Безименни термове

Трифон Трифонов

λ-смятане и теория на доказателствата, 2018/19 г.

5–12 март 2019 г.

Проблеми на $\stackrel{\alpha}{=}$

- Не е удобна за програмна реализация

Проблеми на $\stackrel{\alpha}{=}$

- Не е удобна за програмна реализация
- Неконстантна проверка за еквивалентност

Проблеми на $\stackrel{\alpha}{=}$

- Не е удобна за програмна реализация
- Неконстантна проверка за еквивалентност
- Можем ли да сменим синтаксиса като “вградим” $\stackrel{\alpha}{=}$ така, че да се превърне в $\equiv$?

Проблеми на $\stackrel{\alpha}{=}$

- Не е удобна за програмна реализация
- Неконстантна проверка за еквивалентност
- Можем ли да сменим синтаксиса като “вградим” $\stackrel{\alpha}{=}$ така, че да се превърне в $\equiv$?
- Трябва да измислим представяне, което “канонизира” свързаните променливи

Проблеми на $\stackrel{\alpha}{=}$

- Не е удобна за програмна реализация
- Неконстантна проверка за еквивалентност
- Можем ли да сменим синтаксиса като “вградим” $\stackrel{\alpha}{=}$ така, че да се превърне в $\equiv$?
- Трябва да измислим представяне, което “канонизира” свързаните променливи

Пример: $\lambda_x xz(\lambda_y yxz) \stackrel{\alpha}{=} \lambda_u uz(\lambda_v vuz)$

Граф на λ-терм

Задача

Да се дефинира формално понятието "граф, съответстващ на λ-терм".

Граф на λ-терм

Задача

Да се дефинира формално понятието "граф, съответстващ на λ-терм". Да се докаже, че два λ-терма са α-еквивалентни тогава и само тогава, когато съответните им графи са изоморфни.

Изисквания към дефиницията

Устойчивост при:

- прилагане на субституция

Изисквания към дефиницията

Устойчивост при:

- прилагане на субституция
- разглеждане на подтерм

Изисквания към дефиницията

Устойчивост при:

- прилагане на субституция
- разглеждане на подтерм
- конструиране на нов терм

Изисквания към дефиницията

Устойчивост при:

- прилагане на субституция
- разглеждане на подтерм
- конструиране на нов терм

Идея: номериране на λ-възлите отвътре навън.

Примери за безименни λ-термове

- $\lambda_x x$ $\lambda\ 0$

Примери за безименни λ-термове

- $\lambda_x x$
- $\lambda_x \lambda_y y$

$\lambda\lambda 0$

$\lambda\lambda 1$

$\lambda_{x,y}$ x

Примери за безименни λ-термове

- $\lambda_x x$
- $\lambda_x \lambda_y y$
- $\lambda_x \lambda_y x$

Примери за безименни λ-термове

$\lambda_x \lambda_z x \stackrel{\alpha}{=} \lambda_{x,y} x$

$\uparrow \beta$

- $\lambda_x x$
- $\lambda_x \lambda_y y$
- $\lambda_x \lambda_y x$
- $\lambda(\lambda 0)(\lambda 1)$

$\lambda_x (\lambda_y y)(\lambda_z x)$

Примери за безименни λ-термове

- $\lambda_x x$
- $\lambda_x \lambda_y y$
- $\lambda_x \lambda_y x$
- $\lambda(\lambda 0)(\lambda 1)$
- $\lambda_x x(\lambda_z z(\lambda_y xyz)x)$

Свободни променливи

Как да представяме свободните променливи?

- лесно свързване на променливи с еднакви имена при конструиране на нов терм

Свободни променливи

Как да представяме свободните променливи?

- лесно свързване на променливи с еднакви имена при конструиране на нов терм
- **Идея:** да ги представяме с подходящо число

Свободни променливи

Как да представяме свободните променливи?

- лесно свързване на променливи с еднакви имена при конструиране на нов терм
- **Идея:** да ги представяме с подходящо число
- **Проблем:** всяко срещане на x може да е скрито под различен брой λ-абстракции!

Свободни променливи

Как да представяме свободните променливи?

- лесно свързване на променливи с еднакви имена при конструиране на нов терм
- **Идея:** да ги представяме с подходящо число
- **Проблем:** всяко срещане на x може да е скрито под различен брой λ-абстракции!
- **Инвариант:** разликата между номера на променливата и броя λ-абстракции, зад които е скрита

Свободни променливи

Как да представяме свободните променливи?

- лесно свързване на променливи с еднакви имена при конструиране на нов терм
- **Идея:** да ги представяме с подходящо число
- **Проблем:** всяко срещане на x може да е скрито под различен брой λ-абстракции!
- **Инвариант:** разликата между номера на променливата и броя λ-абстракции, зад които е скрита
- **индекс на de Bruijn**

Примери за безименни термове със свободни променливи

- $\lambda_x y$

Примери за безименни термове със свободни променливи

- $\lambda_x y$
- $y(\lambda_x xyz)z$

Примери за безименни термове със свободни променливи

- $\lambda_x y$
- $y(\lambda_x xyz)z$
- $(\lambda_x xy(\lambda_u zu(\lambda_v vy))z)$

Примери за безименни термове със свободни променливи

- $\lambda_x y$
- $y(\lambda_x xyz)z$
- $(\lambda_x xy(\lambda_u zu(\lambda_v vy))z)$
- $\lambda(\lambda 0((\lambda 1)21))1$ $\to 0$ $\qquad 0 \to z$

Безименни термове

Дефиниция (Безименни термове, Λ_n, Λ^*)

С едновременна индукция за всяко n дефинираме множествата Λ_n от безименни термове с не повече от n различни свободни променливи.

Безименни термове

Дефиниция (Безименни термове, Λ_n, Λ^*)

С едновременна индукция за всяко n дефинираме множествата Λ_n от безименни термове с не повече от n различни свободни променливи.

1. $i \in \Lambda_n$ за всяко естествено число i, за което $0 \leq i < n$.

Безименни термове

Дефиниция (Безименни термове, Λ_n, Λ^*)

С едновременна индукция за всяко n дефинираме множествата Λ_n от безименни термове с не повече от n различни свободни променливи.

1. $i \in \Lambda_n$ за всяко естествено число i, за което $0 \leq i < n$.
2. Ако $M, N \in \Lambda_n$, то $(MN) \in \Lambda_n$ е апликацията на M над N.

Ако $M \in \Lambda_{n+1}$, то $(\lambda M) \in \Lambda_n$

Безименни термове

Дефиниция (Безименни термове, Λ_n, Λ^*)

С едновременна индукция за всяко n дефинираме множествата Λ_n от безименни термове с не повече от n различни свободни променливи.

1. $i \in \Lambda_n$ за всяко естествено число i, за което $0 \leq i < n$.
2. Ако $M, N \in \Lambda_n$, то $(MN) \in \Lambda_n$ е апликацията на M над N.
3. Ако $M \in \Lambda_{n+1}$, то $(\lambda M) \in \Lambda_n$ е абстракцията над променливата с индекс 0 в M.

Безименни термове

Дефиниция (Безименни термове, Λ_n, Λ^*)

С едновременна индукция за всяко n дефинираме множествата Λ_n от безименни термове с не повече от n различни свободни променливи.

1. $i \in \Lambda_n$ за всяко естествено число i, за което $0 \leq i < n$.
2. Ако $M, N \in \Lambda_n$, то $(MN) \in \Lambda_n$ е апликацията на M над N.
3. Ако $M \in \Lambda_{n+1}$, то $(\lambda M) \in \Lambda_n$ е абстракцията над променливата с индекс 0 в M.

С $\Lambda^* := \bigcup_{n=0}^{\infty} \Lambda_n$ отбелязваме множеството на всички безименни λ-термове.

Безименни термове

Дефиниция (Безименни термове, Λ_n, Λ^*)

С едновременна индукция за всяко n дефинираме множествата Λ_n от безименни термове с не повече от n различни свободни променливи.

1. $i \in \Lambda_n$ за всяко естествено число i, за което $0 \leq i < n$.
2. Ако $M, N \in \Lambda_n$, то $(MN) \in \Lambda_n$ е апликацията на M над N.
3. Ако $M \in \Lambda_{n+1}$, то $(\lambda M) \in \Lambda_n$ е абстракцията над променливата с индекс 0 в M.

С $\Lambda^* := \bigcup_{n=0}^{\infty} \Lambda_n$ отбелязваме множеството на всички безименни λ-термове.

Задача

Ако $m < n$, то $\Lambda_m \subsetneq \Lambda_n$.

Безименни термове

Дефиниция (Безименни термове, Λ_n, Λ^*)

С едновременна индукция за всяко n дефинираме множествата Λ_n от безименни термове с не повече от n различни свободни променливи.

1. $i \in \Lambda_n$ за всяко естествено число i, за което $0 \leq i < n$.
2. Ако $M, N \in \Lambda_n$, то $(MN) \in \Lambda_n$ е апликацията на M над N.
3. Ако $M \in \Lambda_{n+1}$, то $(\lambda M) \in \Lambda_n$ е абстракцията над променливата с индекс 0 в M.

С $\Lambda^* := \bigcup_{n=0}^{\infty} \Lambda_n$ отбелязваме множеството на всички безименни λ-термове.

Задача

Ако $m < n$, *то* $\Lambda_m \subsetneq \Lambda_n$.

Можем да проверим, че Λ^* е изоморфно на $\Lambda_{/\overset{\alpha}{\equiv}}$.

Изтриване и добавяне на имена

Ще дефинираме две изображения:

Изтриване и добавяне на имена

Ще дефинираме две изображения:

- $\sharp(M)$ — добавя имена към безименния терм M

Изтриване и добавяне на имена

Ще дефинираме две изображения:

- $\sharp(M)$ — добавя имена към безименния терм M
- $\flat(M)$ — изтрива имената на λ-терма M

Изтриване и добавяне на имена

Ще дефинираме две изображения:

- $\sharp(M)$ — добавя имена към безименния терм M
- $\flat(M)$ — изтрива имената на λ-терма M

Дефиниция (Контекст от имена)

Крайна редица от различни променливи $\Gamma := x_{n-1}, \ldots, x_0 \in V$ ще наричаме *контекст от имена*.

Изтриване и добавяне на имена

Ще дефинираме две изображения:

- $\sharp(M)$ — добавя имена към безименния терм M
- $\flat(M)$ — изтрива имената на λ-терма M

Дефиниция (Контекст от имена)

Крайна редица от различни променливи $\Gamma := x_{n-1}, \ldots, x_0 \in V$ ще наричаме *контекст от имена*.
Дължината на контекста ще бележим с $|\Gamma| := n$, а множеството $\operatorname{dom}\Gamma := \{x_{n-1}, \ldots, x_0\}$ ще наричаме домейн на контекста.

Изтриване и добавяне на имена

Ще дефинираме две изображения:

- $\sharp(M)$ — добавя имена към безименния терм M
- $\flat(M)$ — изтрива имената на λ-терма M

Дефиниция (Контекст от имена)

Крайна редица от различни променливи $\Gamma := x_{n-1}, \ldots, x_0 \in V$ ще наричаме *контекст от имена*.
Дължината на контекста ще бележим с $|\Gamma| := n$, а множеството $\operatorname{dom}\Gamma := \{x_{n-1}, \ldots, x_0\}$ ще наричаме домейн на контекста.
При даден контекст Γ, ще казваме, че променливата x_i има индекс i.

Изоморфизъм между $\Lambda_{/\stackrel{\alpha}{=}}$ и Λ^*

Дефиниция

Нека $X \subseteq V$ е множество от променливи. Дефинираме $\Lambda_X := \{M \in \Lambda \mid \mathrm{FV}(M) \subseteq X\}$.

Изоморфизъм между $\Lambda_{/\stackrel{\alpha}{\equiv}}$ и Λ^*

Дефиниция

Нека $X \subseteq V$ е множество от променливи. Дефинираме $\Lambda_X := \{M \in \Lambda \mid \mathrm{FV}(M) \subseteq X\}$.

Задача

Да се дефинират фамилиите от изображения $\sharp_\Gamma : \Lambda_{|\Gamma|} \to \Lambda_{\mathrm{dom}\,\Gamma}$ *и* $\flat_\Gamma : \Lambda_{\mathrm{dom}\,\Gamma} \to \Lambda_{|\Gamma|}$ *за даден контекст от имена* $\Gamma := x_{n-1}, \ldots, x_0$, *които извършват превод между термове с имена и безименни термове.*

Изоморфизъм между $\Lambda_{/\overset{\alpha}{\equiv}}$ и Λ^*

Дефиниция

Нека $X \subseteq V$ е множество от променливи. Дефинираме $\Lambda_X := \{M \in \Lambda \mid \mathrm{FV}(M) \subseteq X\}$.

Задача

Да се дефинират фамилиите от изображения $\sharp_\Gamma : \Lambda_{|\Gamma|} \to \Lambda_{\mathrm{dom}\,\Gamma}$ *и* $\flat_\Gamma : \Lambda_{\mathrm{dom}\,\Gamma} \to \Lambda_{|\Gamma|}$ *за даден контекст от имена* $\Gamma := x_{n-1}, \ldots, x_0$*, които извършват превод между термове с имена и безименни термове. Да се покаже, че:*

1. $\sharp_\Gamma(\flat_\Gamma(M)) \overset{\alpha}{\equiv} M$ *за всеки терм* $M \in \Lambda_{\mathrm{dom}\,\Gamma}$

Изоморфизъм между $\Lambda_{/\overset{\alpha}{\equiv}}$ и Λ^*

Дефиниция

Нека $X \subseteq V$ е множество от променливи. Дефинираме $\Lambda_X := \{M \in \Lambda \mid \mathsf{FV}(M) \subseteq X\}$.

Задача

Да се дефинират фамилиите от изображения $\sharp_\Gamma : \Lambda_{|\Gamma|} \to \Lambda_{\mathrm{dom}\,\Gamma}$ *и* $\flat_\Gamma : \Lambda_{\mathrm{dom}\,\Gamma} \to \Lambda_{|\Gamma|}$ *за даден контекст от имена* $\Gamma := x_{n-1}, \ldots, x_0$*, които извършват превод между термове с имена и безименни термове. Да се покаже, че:*

1. $\sharp_\Gamma(\flat_\Gamma(M)) \overset{\alpha}{\equiv} M$ *за всеки терм* $M \in \Lambda_{\mathrm{dom}\,\Gamma}$
2. $\flat_\Gamma(\sharp_\Gamma(M)) \equiv M$ *за всеки терм* $M \in \Lambda_{|\Gamma|}$

Субституция в λ^*

Какъв би трябвало да е резултатът от:

- $\lambda 01[1 \mapsto \lambda 0]$ $\equiv \lambda 01$

$[z \mapsto \lambda u\, u]$

$\sharp_{z,x}(\lambda 0 1) \equiv \lambda y\, y x$

0

Субституция в λ^*

Какъв би трябвало да е резултатът от:

- $\lambda 01[1 \mapsto \lambda 0]$
- $\lambda 01[0 \mapsto \lambda 0] \equiv \lambda 0(\lambda 0)$

$(\lambda y\, yx)[x \mapsto \lambda u\, u] \equiv \lambda y\, y(\lambda u\, u)$

Субституция в λ^*

Какъв би трябвало да е резултатът от:

- $\lambda 01[1 \mapsto \lambda 0]$
- $\lambda 01[0 \mapsto \lambda 0]$
- $0(\lambda 01)[0 \mapsto \lambda 0] \equiv (\lambda 0)(\lambda 0\,(\lambda 0))$

$(\lambda_x x)(\lambda_y y\,(\lambda_u u))$

Субституция в λ^*

Какъв би трябвало да е резултатът от:

- $\lambda 01[1 \mapsto \lambda 0]$
- $\lambda 01[0 \mapsto \lambda 0]$
- $0(\lambda 01)[0 \mapsto \lambda 0]$
- $0(\lambda 01)[0 \mapsto 1]$ $= 1(\lambda 0 2)$

$$(M_1 M_2)[i \mapsto N] := (M_1[i \mapsto N]\ M_2[i \mapsto N])$$

$$i[j \mapsto N] := \begin{cases} N, & i = j \\ i, & i \neq j \end{cases}$$

$$(\lambda M)[i \mapsto N] := \lambda(M[i+1 \mapsto \uparrow^1 N])$$

$$\uparrow^1 (i) := i+1$$

$$\uparrow^1 (M_1 M_2) := (\uparrow^1 M_1)(\uparrow^1 M_2)$$

$$\uparrow^1 (\lambda M') := \lambda (\uparrow^1 M')$$

Субституция в λ^*

Какъв би трябвало да е резултатът от:

- $\lambda 01[1 \mapsto \lambda 0]$
- $\lambda 01[0 \mapsto \lambda 0]$
- $0(\lambda 01)[0 \mapsto \lambda 0]$
- $0(\lambda 01)[0 \mapsto 1]$
- $0(\lambda 01)[0 \mapsto \lambda 01] \equiv (\lambda 01)(\lambda 0(\lambda 02))$

Изместване

Дефиниция (Изместване)

Дефинираме $\uparrow_c^d(M) : \Lambda_n \to \Lambda_{n+d}$:

1. $\uparrow_c^d(k) := \begin{cases} k, & \text{за } 0 \leq k < c, \\ k+d, & \text{за } c \leq k < n. \end{cases}$

Изместване

Дефиниция (Изместване)

Дефинираме $\uparrow_c^d (M) : \Lambda_n \to \Lambda_{n+d}$:

1. $\uparrow_c^d (k) := \begin{cases} k, & \text{за } 0 \leq k < c, \\ k + d, & \text{за } c \leq k < n. \end{cases}$
2. $\uparrow_c^d (M_1 M_2) := (\uparrow_c^d (M_1))(\uparrow_c^d (M_2))$

Изместване

Дефиниция (Изместване)

Дефинираме $\uparrow^d_c(M) : \Lambda_n \to \Lambda_{n+d}$:

1. $\uparrow^d_c(k) := \begin{cases} k, & \text{за } 0 \leq k < c, \\ k+d, & \text{за } c \leq k < n. \end{cases}$
2. $\uparrow^d_c(M_1 M_2) := (\uparrow^d_c(M_1))(\uparrow^d_c(M_2))$
3. $\uparrow^d_c(\lambda N) := \lambda \uparrow^d_{c+1}(N)$

Изместване

Дефиниция (Изместване)

Дефинираме $\uparrow_c^d (M) : \Lambda_n \to \Lambda_{n+d}$:

1. $\uparrow_c^d (k) := \begin{cases} k, & \text{за } 0 \leq k < c, \\ k + d, & \text{за } c \leq k < n. \end{cases}$
2. $\uparrow_c^d (M_1 M_2) := (\uparrow_c^d (M_1))(\uparrow_c^d (M_2))$
3. $\uparrow_c^d (\lambda N) := \lambda \uparrow_{c+1}^d (N)$

Дефинираме $\uparrow^d (M) := \uparrow_0^d (M)$.

Субституция в Λ^*

Дефиниция

Нека $M, N \in \Lambda_n$ и $k \in \mathbb{N}$.

Субституция в Λ^*

Дефиниция

Нека $M, N \in \Lambda_n$ и $k \in \mathbb{N}$.

1. $k[k \mapsto N] := N$

Субституция в Λ^*

Дефиниция

Нека $M, N \in \Lambda_n$ и $k \in \mathbb{N}$.

1. $k[k \mapsto N] := N$
2. $i[k \mapsto N] := i$ за $i \neq k$

Субституция в Λ^*

Дефиниция

Нека $M, N \in \Lambda_n$ и $k \in \mathbb{N}$.

1. $k[k \mapsto N] := N$
2. $i[k \mapsto N] := i$ за $i \neq k$
3. $(M_1 M_2)[k \mapsto N] := (M_1[k \mapsto N])(M_2[k \mapsto N])$

Субституция в Λ^*

Дефиниция

Нека $M, N \in \Lambda_n$ и $k \in \mathbb{N}$.

1. $k[k \mapsto N] := N$
2. $i[k \mapsto N] := i$ за $i \neq k$
3. $(M_1 M_2)[k \mapsto N] := (M_1[k \mapsto N])(M_2[k \mapsto N])$
4. $(\lambda M)[k \mapsto N] := \lambda(M[k+1 \mapsto \uparrow^1 (N)])$

Субституция в Λ^*

Дефиниция

Нека $M, N \in \Lambda_n$ и $k \in \mathbb{N}$.

1. $k[k \mapsto N] := N$
2. $i[k \mapsto N] := i$ за $i \neq k$
3. $(M_1 M_2)[k \mapsto N] := (M_1[k \mapsto N])(M_2[k \mapsto N])$
4. $(\lambda M)[k \mapsto N] := \lambda(M[k+1 \mapsto \uparrow^1 (N)])$

Примери:

Субституция в Λ^*

Дефиниция

Нека $M, N \in \Lambda_n$ и $k \in \mathbb{N}$.

1. $k[k \mapsto N] := N$
2. $i[k \mapsto N] := i$ за $i \neq k$
3. $(M_1 M_2)[k \mapsto N] := (M_1[k \mapsto N])(M_2[k \mapsto N])$
4. $(\lambda M)[k \mapsto N] := \lambda(M[k+1 \mapsto \uparrow^1 (N)])$

Примери:

- $(y(\lambda_x xyz)z)[y \mapsto zy]$

Субституция в Λ^*

Дефиниция

Нека $M, N \in \Lambda_n$ и $k \in \mathbb{N}$.

1. $k[k \mapsto N] := N$
2. $i[k \mapsto N] := i$ за $i \neq k$
3. $(M_1 M_2)[k \mapsto N] := (M_1[k \mapsto N])(M_2[k \mapsto N])$
4. $(\lambda M)[k \mapsto N] := \lambda(M[k+1 \mapsto \uparrow^1 (N)])$

Примери:

- $(y(\lambda_x xyz)z)[y \mapsto zy]$
- $0(\lambda 013)2)[0 \mapsto 20]$

Субституция в Λ^*

Дефиниция

Нека $M, N \in \Lambda_n$ и $k \in \mathbb{N}$.

1. $k[k \mapsto N] := N$
2. $i[k \mapsto N] := i$ за $i \neq k$
3. $(M_1 M_2)[k \mapsto N] := (M_1[k \mapsto N])(M_2[k \mapsto N])$
4. $(\lambda M)[k \mapsto N] := \lambda(M[k+1 \mapsto \uparrow^1 (N)])$

Примери:

- $(y(\lambda_x xyz)z)[y \mapsto zy]$
- $0(\lambda 013)2)[0 \mapsto 20]$
- $y(\lambda_x xy(\lambda_u uxy))[y \mapsto z(\lambda_v vyz)]$

Субституция в Λ^*

Дефиниция

Нека $M, N \in \Lambda_n$ и $k \in \mathbb{N}$.

1. $k[k \mapsto N] := N$
2. $i[k \mapsto N] := i$ за $i \neq k$
3. $(M_1 M_2)[k \mapsto N] := (M_1[k \mapsto N])(M_2[k \mapsto N])$
4. $(\lambda M)[k \mapsto N] := \lambda(M[k + 1 \mapsto \uparrow^1 (N)])$

Примери:

- $(y(\lambda_x xyz)z)[y \mapsto zy]$
- $0(\lambda 013)2)[0 \mapsto 20]$
- $y(\lambda_x xy(\lambda_u uxy))[y \mapsto z(\lambda_v vyz)]$
- $0(\lambda 01(\lambda 012))[0 \mapsto 2(\lambda 013)]$

Съвместимост на субституциите

Задача

Да се провери, че двете дефиниции за субституции са съгласувани относно изображенията $\sharp_\Gamma$ *и* $\flat_\Gamma$*.*

Съвместимост на субституциите

Задача

Да се провери, че двете дефиниции за субституции са съгласувани относно изображенията $\sharp_\Gamma$ *и* $\flat_\Gamma$*.*
За целта, нека фиксираме контекст от имена $\Gamma := x_{n-1}, \ldots, x_0$*.*

Съвместимост на субституциите

Задача

Да се провери, че двете дефиниции за субституции са съгласувани относно изображенията $\sharp_\Gamma$ *и* $\flat_\Gamma$.
За целта, нека фиксираме контекст от имена $\Gamma := x_{n-1}, \ldots, x_0$.
Да се покаже, че

1. $\sharp_\Gamma(M)[x_i \mapsto \sharp_\Gamma(N)] \stackrel{\alpha}{=} \sharp_\Gamma(M[i \mapsto N])$ *за произволни* $M, N \in \Lambda_n$,

Съвместимост на субституциите

Задача

Да се провери, че двете дефиниции за субституции са съгласувани относно изображенията $\sharp_\Gamma$ и $\flat_\Gamma$.
За целта, нека фиксираме контекст от имена $\Gamma := x_{n-1}, \ldots, x_0$.
Да се покаже, че

1. $\sharp_\Gamma(M)[x_i \mapsto \sharp_\Gamma(N)] \overset{\alpha}{=} \sharp_\Gamma(M[i \mapsto N])$ *за произволни* $M, N \in \Lambda_n$,
2. $\flat_\Gamma(M)[i \mapsto \flat_\Gamma(N)] \equiv \flat_\Gamma(M[x_i \mapsto N])$ *за произволни* $M, N \in \Lambda_{\mathrm{dom}\,\Gamma}$.

Съвместимост на субституциите

Задача

Да се провери, че двете дефиниции за субституции са съгласувани относно изображенията $\sharp_\Gamma$ *и* $\flat_\Gamma$.
За целта, нека фиксираме контекст от имена $\Gamma := x_{n-1}, \ldots, x_0$.
Да се покаже, че

1. $\sharp_\Gamma(M)[x_i \mapsto \sharp_\Gamma(N)] \overset{\alpha}{=} \sharp_\Gamma(M[i \mapsto N])$ *за произволни* $M, N \in \Lambda_n$,
2. $\flat_\Gamma(M)[i \mapsto \flat_\Gamma(N)] \equiv \flat_\Gamma(M[x_i \mapsto N])$ *за произволни* $M, N \in \Lambda_{\mathrm{dom}\,\Gamma}$.

Задача

Да се направи програмна реализация на субституция над безименни термове.