d)$ множества $\mathcal{A}(\tau, d)$. Тогда

$$\mathrm{Long}(\tau) = \bigcup_{d \geqslant 0} \mathcal{A}_{\mathrm{term}}(\tau, d)$$

Пример

Пример для $\mathrm{Nat} = (\alpha \to \alpha) \to \alpha \to \alpha$

$$\mathcal{A}(\mathrm{Nat}, 0) = \{V^{(\alpha\to\alpha)\to\alpha\to\alpha}\}$$

$$\mathcal{A}(\mathrm{Nat}, 1) = \{\lambda s^{\alpha\to\alpha}\, z^{\alpha}.\, s\, V^{\alpha},\ \ \lambda s^{\alpha\to\alpha}\, z^{\alpha}.\, z\}$$

$$\mathcal{A}(\mathrm{Nat}, 2) = \{\lambda s^{\alpha\to\alpha}\, z^{\alpha}.\, s\,(s\, V^{\alpha}),\ \ \lambda s^{\alpha\to\alpha}\, z^{\alpha}.\, s\, z\}$$

$$\mathcal{A}(\mathrm{Nat}, 3) = \{\lambda s^{\alpha\to\alpha}\, z^{\alpha}.\, s\,(s\,(s\, V^{\alpha})),\ \ \lambda s^{\alpha\to\alpha}\, z^{\alpha}.\, s\,(s\, z)\}$$

$\ldots$

Алгоритм в целом

- Стартуем со слабейшей аппроксимации $\mathcal{A}(\tau, 0) = \{V^\tau\}$.
- Переход от $\mathcal{A}(\tau, d)$ к $\mathcal{A}(\tau, d+1)$ делаем по правилам:
  - Если $\mathcal{A}(\tau, d)$ пустое или не содержит NF-схем с метапеременными, закругляемся.
  - Иначе берем собственную NF-схему $X_i^\tau \in \mathcal{A}(\tau, d)$ с набором мета-переменных

    $$V_1^{\rho_1}, \ldots, V_q^{\rho_q}, \qquad (q \geqslant 1)$$

    и применяем к каждой из мета-переменных *субалгоритм раскрытия*, заменяющий каждую метапеременную на (0,1, или больше) *подходящую* NF-схему.
  - Если где-то 0, то закругляемся с этим X_i, если нет, то генерируем список всевозможных вариантов замен в семантике каждый-с-каждым для всех V_i.
- Для каждой X_i получаем в результате множество состоящее из термов глубины d и/или собственных NF-схем глубины $d+1$, формируя $\mathcal{A}(\tau, d+1)$.

Субалгоритм раскрытия (1)

- Пусть имеется мета-переменная V^{ρ} в собственной NF-схеме $X_i^{\tau} \in \mathcal{A}(\tau, d)$ и

$$\rho = \sigma_1 \to \ldots \to \sigma_m \to \alpha, \qquad (m \geqslant 0)$$

- Выбираем все $i \leqslant m$, для которых σ_i заканчивается на α

$$\sigma_i = \sigma_{i,1} \to \ldots \to \sigma_{i,n_i} \to \alpha, \qquad (n_i \geqslant 0)$$

- Для каждого i определяем ***подходящую замену*** (suitable replacement)

$$Y_i^{\rho} = \lambda x_1^{\sigma_1} \ldots x_n^{\sigma_n}.\,(x_i^{\sigma_i}\, V_{i,1}^{\sigma_{i,1}}, \ldots, V_{i,n_i}^{\sigma_{i,n_i}})^{\alpha}$$

Здесь xы и Vшки — различные свежие переменные.

Субалгоритм раскрытия (2)

- Для (единственного) вхождения мета-переменной V^ρ в $X_i^\tau \in \mathcal{A}(\tau, d)$ перечислим все открытые абстракторы

$$\lambda z_1^{\zeta_1} \dots z_t^{\zeta_t}, \qquad (t \geqslant 0)$$

- Выбираем все $j \leqslant t$, для которых ζ_j заканчивается на α

$$\zeta_j = \zeta_{j,1} \to \dots \to \zeta_{j,h_j} \to \alpha, \qquad (h_j \geqslant 0)$$

- Для каждого i определяем ***подходящую замену*** (suitable replacement)

$$Z_j^\rho = \lambda x_1^{\sigma_1} \dots x_n^{\sigma_n} . \left(z_j^{\zeta_j} V_{j,1}^{\zeta_{j,1}}, \dots, V_{j,h_j}^{\zeta_{j,h_j}}\right)^\alpha$$

Здесь xы и Vшки — различные свежие переменные.

План лекции

Обозначения

Куммулятивные аппроксимации

$$\mathcal{A}(\sigma, \leqslant d) = \mathcal{A}(\sigma, 0) \cup \ldots \cup \mathcal{A}(\sigma, d)$$

$$\mathcal{A}_{\mathrm{term}}(\sigma, \leqslant d) = \mathcal{A}_{\mathrm{term}}(\sigma, 0) \cup \ldots \cup \mathcal{A}_{\mathrm{term}}(\sigma, d)$$

Метрика 1 для типа

Обозначим через $|\sigma|$ общее число атомов в типе σ.

Метрика 2 для типа

Обозначим через $||\sigma||$ число различных атомов в типе σ.

Метрика 3 для типа

Введем обозначение $\mathbb{D}(\sigma) = |\sigma| \cdot ||\sigma||$.

Важные вспомогательные факты

Stretching Lemma

Если существует $M \in \mathrm{Long}(\sigma)$ с $\mathrm{Depth}(M) \geqslant ||\sigma||$, то в $\mathrm{Long}(\sigma)$ есть элементы высоты, превосходящей любое число, а само $\mathrm{Long}(\sigma)$ — бесконечно.

- Например, для Nat метрика $||\mathrm{Nat}|| = 1$ и

  $$\mathrm{Depth}(\lambda s^{\alpha\to\alpha}\, z^{\alpha}.\, s\, z) = 1 \geqslant ||\mathrm{Nat}||$$

  Искомые подтермы — $z : \alpha$, $s\, z : \alpha$.

- **Идея доказательства.** Если посылка выполнена, то всегда есть два подтерма M одного типа, причем один — подтерм другого. Подставляя больший вместо меньшего в большем, можем организовать бесконечный генератор.

Важные вспомогательные факты (2)

Shrinking Lemma

Если существует $M \in \mathrm{Long}(\sigma)$ с $\mathrm{Depth}(M) \geqslant \mathbb{D}(\sigma)$, то существует $\mathrm{N} \in \mathrm{Long}(\sigma)$, такой что

$$\mathbb{D}(\sigma) - \|\sigma\| \leqslant \mathrm{Depth}(\mathrm{N}) < \mathbb{D}(\sigma)$$

Идея доказательства. Та же, что и в предыдущей лемме, но подставляем меньший вместо большего. Могут возникнуть проблемы с контекстом, но аккуратный анализ структуры терма показывает, что подходящая пара найдется.

Следствие

Если существует $M \in \mathrm{Long}(\sigma)$ с $\mathrm{Depth}(M) \geqslant \mathbb{D}(\sigma)$, то существует $\mathrm{N} \in \mathrm{Long}(\sigma)$, такой что

$$\|\sigma\| \leqslant \mathrm{Depth}(\mathrm{N}) < \mathbb{D}(\sigma)$$

Вход: тип σ. Выход: число обитателей σ и перечисление $\mathrm{Long}(\sigma)$.

Реализация

- Запускаем алгоритм поиска, генерируя последовательно $\mathcal{A}(\sigma, 0)$, $\mathcal{A}(\sigma, 1)$, $\ldots$.
- Останавливаемся, достигнув $\mathcal{A}(\sigma, \mathbb{D}(\sigma))$ и строим $\mathcal{A}_{\mathrm{term}}(\sigma, \leqslant \mathbb{D}(\sigma))$ (содержит всех обитателей σ длиной меньше $\mathbb{D}(\sigma)$).
- Анализируем:
  - $\mathcal{A}_{\mathrm{term}}(\sigma, \leqslant \mathbb{D}(\sigma)) = \varnothing$. Тогда $\mathrm{Long}(\sigma) = \varnothing$.
  - $\mathcal{A}_{\mathrm{term}}(\sigma, \leqslant \mathbb{D}(\sigma))$ не пусто, но все его элементы мельче $\|\sigma\|$. Тогда $\mathrm{Long}(\sigma) = \mathcal{A}_{\mathrm{term}}(\sigma, \leqslant \mathbb{D}(\sigma))$.
  - $\mathcal{A}_{\mathrm{term}}(\sigma, \leqslant \mathbb{D}(\sigma))$ не пусто, и есть элементы не мельче $\|\sigma\|$. Число элементов бесконечно, можно продолжить перечисление.

Литература

Henk Barendregt, Wil Dekkers, and Richard Statman.
Lambda Calculus with Types.
Cambridge University Press, New York, NY, USA, 2013.

J. Roger Hindley.
Basic Simple Type Theory.
Cambridge University Press, New York, NY, USA, 1997.