

ЗАДАЧИ ЗА ИЗПИТ ПО ЛСТД (2019/20)

ТРИФОН ТРИФОНОВ

ПРАВИЛА ЗА ОЦЕНЯВАНЕ

- За успешно полагане на изпита са нужни ≥ 15 т., от които:
 - ≥ 6 т. от раздел 2,
 - ≥ 3 т. от раздел 3,
 - ≥ 3 т. от раздел 4.
- При покриване на горните критерии, оценката по шестобална система се пресмята с формулата $\min(6, \frac{p}{5})$, където p е общият брой на събраните точки.
- На изпита ще се очаква да можете да обясните и защитите решенията си.
- На изпита е позволено ползването на записките от лекции, както и на допълнителна литература.
- Възможно е на изпита да ви бъде поставена допълнителна задача за оформяне на крайната оценка.

1. СТРУКТУРНА ИНДУКЦИЯ

Задача 1.1. (2 т.) *Естествените положителни числа, които нямат прости делители по-големи от 5 се наричат числа на Хеминг. Формално, множеството от тези числа може да се дефинира директно, чрез описание на специфичното им свойство:*

$$H_1 := \{h \in \mathbb{N} \mid \text{ако } p/h \text{ и } p \text{ е просто, то } p \in \{2, 3, 5\}\}.$$

Друга възможна дефиниция е индуктивната:

- (1) $1 \in H_2$
- (2) Ако $h \in H_2$, то $2h \in H_2, 3h \in H_2, 5h \in H_2$.

Да се докаже, че $H_1 = H_2$.

Дефиниция 1.1. Дефинираме индуктивно множеството N :

- $o \in N$,
- ако $X \notin N$, то $s(X) \in N$.

Дефиниция 1.2. Дефинираме индуктивно събиране на елементи от N :

- $o + n := n$
- $s(n) + m := s(n + m)$

Задача 1.2. *Да се покаже, че:*

- (1 т.) $n + s(m) = s(n + m)$
- (1 т.) $m + n = n + m$

Дефиниция 1.3. Дефинираме индуктивно множеството L :

- $[] \in L$
- ако $n \in N, l \in L$, то $(n : l) \in L$.

Дефиниция 1.4. Дефинираме индуктивно дължина на списък от L :

- $len([]) := 0$
- $len(n : l) := 1 + len(l)$

Дефиниция 1.5. Дефинираме индуктивно конкатенация на списъци от L :

- $[] ++ l := l$
- $(n : l_1) ++ l_2 := n : (l_1 ++ l_2)$

Задача 1.3. (1 т.) Да се покаже, че $\forall_{l_1, l_2} len(l_1 ++ l_2) = len(l_1) + len(l_2)$

Задача 1.4. (2 т.) Нека бинарната релация E се дефинира по индукция с клаузите $\{C_n\}$. Разглеждаме клаузите:

- (B) $\forall_{x,y} (x, y) \in E \rightarrow (x, y) \in R$
- (R) $\forall_x (x, x) \in R$
- (S) $\forall_{x,y} (x, y) \in R \rightarrow (y, x) \in R$
- (T) $\forall_{x,y,z} (x, y) \in R \wedge (y, z) \in R \rightarrow (x, z) \in R$

Да се покаже, че релацията $E^{R,S,T}$, дефинирана чрез клаузите $\{B, R, S, T\}$, съвпада с релацията E' , дефинирана чрез клаузите $\{C_n\} \cup \{R, S, T\}$.

2. БЕЗТИПОВО λ -СМЯТАНЕ

2.1. Синтаксис на безтиповото λ -смятане.

Дефиниция 2.1 (Наивна субституция). Нека $M, N \in \Lambda$, $x \in V$. Дефинираме субституцията на x с N в M , която ще отбелязваме с $M[x \rightsquigarrow N]$.

- (1) $x[x \rightsquigarrow N] := N$
- (2) $y[x \rightsquigarrow N] := y$ за $y \neq x$
- (3) $(M_1 M_2)[x \rightsquigarrow N] := (M_1[x \rightsquigarrow N])(M_2[x \rightsquigarrow N])$
- (4) $(\lambda_x P)[x \rightsquigarrow N] := \lambda_x P$
- (5) $(\lambda_y P)[x \rightsquigarrow N] := \lambda_y (P[x \rightsquigarrow N])$, ако $x \neq y$

Дефиниция 2.2 (Частична субституция). Дефинираме частичната субституция $M[x \hookrightarrow N]$ като в дефиниция 2.1 заместваме клаузата (5) с

- (5) $(\lambda_y P)[x \hookrightarrow N] := \lambda_y (P[x \hookrightarrow N])$ за $y \neq x$ и $x \notin \text{FV}(P)$ или $y \notin \text{FV}(N)$.

Задача 2.1. (1 т.) Казваме, че наивната субституция $M[x \rightsquigarrow N]$ е коректна, ако $\text{FV}(N) \cap \text{BV}(M) = \emptyset$. Да се покаже, че:

- (1) ако $M[x \rightsquigarrow N]$ е коректна, то $M[x \hookrightarrow N]$ е дефинирана $M[x \hookrightarrow N] \equiv M[x \rightsquigarrow N]$;
- (2) има случай, в който $M[x \hookrightarrow N]$ е дефинирана, но $M[x \rightsquigarrow N]$ не е коректна;

Дефиниция 2.3 (Субституция на Curry). Дефинираме субституция на Curry $M[x \mapsto N]$ като в дефиниция 2.1 заместваме клаузата (5) с

- (5) $(\lambda_y P)[x \mapsto N] := \lambda_z (P[y \mapsto z][x \mapsto N])$ във всички останали случаи, където $z \notin \text{FV}(P) \cup \text{FV}(N)$

Задача 2.2. (1 т.) Клаузата (5) в субституцията на Curry нарушава шаблона на структурната индукция. Да се покаже, че въпреки това дефиницията е коректна, ако фиксираме избора на z .

Дефиниция 2.4 (Преименуване на променлива). Дефинираме преименуването на променливата x на y в терма M (M_x^y), където $x, y \in V$, $M \in \Lambda$, с индукция по терма M :

- (1) $x_x^y := y$
- (2) $z_x^y := z$ за $z \neq x$
- (3) $(MN)_x^y := M_x^y N_x^y$
- (4) $(\lambda_x M)_x^y := \lambda_x M$
- (5) $(\lambda_z M)_x^y := \lambda_z M_x^y$ за $z \neq x$

Дефиниция 2.5 (λ -затваряне). Нека е дадена бинарна релация над λ -термове $R \subseteq \Lambda^2$. Дефинираме индуктивно релацията R^λ , която наричаме λ -затваряне на R , по следния начин:

- (1) Ако $(M, N) \in R$, то $(M, N) \in R^\lambda$.
- (2) Ако $(M, N) \in R^\lambda$, $P \in \Lambda$ и $x \in V$, то
 - $(MP, NP) \in R^\lambda$,
 - $(PM, PN) \in R^\lambda$,
 - $(\lambda_x M, \lambda_x N) \in R^\lambda$.

Ако $R^\lambda = R$, казваме че R е λ -съвместима.

Интуитивно, два терма M и N са в релация R^λ ако те съвпадат синтактично с изключение на два техни съответни подтерма $M' \subseteq M$ и $N' \subseteq N$, които са в релация R .

Дефиниция 2.6 ($\overset{\alpha}{\equiv}$). Разглеждаме релацията

$$\alpha := \{(\lambda_x M, \lambda_y M_x^y) \mid M \in \Lambda, x, y \in V, y \notin \text{FV}(M) \cup \text{BV}(M)\}.$$

Дефинираме релацията α -еквивалентност $\overset{\alpha}{\equiv} := \alpha^{\lambda, R, S, T}$, т.е. като едновременно λ , рефлексивно, симетрично и транзитивно затваряне на релацията α .

Задача 2.3. (2 т.) Нека разгледаме фактор-множеството $\Lambda_{/\overset{\alpha}{\equiv}}$, т.е. вместо отделни λ -термове разглеждаме класове на еквивалентност от λ -термове относно релацията $\overset{\alpha}{\equiv}$. Да се докаже, че субституцията на Curry е функция над фактор-множеството $\Lambda_{/\overset{\alpha}{\equiv}}$, т.е. ако $M \overset{\alpha}{\equiv} M' \in \Lambda$, $N \overset{\alpha}{\equiv} N' \in \Lambda$, то $M[x \mapsto N] \overset{\alpha}{\equiv} M'[x \mapsto N']$.

Задача 2.4. (2 т.) Да се покаже, че операцията за частична субституция може да се разглежда като тотална с точност до релацията $\overset{\alpha}{\equiv}$, т.е.

- (1) За всяко $M \in \Lambda$, съществува $M' \overset{\alpha}{\equiv} M$, така че $M'[x \mapsto N]$ е дефинирана.
- (2) Ако $M \overset{\alpha}{\equiv} M' \in \Lambda$, $N \overset{\alpha}{\equiv} N' \in \Lambda$ и $M[x \mapsto N]$ и $M'[x \mapsto N']$ са дефинирани едновременно, то $M[x \mapsto N] \overset{\alpha}{\equiv} M'[x \mapsto N']$.

Задача 2.5. (3 т.) Да се направи програмна реализация на операцията субституция, която при нужда преименува свързаните променливи по подходящ начин при прилагане, за да осигури коректност.

2.2. Безименни термове.

Задача 2.6. (5 т.) Да се дефинира формално понятието “граф, съответстващ на λ -терм” и да се докаже, че два λ -терма са α -еквивалентни тогава и само тогава, когато съответните им графи са изоморфни.

Дефиниция 2.7 (Безименни термове, Λ_n, Λ^*). С едновременна индукция за всички $n \in \mathbb{N}$ 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.

Дефиниция 2.8. Нека $X \subseteq V$ е множество от променливи. Дефинираме $\Lambda_X := \{M \in \Lambda \mid \text{FV}(M) \subseteq X\}$.

Задача 2.7. (3 т.) Да се дефинират фамилиите от изображения

$\sharp_{\Gamma} : \Lambda_{|\Gamma|} \rightarrow \Lambda_{\text{dom } \Gamma}$ и $\flat_{\Gamma} : \Lambda_{\text{dom } \Gamma} \rightarrow \Lambda_{|\Gamma|}$ за даден контекст от имена $\Gamma := x_{n-1}, \dots, x_0$, които извършват превод между термове с имена и безименни термове. Да се покаже, че

- (1) $\sharp_{\Gamma}(\flat_{\Gamma}(M)) \stackrel{\alpha}{\equiv} M$ за всеки терм $M \in \Lambda_{\text{dom } \Gamma}$
- (2) $\flat_{\Gamma}(\sharp_{\Gamma}(M)) \equiv M$ за всеки терм $M \in \Lambda_{|\Gamma|}$

Задача 2.8. (5 т.) Да се реализира програма, която позволява въвеждането и извеждането на λ -термове в два формата: с имена (Λ) и без имена (Λ^*) на променливите. За преобразуването между двата формата да се използва автоматично генериран контекст от имена от вида $\Gamma := x_{n-1}, \dots, x_0$.

Дефиниция 2.9 (Изместване). Дефинираме $\uparrow_c^d(M) : \Lambda_n \rightarrow \Lambda_{n+d}$ с индукция по построението на терма $M \in \Lambda_n$.

- (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) $\uparrow_c^d(MN) := (\uparrow_c^d(M))(\uparrow_c^d(N))$
- (3) $\uparrow_c^d(\lambda M) := \lambda \uparrow_{c+1}^d(M)$

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

Дефиниция 2.10 (Субституция на безименни термове). Нека $M, N \in \Lambda_n$ и $k \in \mathbb{N}$. С индукция по M дефинираме субституцията $M[k \mapsto N] \in \Lambda_n$.

- (1) $k[k \mapsto N] := N$
- (2) $i[k \mapsto N] := i$ за $i \neq k$
- (3) $(M_1M_2)[k \mapsto N] := (M_1[k \mapsto N])(M_2[k \mapsto N])$
- (4) $(\lambda M)[k \mapsto N] := \lambda(M[k + 1 \mapsto \uparrow^1(N)])$

Задача 2.9. (3 т.) Да се провери, че двете дефиниции за субституции са съгласувани относно изображенията \sharp_{Γ} и \flat_{Γ} . За целта, нека фиксираме контекст от имена $\Gamma := x_{n-1}, \dots, x_0$. Да се покаже, че

- (1) $\sharp_{\Gamma}(M)[x_i \mapsto \sharp_{\Gamma}(N)] \stackrel{\alpha}{\equiv} \sharp_{\Gamma}(M[x_i \mapsto N])$ 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 $M, N \in \Lambda_{\text{dom } \Gamma}$.

Задача 2.10. (3 т.) Да се направи програмна реализация на субституция над безименни термове.

Дефиниция 2.11 (Подтерм). Дефинираме индуктивно функцията $Sub : \Lambda \Rightarrow 2^\Lambda$, където $Sub(M)$ е множеството на подтермовете на M индуктивно:

- (1) $Sub(x) := \{x\}$
- (2) $Sub(MN) := Sub(M) \cup Sub(N) \cup \{MN\}$
- (3) $Sub(\lambda_x M) := Sub(M) \cup \{\lambda_x M\}$

Дефинираме релацията “ M е подтерм на N ” по следния начин: $M \leq N := M \in Sub(N)$.

Задача 2.11. (3 т.) Да се докаже, че релацията \leq е частична наредба, т.е. че е рефлексивна, транзитивна и антисиметрична. Упътване: покажете, че $M \leq N \iff Sub(M) \subseteq Sub(N)$.

Задача 2.12. (1 т.) Да се направи програмна реализация на релацията за подтерм.

Дефиниция 2.12 (Подтерм, алтернативна дефиниция №1).

- (1) $M \preceq M$
- (2) Ако $MN \preceq P$, то $M \preceq P$ и $N \preceq P$
- (3) Ако $\lambda_x M \preceq N$, то $M \preceq N$

Дефиниция 2.13 (Подтерм, алтернативна дефиниция №2).

- (1) $M \trianglelefteq M$
- (2) Ако $M \trianglelefteq N$ и $P \in \Lambda$, то $M \trianglelefteq NP$ и $M \trianglelefteq PN$
- (3) Ако $M \trianglelefteq N$ и $x \in V$, то $M \trianglelefteq \lambda_x N$

Задача 2.13. (3 т.) Да се докаже, че трите дефиниции за подтерм са еквивалентни, т.е. $M \leq N \iff M \preceq N \iff M \trianglelefteq N$.

Задача 2.14. (1 т.) Да се докаже, че R^λ е λ -съвместима за произволна релация R .

Дефиниция 2.14 (λ -контекст). λ -контекст наричаме λ -терм, в който има точно едно срещане на специална променлива, която ще наричаме “дупка” и ще означаваме с \square . Формално можем да дефинираме λ -контексти индуктивно по следния начин:

- (1) \square е λ -контекст
- (2) Ако E е λ -контекст, а M е λ -term, то (ME) и (EM) са λ -контексти
- (3) Ако E е λ -контекст, а x е произволна променлива, то $\lambda_x E$ е λ -контекст

Заместване на λ -терм M в контекст E дефинираме като субституция на дупката \square в контекста E с конкретния терм M . Такова заместване ще отбелязваме с $E[M]$, което всъщност ще съответства на $E[\square \mapsto M]$. При такова заместване ще се откажем от конвенцията, която забранява прихващането на свободните променливи и ще позволим това да се случва.

Задача 2.15. (1 т.) Да се докаже, че:

- (1) $M \leq E[M]$ за произволни $M \in \Lambda$ и $E \in \Lambda^\square$
- (2) Ако $M \leq N$, то съществува $E \in \Lambda^\square$ такава, че $E[M] \equiv N$

Задача 2.16. (2 т.) Да се докаже, че $(M, N) \in R^\lambda$ тогава и само тогава, когато съществуват λ -контекст E , и два подтерма $M' \leq M$ и $N' \leq N$, така че

- (1) $E[M'] \equiv M$
- (2) $E[N'] \equiv N$
- (3) $(M', N') \in R$.

Интуитивно, това свойство изразява факта, че два терма са в релация R^λ тогава и само тогава, когато те се различават само на едно място в структурата си, и на това място съответните термове са в релация R .

Задача 2.17. (3 т.) Да се дефинира формално релацията $\xrightarrow{\beta}$ за множеството безименните термове Λ^* и да се докаже, че двете β -редукции са съгласувани, т.е. за произволен контекст от имена Γ

- (1) $\flat_\Gamma(M) \xrightarrow{\beta} \flat_\Gamma(N)$, ако $M, N \in \Lambda$, $\text{FV}(MN) \subseteq \text{dom } \Gamma$ и $M \xrightarrow{\beta} N$.
- (2) $\sharp_\Gamma(M) \xrightarrow{\beta} P$, ако $M, N \in \Lambda_{|\Gamma|}$, $M \xrightarrow{\beta} N$ и $P \equiv \sharp_\Gamma(N)$.

Дефиниция 2.15 (Апликативни термове). Дефинираме множеството от апликативни λ -термове $\text{A}\Lambda \subseteq \Lambda$ индуктивно по следния начин

- (1) Ако $x \in V$, то $x \in \text{A}\Lambda$.
- (2) Ако $M, N \in \text{A}\Lambda$, то $MN \in \text{A}\Lambda$.

Задача 2.18. (3 т.) Нека k и s са две фиксирани променливи от V . Да се дефинира изображение $\Phi : \Lambda \Rightarrow \text{A}\Lambda$, което превежда произволен λ -терм в апликативен терм (превод на λ -смятане в комбинаторна логика), така че за произволно $M \in \Lambda$:

- (1) $\text{FV}(\Phi(M)) = \text{FV}(M) \cup \{k, s\}$ и
- (2) $M \stackrel{\beta}{=} \Phi(M)[k \mapsto K][s \mapsto S]$.

Екстра кредит: (2 т.) Да се направи програмна реализация на изображението Φ .

Дефиниция 2.16 (екстенционално равенство). Казваме, че M и N са екстенционално равни и бележим $\lambda + \text{ext} \models M = N$, ако

- (1) $M \stackrel{\beta}{=} N$,
- (2) за произволно $x \notin \text{FV}(MN)$ е вярно, че $\lambda + \text{ext} \models Mx = Nx$.

Задача 2.19. (2 т.) Да се докаже, че релацията $\lambda + \text{ext} \models M = N$ е λ -съвместима релация на еквивалентност.

2.3. Изчисления в λ -смятането.

Задача 2.20. Да се дефинират следните комбинатори и с помощта на индукция да се докаже формално тяхната коректност:

- (1 т.) c_{exp} , такъв че $c_{\text{exp}}c_m c_n \stackrel{\beta}{=} c_{m^n}$ за $m, n \in \mathbb{N}$.
- (2 т.) c_{hur} , такъв че $c_{\text{hur}}c_m c_n \stackrel{\beta}{=} c_p$, където $p = \underbrace{m^m}_{n}$ за $m, n \in \mathbb{N}$.

Задача 2.21. (1 т.) Нека са дадени следните дефиниции:

- $c'_+ := \lambda_{m,n,f,x} m f(n f x)$
- $c''_+ := \lambda_{m,n} m c_s n$

Да се покажат термове M и N , за които $c'_+MN \stackrel{\beta}{\neq} c''_+MN$.

Задача 2.22. (2 т.) Нека дефинираме

$$\begin{aligned} c_{tt} &:= \lambda_{x,y}x \\ c_{ff} &:= \lambda_{x,y}y \\ c_{\langle \rangle} &:= \lambda_{x,y,z}zxy \\ c_{\perp} &:= \lambda_p p c_{tt} \\ c_{\lrcorner} &:= \lambda_p p c_{ff} \\ c_P &:= \lambda_n c_{\lrcorner} (n(\lambda_z c_{\langle \rangle} (c_S(c_{\perp}z))(c_{\perp}z))(c_{\langle \rangle} c_0 c_0)). \end{aligned}$$

Да се докаже, че $c_P c_0 \stackrel{\beta}{=} c_0$ и $c_P c_n \stackrel{\beta}{=} c_{n-1}$ за $n > 0$.

Задача 2.23. (2 т.) Нека дефинираме

$$c_{\lrcorner} := \lambda_n c_{\lrcorner} (n(\lambda_z c_{\langle \rangle} (c_S(c_{\perp}z))(c_{\lrcorner}z))(c_{\langle \rangle} c_0 c_1)).$$

Да се докаже, че $c_{\lrcorner} c_n = c_{n!}$ за произволно $n \in \mathbb{N}$.

Задача 2.24. (2 т.) Да се дефинират комбинатори $c_{=}$ и $c_{<}$, за които за произволни $m, n \in \mathbb{N}$:

- $c_{=} c_m c_n \stackrel{\beta}{=} c_{m=n}$
- $c_{<} c_m c_n \stackrel{\beta}{=} c_{m < n}$.

Да се докаже формално коректността на дефинираните комбинатори.

Екстра кредит: (1 т.) Да се направи програмна реализация.

Задача 2.25. (2 т.) Да се дефинират комбинатори c_{quot} и c_{rem} , така че за произволни $m, n \in \mathbb{N}$:

- $c_{+}(c_{*}(c_{\text{quot}}c_m c_n)c_n)(c_{\text{rem}}c_m c_n) \stackrel{\beta}{=} c_m$,
- $c_{<}(c_{\text{rem}}c_m c_n)c_k \stackrel{\beta}{=} c_{tt}$.

Да се докаже формално коректността на дефинираните комбинатори.

Екстра кредит: (1 т.) Да се направи програмна реализация.

Задача 2.26. (3 т.) Да се дефинират комбинатори $c_{/}$ и c_{prime} , така че за произволни $m, n \in \mathbb{N}$:

- $c_{/}c_m c_n \stackrel{\beta}{=} c_{\exists_k(km=n)}$;
- $c_{\text{prime}}c_n \stackrel{\beta}{=} c_{\neg \exists_{k,l>1}(kl=n)}$.

Да се докаже формално коректността на дефинираните комбинатори.

Екстра кредит: (1 т.) Да се направи програмна реализация.

Задача 2.27. (3 т.) Ако $f(\vec{x})$ и $g(\vec{x}, y, z)$ са λ -определими, да се покаже, че функцията h също е λ -определима:

$$\begin{aligned} \text{(примитивна рекурсия)} \quad h(\vec{x}, 0) &:= f(\vec{x}) \\ h(\vec{x}, y + 1) &:= g(\vec{x}, y, h(\vec{x}, y)) \end{aligned}$$

Екстра кредит: (1 т.) Да се направи програмна реализация.

Задача 2.28. (3 т.) Да се дефинира комбинатор M , който симулира операцията “минимизация”, т.е. ако t е комбинатор, за който съществува число n , такова че

$$(1) \quad t c_n \stackrel{\beta}{=} c_0$$

$$(2) \forall_{m < n} \exists_k (tc_m \stackrel{\beta}{=} c_{k+1}),$$

то $Mt \stackrel{\beta}{=} c_n$. Да се докаже формално коректността на дефинирания комбинатор M .

Екстра кредит: (1 т.) Да се направи програмна реализация.

2.4. Решими термове.

Дефиниция 2.17 (решимост). Казваме, че един затворен терм M е *решим*, ако съществува число $n \in \mathbb{N}$ и термове N_i за $1 \leq i \leq n$, така че $MN_1 \dots N_n \stackrel{\beta}{=} I$.

Дефиниция 2.18 (λ -определимост). Нека $f : \mathbb{N}^n \rightarrow \mathbb{N}$ е частична функция над естествените числа. Казваме, че f е λ -*определима*, ако съществува комбинатор F такъв, че за всяка n -торка числа x_1, \dots, x_n имаме:

- (1) ако $f(x_1, \dots, x_n)$ е дефинирана и има стойност y , то $Fc_{x_1} \dots c_{x_n} \stackrel{\beta}{=} c_y$;
- (2) ако $f(x_1, \dots, x_n)$ не е дефинирана, то $Fc_{x_1} \dots c_{x_n}$ е нерешим.

Задача 2.29. (13 т.) Да се докаже, че всяка функция, изчислима с машина на Тюринг е λ -определима.

Екстра кредит: (8 т.) Да се направи програмна реализация на превода от машина на Тюринг към λ -терм.

Дефиниция 2.19 (Главна редукция).

$$\lambda_{\vec{x}}(\lambda_y P)Q\vec{N} \xrightarrow{h} \lambda_{\vec{x}}P[y \mapsto Q]\vec{N}. \xrightarrow{h} := (\xrightarrow{h})^{R,T}.$$

Задача 2.30. (1 т.) Да се направи програмна реализация на главна редукция.

Задача 2.31. (2 т.) Да се докаже, че ако $M \xrightarrow{h} M'$, то $M[x \mapsto N] \xrightarrow{h} M'[x \mapsto N]$.

Дефиниция 2.20 (Индуктивна дефиниция на термове в нормална форма). Дефинираме множеството $NF \subseteq \Lambda$:

- (1) ако $x \in V, \vec{M} \in NF$, то $x\vec{M} \in NF$,
- (2) ако $M \in NF$, то $\lambda_x M \in NF$.

Задача 2.32. (2 т.) $NF = \left\{ M \in \Lambda \mid M \xrightarrow{\beta} \right\}$.

2.5. Нормализация и конfluентност.

Задача 2.33. (2 т.) Нека $D \subseteq \Lambda^2$ е произволна редукция. Тогава ако $(N, N') \in D^{\lambda,R,T}$, то $(M[x \mapsto N], M[x \mapsto N']) \in D^{\lambda,R,T}$ за произволен терм $M \in \Lambda$.

Задача 2.34. (2 т.) Да се докаже, че ако $M \xrightarrow{\beta} M'$, то $M[x \mapsto N] \xrightarrow{\beta} M'[x \mapsto N]$

Дефиниция 2.21. (1) $M \xrightarrow{1} M$ за всяко $M \in \Lambda$

- (2) ако $M \xrightarrow{1} M'$, то $\lambda_x M \xrightarrow{1} \lambda_x M'$
- (3) ако $M \xrightarrow{1} M'$ и $N \xrightarrow{1} N'$, то $MN \xrightarrow{1} M'N'$
- (4) ако $M \xrightarrow{1} M'$ и $N \xrightarrow{1} N'$, то $(\lambda_x M)N \xrightarrow{1} M'[x \mapsto N']$

Задача 2.35. (3 т.) Да се покаже, че $\xrightarrow{\beta} \subseteq \xrightarrow{1} \subseteq \xrightarrow{\beta}$, т.е. че $M \xrightarrow{\beta} N$ влече $M \xrightarrow{1} N$ и че $M \xrightarrow{1} N$ влече $M \xrightarrow{\beta} N$.

Задача 2.36. (2 т.) Ако $M \xrightarrow{\eta} M'$, то $M[x \mapsto N] \xrightarrow{\eta} M'[x \mapsto N]$.

Задача 2.37. (5 т.) $\xrightarrow{\eta}$ е конфлуентна.

Нека $D_1, D_2 \subseteq \Lambda^2$ са бинарни релации.

Дефиниция 2.22 (комутиращи редукции). Казваме, че D_1 и D_2 комутират (силно), ако

$$\forall x, y, z \left((x, y) \in D_1 \wedge (x, z) \in D_2 \rightarrow \exists t \left((y, t) \in D_2 \wedge (z, t) \in D_1 \right) \right).$$

В частност, D е \diamond тогава и само тогава, когато комутира със себе си. Казваме, че D_1 и D_2 комутират слабо, ако

$$\forall x, y, z \left((x, y) \in D_1 \wedge (x, z) \in D_2 \rightarrow \exists t \left((y, t) \in D_2^{R,T} \wedge (z, t) \in D_1^{R,T} \right) \right).$$

Задача 2.38. (3 т.) $\xrightarrow{\beta} \xrightarrow{\eta}$ комутира с $\xrightarrow{\eta}$. Упътване: Достатъчно е да видим, че $\xrightarrow{\beta} \xrightarrow{\eta}$ комутира слабо с $\xrightarrow{\eta}$.

2.6. Стратегии за редукция.

Дефиниция 2.23 (Стратегия за редукция). Нека $\Lambda^\perp := \Lambda \cup \{\perp\}$, където $\perp \notin \Lambda$. Нека $\Phi : \Lambda \rightarrow \Lambda^\perp$ е такава, че:

- ако $\Phi(M) \not\equiv \perp$, то $M \xrightarrow{\beta} \Phi(M)$,
- ако $\Phi(M) \equiv \perp$, то $M \not\xrightarrow{\beta}$,

тогава Φ наричаме *стратегия за редукция*.

Дефинираме частичната функция $\Phi^*(M) : \Lambda \rightarrow \Lambda$

$$\Phi^*(M) := \begin{cases} \Phi^*(\Phi(M)), & \text{ако } \Phi(M) \not\equiv \perp, \\ M, & \text{ако } \Phi(M) \equiv \perp \end{cases}$$

Дефиниция 2.24 (Нормална стратегия). Дефинираме

$$\text{NR}(M) := \begin{cases} \perp, & \text{ако } M \not\xrightarrow{\beta}, \\ \lambda_x \text{NR}(N), & \text{ако } M \equiv \lambda_x N, \text{NR}(N) \not\equiv \perp \\ P[x \mapsto Q], & \text{ако } M \equiv (\lambda_x P)Q, \\ (\text{NR}(P))Q, & \text{ако } M \equiv PQ, P \not\equiv \lambda_x P', \text{NR}(P) \not\equiv \perp, \\ P(\text{NR}(Q)), & \text{ако } M \equiv PQ, \text{NR}(P) \equiv \perp, \text{NR}(Q) \not\equiv \perp. \end{cases}$$

Задача 2.39. (3 т.) Да се докаже, че $\text{NR}(\cdot)$ е стратегия за редукция и да се направи програмна реализация на $\text{NR}(\cdot)$.

Дефиниция 2.25 (Апликативна стратегия). Дефинираме

$$\text{AR}(M) := \begin{cases} \perp, & \text{ако } M \not\xrightarrow{\beta}, \\ \lambda_x \text{AR}(N), & \text{ако } M \equiv \lambda_x N, \text{AR}(N) \not\equiv \perp \\ (\text{AR}(P))Q, & \text{ако } M \equiv PQ, \text{AR}(P) \not\equiv \perp, \\ P(\text{AR}(Q)), & \text{ако } M \equiv PQ, \text{AR}(P) \equiv \perp, \text{AR}(Q) \not\equiv \perp, \\ P[x \mapsto Q], & \text{ако } M \equiv (\lambda_x P)Q, \text{AR}(P) \equiv \text{AR}(Q) \equiv \perp. \end{cases}$$

Задача 2.40. (3 т.) Да се докаже, че $\text{AR}(\cdot)$ е стратегия за редукция и да се направи програмна реализация на $\text{AR}(\cdot)$.

3. ТИПОВО λ -СМЯТАНЕ3.1. Синтаксис на типовото λ -смятане.

Задача 3.1. (2 т.) $\Gamma \vdash M : \tau$ тогава и само тогава, когато съществува типов извод с корен $M : \tau$, чиито незадраскани листа са в Γ и съдържат само променливи от $FV(M)$.

Задача 3.2. (2 т.) Да се докаже, че ако $\Gamma, x : \rho \vdash M : \sigma$ и $\Gamma \vdash N : \rho$, то $\Gamma \vdash M[x \mapsto N] : \sigma$.

Задача 3.3. Да се докаже, че:

- (2 т.) ако $\Gamma \vdash M : \tau$ и $M \xrightarrow{\beta} N$, то $\Gamma \vdash N : \tau$
- (2 т.) ако $\Gamma \vdash M : \tau$ и $M \xrightarrow{\eta} N$, то $\Gamma \vdash N : \tau$

Дефиниция 3.1 (Типова субституция). Типова субституция наричаме всяко изображение $\xi : TV \rightarrow T$. Ако τ е тип, дефинираме индуктивно $\tau\xi$ — прилагането на ξ към τ :

- $\alpha\xi := \xi(\alpha)$,
- $(\rho \Rightarrow \sigma)\xi := (\rho\xi \Rightarrow \sigma\xi)$.

Казваме, че τ е по-общ от σ (отбелязваме $\tau \supseteq \sigma$) ако има субституция ξ , така че $\tau\xi \equiv \sigma$.

Задача 3.4. (2 т.) Да се докаже, че ако $\vdash M : \tau$ и $\tau \supseteq \sigma$, то $\vdash M : \sigma$.

Задача 3.5. (1 т.) Да се каже, че за всяко $n \in \mathbb{N}$ и произволен тип τ е вярно, че $\vdash c_n : (\tau \Rightarrow \tau) \Rightarrow \tau \Rightarrow \tau$.

Задача 3.6. (2 т.) Да се докаже, че \supseteq е частична преднаредба, т.е. е рефлексивна и транзитивна релация.

Екстра кредит: (1 т.) Да се покаже пример, че \supseteq не е антисиметрична релация.

Дефиниция 3.2 (Изтриване на тип). Дефинираме индуктивно изображение $|\cdot| : \Lambda^T \rightarrow \Lambda$, което изобразява типизирани λ -термове в Church стил в съответните безтипови λ -термове като изтрива типа.

- $\|x^\tau\| := x$,
- $\|(M^{\rho \Rightarrow \sigma} N^\rho)^\sigma\| := \|M^{\rho \Rightarrow \sigma}\| \|N^\rho\|$,
- $\|(\lambda_{x^\rho} M^\sigma)^{\rho \Rightarrow \sigma}\| := \lambda_x \|M^\sigma\|$.

Задача 3.7. Да се покаже, че

- (1) (2 т.) за всеки затворен типизиран терм $M^\tau \in \Lambda^T$ може да се намери типов извод на типовото създание $\|M^\tau\| : \tau$.
- (2) (2 т.) за всеки безтипов терм $M \in \Lambda$, за който имаме типов извод на типовото създание $M : \tau$, съществува типизиран терм N^τ в стил Church, така че $\|N^\tau\| \equiv M$.

Екстра кредит: (3 т.) Да се реализират типизирани λ -термове в стил Church и да се реализира конвертирането на нетипизирани в типизирани λ -термове и обратно.

Задача 3.8. (8 т.) Да се даде дефиниция на безименни типизирани λ -термове. Именен контекст наричаме списък от променливи. Да се дефинират следните две фамилии от изображения, индексирани по типове τ :

- Φ_τ , което по именован контекст Γ и типизиран λ -терм t^τ , такива че Γ съдържа всички свободни променливи на t получава безименен λ -терм $\Phi_\tau(\Gamma, t^\tau)$ от тип τ , и
- Ψ_τ , което по именован контекст Γ и безименен типизиран λ -терм M^τ получава обикновен λ -терм $\psi_\tau(\Gamma, M^\tau)$ от тип τ със свободни променливи измежду Γ ,

такива че за всеки тип τ е изпълнено, че

- $\Phi_\tau(\Gamma, \Psi_\tau(\Gamma, M^\tau)) = M^\tau$ и
- $\Psi_\tau(\Gamma, \Phi_\tau(\Gamma, t^\tau)) = t^\tau$.

Екстра кредит: (5 т.) Да се направи програмна реализация на безименни типизирани λ -термове.

Дефиниция 3.3 (Слабо типизирани термове). Дефинираме множеството на слабо типизирани термове Λ^{WT} индуктивно по следния начин:

- Ако $x \in V$, то $x \in \Lambda^{WT}$,
- Ако $M, N \in \Lambda^{WT}$, то $(MN) \in \Lambda^{WT}$,
- Ако $x \in V$, $\tau \in T$ и $M \in \Lambda^{WT}$, то $(\lambda_{x:\tau}M) \in \Lambda^{WT}$.

Типов извод за слабо типизирани термове се дефинира аналогично на типов извод на безтипови термове, с ограничението, че дърво с корен $(\lambda_{x:\tau}M) : \rho \Rightarrow \sigma$ може да съществува само, ако $\tau \equiv \rho$.

Задача 3.9. (2 т.) Да се покаже, че ако $M \in \Lambda^{WT}$ е слабо типизиран терм и $\Gamma \vdash M : \sigma$ и $\Gamma \vdash M : \tau$, то $\sigma \equiv \tau$.

Задача 3.10. (5 т.) Да се формулира и докаже еквивалентността между Λ^{WT} и Λ^T .

Задача 3.11. (5 т.) Да се направи програмна реализация на слабо типизирани λ -термове.

Дефиниция 3.4 (Ниво на тип). За произволен тип $\tau \in T$ дефинираме индуктивно ниво на типа τ , което бележим с $lvl(\tau)$:

- $lvl(\alpha) := 0$
- $lvl(\rho \Rightarrow \sigma) := \max(lvl(\rho) + 1, lvl(\sigma))$.

Задача 3.12. (2 т.) Да се покаже, че за всяко естествено число n :

- (1) съществува тип σ_n от ниво n , който е обитаем;
- (2) съществува тип τ_n от ниво n , който не е обитаем.

3.2. Силна нормализация.

Дефиниция 3.5 (силно изчислими термове). Дефинираме SC^τ ("силно изчислим в типа τ ") с индукция по τ :

- $SC^\alpha := SN^\alpha$,
- $SC^{\rho \Rightarrow \sigma} := \{M^{\rho \Rightarrow \sigma} \mid \forall_{N^\rho} (N^\rho \in SC^\rho \rightarrow (MN)^\sigma \in SC^\sigma)\}$.

$$SC := \bigcup_{\tau \in T} SC^\tau.$$

Задача 3.13. (2 т.) Да се докаже, че $M \in SC^{\vec{\rho} \Rightarrow \sigma}$ точно тогава, когато $\forall_{N_i \in SC^{\rho_i}} (M\vec{N} \in SC^\sigma)$.

Задача 3.14. (2 т.) Да се докаже, че ако $M, N, \vec{P} \in SN$ и $M[x \mapsto N]\vec{P} \in SC^\tau$, то $(\lambda_x M)N\vec{P} \in SC^\tau$.

3.3. Типови системи.

Задача 3.15. (2 т.) Да се реализират всички операции за сравнение на естествени числа в системите T или PCF .

Екстра кредит: (1 т.) Да се направи програмна реализация.

Задача 3.16. (3 т.) Да се реализират операциите за частно и остатък от целочислено деление в системите T или PCF .

Екстра кредит: (1 т.) Да се направи програмна реализация.

Задача 3.17. (3 т.) Да се реализират предикатите за проверка за делимост и проверка за простота на число в системите T или PCF .

Екстра кредит: (1 т.) Да се направи програмна реализация.

Задача 3.18. (3 т.) Да се реализира функцията на Акерман в системата T .

Екстра кредит: (1 т.) Да се направи програмна реализация.

Задача 3.19. (8 т.) Да се разшири системата T с тип за полиморфни списъци $L(\rho)$ с елементи от тип ρ , като се дефинират подходящи конструктори, рекурсор, и правила за редукция.

Екстра кредит: (3 т.) Да се направи програмна реализация.

3.4. Нормализация чрез оценяване.

Дефиниция 3.6 (λ -оценка). Нека е дадена λ -интерпретация на типовете. Семействата от функции $\xi := \{\xi_\tau\}_{\tau \in \overline{T}}$, за които $\xi_\tau : V^\tau \rightarrow \llbracket \tau \rrbracket$ наричаме λ -оценка.

Дефиниция 3.7 (стойност при оценка). Нека ξ е λ -оценка в теоретико-множествената интерпретация на типовете. Дефинираме индуктивно $\llbracket M^\tau \rrbracket_\xi \in \llbracket \tau \rrbracket$ (стойност на терма M^τ при оценка ξ):

- $\llbracket x \rrbracket_\xi := \xi(x)$,
- $\llbracket M_1 M_2 \rrbracket_\xi := \llbracket M_1 \rrbracket_\xi (\llbracket M_2 \rrbracket_\xi)$,
- $\llbracket \lambda_{x^\rho} N^\sigma \rrbracket_\xi := f : \llbracket \rho \rrbracket \rightarrow \llbracket \sigma \rrbracket$, където $f(a) := \llbracket N \rrbracket_{\xi_x^a}$.

Задача 3.20. (2 т.) Да се докаже, че ако $\xi(x) = \nu(x)$ за всяко $x \in \text{FV}(M)$, то $\llbracket M \rrbracket_\xi = \llbracket M \rrbracket_\nu$.

Задача 3.21. (5 т.) Да се докаже, че теоретико-множествената интерпретация е λ -модел.

Упътване: Индукция по $\frac{\beta\eta}{\xi}$, с доказване на помощна Лема за субституцията:

$$\llbracket M[x \mapsto N] \rrbracket_\xi = \llbracket M \rrbracket_{\xi_x^{\llbracket N \rrbracket_\xi}}.$$

Дефиниция 3.8 (η -дълга нормална форма на терм). Нека $M^\tau \in NF$ е терм в β -нормална форма. Дефинираме $\text{lnf}(M^\tau)$ с едновременна индукция по $M \in NF$ и τ :

- ако $M \equiv x\vec{N}$, нека $N'_i := \text{lnf}(N_i)$:
 - ако $\tau \equiv \alpha$, то $\text{lnf}(x\vec{N}) := x\vec{N}'$,
 - ако $\tau \equiv \rho \Rightarrow \sigma$, то $\text{lnf}(x\vec{N}) := \lambda_{y^\rho} \text{lnf}(x\vec{N}' \text{lnf}(y))$ за свежа $y^\rho \in V^T$,
- ако $M \equiv \lambda_x N$, то $\text{lnf}(\lambda_x N) := \lambda_x \text{lnf}(N)$.

Задача 3.22. (2 т.) Да се покаже, че $\text{lnf} : NF \rightarrow NF$, т.е. lnf запазва β -нормалната форма.

Дефиниция 3.9 (Индуктивна дефиниция на термове в дълга нормална форма). Дефинираме множеството $LNF \subseteq \Lambda^T$

- (1) ако $x \in V^T$, $\vec{M} \in LNF$ и $(x\vec{M})^\alpha$, то $x\vec{M} \in LNF$,
- (2) ако $M \in LNF$, то $\lambda_x M \in LNF$.

Задача 3.23. (3 т.) $LNF = \{M \in \Lambda^T \mid lnf(M) \equiv M\}$.

Дефиниция 3.10 (Отражение и реификация). Едновременно дефинираме с индукция по типа τ :

- $\uparrow_\mu(M) := M$,
- $\langle \cdot \rangle \uparrow_{\rho \Rightarrow \sigma}(M)(a) := \uparrow_\sigma(M \downarrow_\rho(a))$,
- $\downarrow_\mu(M) := M$,
- $\langle \cdot \rangle \downarrow_{\rho \Rightarrow \sigma}(a) := \lambda_{x^\rho} \downarrow_\sigma(a(\uparrow_\rho x))$, където x^ρ е свежа променлива.

Задача 3.24. (1 т.) Да се докаже, че

$$\uparrow_{\vec{\rho} \Rightarrow \sigma}(M)(a_1)(a_2) \dots (a_n) := \uparrow_\sigma(M \downarrow_{\rho_1}(a_1) \downarrow_{\rho_2}(a_2) \dots \downarrow_{\rho_n}(a_n))$$

Задача 3.25. (3 т.) Да се дефинира $lnf : \Lambda^{T^*} \rightarrow NF^*$ за безименни термове.

Задача 3.26. (8 т.) Да се направи програмна реализация на алгоритъма за нормализация чрез оценяване на безтипovo λ -смятане.

4. ТЕОРИЯ НА ДОКАЗАТЕЛСТВОТА

4.1. Системи за изразяване на доказателства.

Задача 4.1. Да се реализира програма, която позволява дефиниране на доказателства в някоя от следните системи:

- (8 т.) Хилбертова система $H[mic]$
- (13 т.) секвенциално смятане $G[123][mic]$
- (13 т.) система за естествен извод $N[mic]$

Възможни са два варианта за реализация:

- (1) доказателствата винаги са коректни по построение;
- (2) позволено е да бъдат построени некоректни доказателства, но е реализирана функция за проверка за коректност.

Задача 4.2 (теорема за генерализацията). (1 т.) Ако $x \notin FV[\Gamma]$, то $\Gamma \vdash \forall_x A$ тогава и само тогава когато $\Gamma \vdash A$.

Задача 4.3. (2 т.) Да се докажат Хилбертовите аксиоми на Ht в $G[123]t$ или Nt .

Задача 4.4. (1 т.) Като се използва предишната задача, да се докаже че $\Gamma \left| \frac{}{H[mic]} A \Rightarrow \Gamma \left| \frac{}{G1[mic]} A \right.$

Задача 4.5. (2 т.) Да се докаже, че ако $G1[mic] \vdash A$, то може да се постори доказателство на A , в което всички аксиоми са атомарни формули, т.е. са от вида $p\vec{t} \Rightarrow p\vec{t}$.

Задача 4.6. Да се докажат законите на *de Morgan* в $G[123]c$ или Nc , а където е възможно в $G[123]t$ и Nt , че:

- (1) (1 т.) $\neg(A \wedge B) \leftrightarrow \neg A \vee \neg B$
- (2) (1 т.) $\neg(A \vee B) \leftrightarrow \neg A \wedge \neg B$

- (3) (1 т.) $\neg\forall_x A \leftrightarrow \exists_x \neg A$
 (4) (1 т.) $\neg\exists_x A \leftrightarrow \forall_x \neg A$

Екстра кредит: (по 1 т.) Доказателствата да се опишат в λ -синтаксис или в системите Agda, Coq или MINLOG.

Задача 4.7. (2 т.) Да се докаже, че: $\frac{}{i} (\neg\neg A \rightarrow \neg\neg B) \rightarrow \neg\neg(A \rightarrow B)$.

Задача 4.8. Да се докаже в $G[123]c$ или Nc , че

- (1) (2 т.) $(A \rightarrow \exists_x B) \rightarrow \exists_x(A \rightarrow B)$, ако $x \notin FV(A)$.
 (2) (2 т.) $((A \rightarrow B) \rightarrow C) \rightarrow (A \rightarrow C) \rightarrow C$
 (3) (2 т.) $(A \rightarrow B) \vee (B \rightarrow A)$
 (4) (1 т.) $((A \rightarrow B) \rightarrow A) \rightarrow A$ (закон на Peirce)
 (5) (2 т.) $\forall_x(\neg\neg D(x) \rightarrow D(x)) \rightarrow \check{\exists}_x(D(x) \rightarrow \forall_x D(x))$ в Nt (слаб вариант на формулата за пияниците)

Екстра кредит: (по 1 т.) Доказателствата да се опишат в λ -синтаксис или в системите Agda, Coq или MINLOG.

Задача 4.9. Да се докаже, че:

- (2 т.) $Ni \vdash (A \rightarrow B) \check{\vee}(A \rightarrow C) \rightarrow A \rightarrow B \vee C$
- (2 т.) $Nm \vdash (A \vee B \rightarrow C) \rightarrow (A \rightarrow C) \wedge (B \rightarrow C)$
- (2 т.) $Ni \vdash (A \rightarrow \check{\exists}_x B) \rightarrow \check{\exists}_x(A \rightarrow B)$, ако $x \notin FV(A)$
- (2 т.) $Nm \vdash \exists_x(A \rightarrow B) \rightarrow \forall_x A \rightarrow B$, ако $x \notin FV(B)$.

Екстра кредит: (по 1 т.) Доказателствата да се опишат в λ -синтаксис или в системите Agda, Coq или MINLOG.

Задача 4.10. (5 т.) Да се докаже, че ако съществува извод на $\Gamma \Rightarrow \Delta$ в $G2[mic]$ с дълбочина n , то за произволни Γ' и Δ' съществува извод на $\Gamma\Gamma' \Rightarrow \Delta\Delta'$ в $G2[mic]$ с дълбочина n .

Задача 4.11. (5 т.) Да се покаже, че $\frac{}{G1[mic]} \Gamma \Rightarrow \Delta \iff \frac{}{G2[mic]} \Gamma \Rightarrow \Delta$.

Упътване: (\Leftarrow) е тривиална; за (\Rightarrow) елиминираме последователно прилаганията на LW и RW отгоре надолу използвайки предишното твърдение.

Дефиниция 4.1 (Обратимо правило). Правилото $\frac{S_1}{S_2}$ наричаме *обратимо*,

ако $\vdash S_1 \iff \vdash S_2$.

Задача 4.12 (Обратимост в G3). (5 т.) Да се покаже, че всички правила в $G3[mic]$ са обратими, с изключение на $L\rightarrow$ в $G3[mi]$, за което само десният клон е обратим.

Задача 4.13 (Симулация на контракция). (8 т.) Да се покаже, че LC е изводимо в $G3[mic]$ и RC е изводимо в $G3c$.

Задача 4.14. (8 т.) Да се напишат доказателства в HA^ω , че операциите от задача 3.15 за сравнение са съответно строга/частична наредба (за $<$, $>$, \leq , \geq) и релация на еквивалентност (за $=$).

Екстра кредит: (2 т.) Доказателствата да се опишат в λ -синтаксис или в системите Agda, Coq или MINLOG.

Задача 4.15. (5 т.) Да се напишат доказателства в HA^ω , които доказват коректността на операциите от задача 3.16 за намиране на частно и остатък при целочислено деление.

Екстра кредит: (2 т.) Доказателствата да се опишат в λ -синтаксис или в системите Agda, Coq или MINLOG.