Задача 2.3

- $(A, A) \in \overset{\alpha}{=}$, $A \in \Lambda$
- ако $(A_1, A_2) \in \overset{\alpha}{=}$, то $(A_2, A_1) \in \overset{\alpha}{=}$, $A_1, A_2 \in \Lambda$
- ако $(A_1, A_2) \in \overset{\alpha}{=}$ и $(A_2, A_3) \in \overset{\alpha}{=}$, то $(A_1, A_3) \in \overset{\alpha}{=}$, $A_1, A_2, A_3 \in \Lambda$
- ако $(A_1, B_1) \in \overset{\alpha}{=}$ и $(A_2, B_2) \in \overset{\alpha}{=}$, то $(A_1B_1, A_2B_2) \in \overset{\alpha}{=}$, $A_1, A_2, B_1, B_2 \in \Lambda$
- Нека имаме $M = \lambda_x A$, $N = \lambda_y B$, $A, B \in \Lambda$ и $x, y \in V$. Ако $(A, B[y \rightsquigarrow x]) \in \overset{\alpha}{=}$ и $y \notin FV(A) \cup BV(A)$, то $(M, N) \in \overset{\alpha}{=}$ (тук $\rightsquigarrow$ е наивна субституция).

Задача 2.1

(1)

Ще напрваим индукция по дефиницията на частичната субституция.

1. $M \equiv x$. В този случай $M[x \rightsquigarrow N] \equiv N$ и $M[x \hookrightarrow N] \equiv N$.
2. $M \equiv y$, $y \not\equiv x$. В този случай $M[x \rightsquigarrow N] \equiv y$ и $M[x \hookrightarrow N] \equiv y$.
3. $M \equiv M_1M_2$. Тогава $M[x \rightsquigarrow N] = (M_1[x \rightsquigarrow N])(M_2[x \rightsquigarrow]N)$ и $M[x \hookrightarrow N] = (M_1[x \hookrightarrow N])(M_2[x \hookrightarrow N])$. От ИП $(M_1[x \rightsquigarrow N]) \equiv (M_1[x \hookrightarrow N])$ и $(M_2[x \rightsquigarrow N]) \equiv (M_2[x \hookrightarrow N])$. Също от ИП знаем, че двете частични субституции са дефинирани. Така получаваме, че $M[x \rightsquigarrow N] \equiv M[x \hookrightarrow N]$, а също сме и сигурни, че $M[x \hookrightarrow N]$ е дефинирано.
4. $M = \lambda_x P$. Тогава $M[x \rightsquigarrow N] \equiv \lambda_x P$ и $M[x \hookrightarrow N] \equiv \lambda_x P$, така получаваме, че $M[x \rightsquigarrow N] \equiv M[x \hookrightarrow N]$.
5. $M = \lambda_y P$ за $y \not\equiv x$. Ясно е, че ако частичната субституция е дефинирана в този случай, то резултатите от двете субституции ще съвпадат. Това, което трябва да се покаже е, че частичната субституция е дефинирана в този случай. По-конкретно трябва да покажем, че $x \notin FV(P)$ или $y \notin FV(N)$. Ще използваме допускането от задачата, че $FV(N) \cap BV(M) = \{\}$.

От дефиницията на BV може да се види, че $y \in BV(M)$. Понеже $FV(N) \cap BV(M) = \{\}$, то излиза, че $y \notin FV(N)$. Това показва, че частичната субституция е дефинирана.

Също така от ИП имаме, че $P[x \hookrightarrow N] \equiv P[x \rightsquigarrow N]$, от което и получаваме, че $[x \hookrightarrow N] \equiv [x \rightsquigarrow N]$

(2)

Да разгледаме терма $M = \lambda_y \lambda_x y$. Искаме да направим субституция като заместим x с y. Забелязваме, че частичната субституция $M[x \hookrightarrow y]$ е дефинирана и дава $\lambda_y \lambda_x y$. Но наивната субституция не е коректна, понеже $BV(M) = \{x, y\}$, а $FV(N) = \{y\}$ (тук N е y) и съответно нямат празно сечение.

Задача 2.23

Нека дефинираме $c_i = \lambda_n n c_s c_0$

1. Да се докаже, че за произволно $n \in \mathbb{N}$ е изпълнено $c_i c_n \stackrel{\beta}{=} c_n$.
2. Вярно ли е, че $c_i \stackrel{\beta\eta}{=} I$?

1

Ще докажем твърдението с индукция $n \in \mathbb{N}$. Преди това ще забележим, че $c_i c_n \stackrel{\beta}{=} c_n c_s c_0 = (\lambda_f \lambda_x f^n x) c_s c_n \stackrel{\beta}{=} c_s^n c_0$.

База

При $n = 0$ имаме, $c_i c_0 \stackrel{\beta}{==} c_0 c_s c_0 = (\lambda_f \lambda_x x) c_s c_0 \stackrel{\beta}{=} c_0$, с което базата е доказана.

Индуктивна стъпка

Нека се опитаме да докажем твърдението за $n+1$. $c_i c_{n+1} \overset{\beta}{=} c_{n+1} c_s c_0 = (\lambda_f \lambda_x f^{n+1} x) c_s c_0 \overset{\beta}{=} c_s^{n+1} c_0$. Използваме дефиницията за n-кратна композиция на функция и записваме, че $c_s^{n+1} c_0 = c_s(c_s^n c_0)$. От наблюдението горе се сещаме, че $c_s(c_s^n c_0) \overset{\beta}{=} c_s(c_i c_n)$. От ИП можем да запишем, че $c_s(c_i c_n) \overset{\beta}{=} c_s c_n$. Сега от свойствата на c_s получаваме, че $c_s c_n = c_{n+1}$. По този начин индуктивната стъпка е завършена.

2

Нека разгледаме $K = \lambda_x \lambda_y x$. Нека приложим K на c_i и на I. $IK \overset{\beta}{=} K = A_1$. От друга страна $c_i K \overset{\beta}{=} K c_s c_0 \overset{\beta}{=} c_s = A_2$. Използваме дефиницията за c_s от лекции $c_s = \lambda_n \lambda_f \lambda_x f(nfx)$. Вижда се, че A_1 и A_2 са два фундаментално различни обекта, понеже ако приложим променлива a към A_1 получаваме $A_1 a = Ka \overset{\beta}{=} \lambda_y a$, което е константа функция, а ако се опитаме да приложим променливата a към A_2 ще се получи грешка, понеже $A_2 a \overset{\beta}{=} \lambda_f \lambda_x f(afx)$. Това няма как да е валидно, понеже изисква a да бъде функция.

Задача 2.32

В тази задача ще използваме наготово аритметиката с нумералите на Church като ще приемем, че имаме работещи основни аритметични функции за числата, както и за тяхното представяне. Също и ще приемем, че имаме работещи основните неща от булевата логика. За момента ще покажем минимизация само за обикновени функции от вида $f : \mathbb{N} \to \mathbb{N}$. Тоест няма да имаме "константни аргументи".

Нека разгледаме терма $\Gamma = \lambda_F \lambda_f \lambda_y (c_{=}(f\ y) c_0) y (F f (c_s\ y))$. Тук функцията c_s дава следващото число, тоест $c_s c_x \overset{\beta}{=} c_{x+1}$. От сега нататък приемаме, че искаме да минимизираме функцията $f_1 \in \Lambda$. Приемаме, че тя има следната семантика $f_1 c_x \overset{\beta}{=} c_y \iff f_2(x) = y$. Термът който ще минимизира f_1 ще бъде $s_1 := Y \Gamma f_1$. Ще докажем, че нашият терм s_1 има се-

мантиката на функцията $s_2(x) = \begin{cases} x & f_2(x) = 0 \\ s_2(x+1) & \exists y > x : f_2(y) = 0 \\ \text{недефинирана} & \nexists y > x : f_2(y) = 0 \end{cases}$.
(тоест, че $s_1 c_x \overset{\beta}{=} c_y \iff s_2(x) = y$). Също и ще покажем, че функцията s_2 всъщност извършва минимизацията на f_2.

Случай 1: $f_1 c_x \overset{\beta}{=} c_0 \iff f_2(x) = 0$

Трябва да покажем, че $s_1 c_x \overset{\beta}{=} c_x$.
$s_1 c_x \equiv (Y\Gamma f_1)c_x \equiv ((\lambda_f(\lambda_x f(xx))(\lambda_x f(xx)))\Gamma f_1)c_x \overset{\beta}{=} (((\lambda_x \Gamma(xx))(\lambda_x \Gamma(xx)))f_1)c_x \overset{\beta}{=} \Gamma((\lambda_x \Gamma(xx))(\lambda_x \Gamma(xx)))f_1 c_x$. Нека $\Delta := ((\lambda_x \Gamma(xx))(\lambda_x \Gamma(xx)))$.
Тогава $\Gamma((\lambda_x \Gamma(xx))(\lambda_x \Gamma(xx)))f_1 c_x \equiv \Gamma \Delta f_1 c_x \equiv \lambda_F \lambda_f \lambda_y (c_=(f\ y)c_0) y (Ff(c_s\ y))\Delta f_1 c_x \overset{\beta}{=} (c_=(f_1\ c_x)c_0)c_x(\Delta f_1(c_s\ c_x))$. От допускането можем да презапишем този израз като $(c_= c_0 c_0)c_x(\Delta f_1(c_s\ c_x)) \overset{\beta}{=} c_t c_x(\Delta f_1(c_s\ c_x)) \overset{\beta}{=} c_x$. По този начин довършваме случая.

Случай 2: $f_1 c_x \overset{\beta}{\neq} c_0 \iff f_2(x) \neq 0$ и $s_2(x)$ е дефинирано

В този случай трябва да покажем, че $s_1 c_x \overset{\beta}{=} s_1 c_{x+1}$. По-късно ще видим като доказваме коректността на функцията s_2, че $s_1 c_{x+1}$ ще бъде λ-определимо. Подобно на миналия случай започваме да правим β-редукции $s_1 c_x \equiv (Y\Gamma f_1)c_x \overset{\beta}{=} \dots \overset{\beta}{=} (c_=(f_1\ c_x)c_0)c_x(\Delta f_1(c_s\ c_x))$. От предположението знаем, че този израз е β-еквивалентен на $c_f c_x(\Delta f_1(c_s\ c_x)) \overset{\beta}{=} \Delta f_1(c_s\ c_x) \equiv ((\lambda_x \Gamma(xx))(\lambda_x \Gamma(xx)))f_1(c_s\ c_x) \overset{\beta}{=} Y\Gamma f_1(c_s\ c_x) \overset{\beta}{=} Y\Gamma f_1 c_{x+1} \equiv s_1 c_{x+1}$. Така доказахме случая.

Случай 3: $s_2(x)$ не е дефинирано

Тук трябва да покажем, че $s_1 c_x$ няма нормална форма, но за жалост не знам как :(.

Знаейки вече, че $s_1 c_x$ има същата семантика като $s_2(x)$, то остава смао да направим доказателство, че $s_2(0)$ наистина намира най-малката стойност x, за която $f_2(x) = 0$, ако изобщо съществува такава. Ще докажем това твърдение със силна индукция наобратно. По-конкретно ще докажем следното твърдение:

$$s_2(x) = \begin{cases} y & \exists y \in \mathbb{N}, y \geq x : f_2(y) = 0 \land \nexists z \in \mathbb{N}, x \leq z < y : f_2(z) = 0 \\ \text{недефинирана} & \nexists y \in \mathbb{N}, y \geq x : f_2(y) = 0 \end{cases}$$

Тоест, ще доказваме, че $s_2(x)$ намира най-близкото не по-малко число от x, което е корен на f_2.

База

Нека $y \in \mathbb{N}$, $f_2(y) = 0$ и y е минималното такова число. Нека $x = y$, тогава е ясно, че $s_2(x) = x = y$. Ако такова y не съществува, то от дефиницията на s_2 става ясно, че $s_2(x)$ е недефинирано $\forall x \in \mathbb{N}$, така че вече ще разглеждаме само случая, в който y съществува. Тогава също и термът $s_1 c_y$ ще е ламбда определим и β-еквивалентен на c_y.

Индуктивно предположение

Допускаме, че функцията s_2 се държи както е описано по-горе за всички стойности в интервала $[x+1, y]$, където y е минималният корен на функцията f_2. А пък, ако такова y не съществува, то $s_2(x)$ не е дефинирано. Също и че термът $s_1 c_a$, $a \in [x+1, y]$ е λ-определеим и има семантиката на $s_2(a)$.

Индуктивна стъпка

Ще докажем, че $s_2(x)$ има очакваното поведение използвайки ИП. От минималността на y знаем, че $f_2(x) \neq 0$. Тогава по определението за s_2 знаем, че $s_2(x) = s_2(x+1)$. От индуктивното предположение става ясно, че $s_2(x+1) = y$, което е очаквания резултат. Също и е ясно, че ако y не съществува, то $s_2(x)$ няма да е дефинирано. Също така, тъй като $s_1 c_x$ ще бъде β-еквивалентно на $s_1 c_{x+1}$ в този случай, то $s_1 c_x$ ще има семантиката на $s_2(x)$.

Така вече знаем, че s_2 върши това, което очакваме да върши и можем да заключим, че $s_1 c_0$ наистина намира най малкия корен на функцията f_1, ако той съществува, а иначе няма нормална форма.

Задача 2.29

- [] - $\lambda_f \lambda_e e$.
- $[x_1]$ - $\lambda_f \lambda_e ((fe)x_1)$
- $[x_1, x_2]$ - $\lambda_f \lambda_e (f((fe)x_1))x_2$
- $[x_1, x_2, ..., x_{n+1}]$ - $\lambda_f \lambda_e (f((l_{[x_1,x_2,....,x_n]} f)e))x_{n+1}$, където $l_{[x_1,x_2,....,x_n]}$ е списъка $[x_1, x_2, ..., x_n]$.

Желаните функции са реализирани във файла lists.scm. Append се казва pushBack. Функцията member е написана да сравнява само числа, за да се тества по лесно.

Задача 2.38

Ще направим индукция по дефиницията на M.

Случай 1 - $M \equiv y$, y е променлива

Случай 1.1 - $y \equiv x$

Тогава $M[x \longmapsto N] \equiv N$ и $M[x \longmapsto N'] \equiv N'$. От условието е ясно, че $(M[x \longmapsto N], M[x \longmapsto N']) \in D^{\lambda,R,T}$ заради рефлексивността.

Случай 1.2 - $y \not\equiv x$

Тогава $M[x \longmapsto N] \equiv y$ и $M[x \longmapsto N'] \equiv y$. От условието е ясно, че $(M[x \longmapsto N], M[x \longmapsto N']) \in D^{\lambda,R,T}$ заради рефлексивността.

Случай 2 - $M \equiv M_1, M_2$

От ИП имаме, че $(M_1[x \longmapsto N], M_1[x \longmapsto N']) \in D^{\lambda,R,T}$ и $(M_2[x \longmapsto N], M_2[x \longmapsto N']) \in D^{\lambda,R,T}$. Тъй като $M[x \longmapsto N] \equiv (M_1[x \longmapsto N])(M_2[x \longmapsto N])$ и $M[x \longmapsto N'] \equiv (M_1[x \longmapsto N'])(M_2[x \longmapsto N'])$, то от ламбда затварянето и транзитивността следва, че $(M[x \longmapsto N], M[x \longmapsto N']) \in D^{\lambda,R,T}$.

Случай 3 - $M \equiv \lambda_y P$

Случай 3.1 - $y \equiv x$

$(\lambda_x P)[x \longmapsto N] \equiv \lambda_x P$ и $(\lambda_x P)[x \longmapsto N'] \equiv \lambda_x P$. От рефликсвоност е вярно, че $(M[x \longmapsto N], M[x \longmapsto N']) \in D^{\lambda,R,T}$.

Случай 3.2 - $y \not\equiv x$

$(\lambda_y P)[x \longmapsto N] \equiv \lambda_y(P[x \longmapsto N])$ и $(\lambda_y P)[x \longmapsto N'] \equiv \lambda_y(P[x \longmapsto N'])$. От ИП имаме, че $(P[x \longmapsto N], P[x \longmapsto N']) \in D^{\lambda,R,T}$. Так от ламбда затварянето имаме, че $(M[x \longmapsto N], M[x \longmapsto N']) \in D^{\lambda,R,T}$.