УНИВЕРЗИТЕТ У БЕОГРАДУ
МАТЕМАТИЧКИ ФАКУЛТЕТ

Андрија Д. Урошевић

ХОМОТОПНА ТЕОРИЈА ТИПОВА

мастер рад

Београд, 2024.

Ментор:

др Сана Стојановић-Ђурђевић, доцент
Универзитет у Београду, Математички факултет

Чланови комисије:

др Филип Марић, редовни професор
Универзитет у Београду, Математички факултет

др Лаза Лазић, доцент
Универзитет у Београду, Математички факултет

Датум одбране: 29. фебруар 2024.

Мами, тати и деди

Наслов мастер рада: Хомотопна теорија типова

Резиме: Homotopy Type Theory/Univalent Foundations (HoTT/UF) is a revolutionary approach to the foundation of mathematics. Although it's revolutionary, HoTT/UF is very slowly gaining popularity among a broader circle of mathematicians and computer scientists. One of the reasons is that during formalization one requires both theoretical knowledge and proof-assistance skills. Acquiring those prerequisites is partially based on one's background. Mathematicians lack functional programming skills, on the other hand, computer scientists lack theoretical knowledge. A few materials tackle both areas, but they are lacking interactability. This thesis proposes a material that formalizes one theoretical area of HoTT/UF in Agda and is doing so while interacting with the user input.

Кључне речи: хомотопна теорија типова, интерактивно доказивање, агда

Садржај

Глава 1

Увод

- Хомотопна теорија типова = интуиционистичка теорија типова + високи индуктивни типови + аксиома унивалентности.
- Пер Мартин-Луф теорија типова се заснива на интиуционистичком програму који је настао по Брауверу.
- Математичко резтоновање је људска активност и математика је језик у коме се математичке идеје преносе.
- Фундаментална људска активност.
- Конструктивна теорија је *доказно релевантна*, тј. доказ је математички објекат као и сваки други.
- Тврђења можемо интерпретирати као типове, те ће доказ представљати *проверу типа*, тј. конструисање терма одређеног типа. (Јако битна уврнута идеја)
- Запажање: Хомотопна тероја и теорија типова представљају исту ствар.
- Хомотопна теорија се бави непрекидним пресликавањима која су *хомотопна* између себе, тј. могу се "непрекидно деформисати" једна у друге.
- Тројство израчуњивости: Програмерска интерпретација, хомотопна интерпретација и логичка интерпретација.
- Типско расуђивање $t : T$ читамо као t је терм типа или терм t настањује T. У програмерској интерпретацији тип представља тип, док терм

неког типа представља израз тог типа. У хомотопној интерпретацији тип представља простор, док терм неког типа представља тачку у том простору.

- Пример јединичног типа $\mathbb{1}$: јединични (`unit` у програмерском смислу), јединствени (*The* у логичком смислу), и контрактибилни (у хомотопном смислу) тип.
- Интенционални и екстенционални типови? (нешто чуно, проучити)
- Раселов парадокс као мотивација за теорију типова.

Глава 2

Интуиционистичка теорија типова

Интуиционистичка теорија типова или Пер Мартин-Луф теорија типова је математичка теорија конструкција. Тип представља врсту конструкције. Елемент, терм или тачка представља резултат конструкције неког типа. Прецизније, елемент a типа A записујемо као $a : A$, и кажемо да елемент a настањује тип A. Битно је напоменути да терм не може да "живи самостално" тј. терм увек мора да настањује неки тип.

Конструкција типова се састоји из низа дедуктивних *правила закључивања*. Правило закључивања записујемо као

$$\frac{\mathcal{H}_1 \qquad \mathcal{H}_2 \qquad \dots \qquad \mathcal{H}_n}{\mathcal{C}}$$

где расуђивања $\mathcal{H}_1, \mathcal{H}_2, \dots, \mathcal{H}_n$ називамо *премисе* или *хипотезе*, а расуђивање $\mathcal{C}$ називамо *закључак*.

Дефиниција 2.0.1. Свако *расуђивање* је облика $\Gamma \vdash \mathcal{J}$, где је Γ *контекст* и $\mathcal{J}$ *теза* расуђивања. Теза може имати четири врсте расуђивања и то су:

(i) A је *(добро-формиран) тип* у контексту Γ.

$$\Gamma \vdash A \text{ type}$$

(ii) A и B су *расуђивачки једнаки типови* у контексту Γ.

$$\Gamma \vdash A \equiv B \text{ type}$$

(iii) a је *елемент* типа A у контексту Γ.

$$\Gamma \vdash a : A$$

(iv) a и b су *расуђивачки једнаки елементи* типа A у контексту Γ.

$$\Gamma \vdash a \equiv_A b : A$$

користећи правила закључивања теорије типова.

Контекст је коначна листа *декларисаних променљивих* облика

$$x_1 : A_1, x_2 : A_2(x_1), \ldots, x_n : A_n(x_1, \ldots, x_{n-1}),$$

под условом да за свако $1 \leq k \leq n$ можемо да изведемо расуђивање

$$x_1 : A_1, x_2 : A_2(x_1), \ldots, x_{k-1} : A_{k-1}(x_1, \ldots, x_{k-2}) \vdash A_k(x_1, x_2, \ldots, x_{k-1}),$$

применом правила закључивања.

2.1 Правила закључивања

Интуиционистичка теорија типова, као и други математички формализми, захтева скуп правила закључивања на којим ће се формализам заснивати. Та правила називамо *структурна правила*.

Пример структурних правила закључивања која описују да је расуђивачка једнакост релација еквиваленције:

$$\frac{\Gamma \vdash A \text{ type}}{\Gamma \vdash A \equiv A \text{ type}} \qquad \frac{\Gamma \vdash A \equiv A' \text{ type}}{\Gamma \vdash A' \equiv A \text{ type}} \qquad \frac{\Gamma \vdash A \equiv A' \text{ type} \qquad \Gamma \vdash A' \equiv A'' \text{ type}}{\Gamma \vdash A \equiv A'' \text{ type}}$$

$$\frac{\Gamma \vdash a : A}{\Gamma \vdash a \equiv_A a : A} \qquad \frac{\Gamma \vdash a \equiv_A a' : A}{\Gamma \vdash a' \equiv_A a : A} \qquad \frac{\Gamma \vdash a \equiv_A a' : A \qquad \Gamma \vdash a' \equiv_A a'' : A}{\Gamma \vdash a \equiv_A a'' : A}$$

Исцрпна листа структурних правила закључивања у интуиционистичкој теорији типова се може наћи у [8]. Da li sada ovo raspisivati?

2.2 Зависни типови

Из дефиниције контекста можемо видети да неки типови зависе од других термова. На пример, $A_2(x_1)$ зависи од $x_1 : A_1$, тј. за разне термове $x_1 : A_1$ имамо разне типове $A_2(x_1)$. Ову идеју можемо уопштити помоћу следећих дефиниција:

Дефиниција 2.2.1. Нека је тип A у контексту Γ. *Фамилија* типова над A у контексту Γ је тип $B(x)$ у контексту $\Gamma, x : A$, тј.

$$\Gamma, x : A \vdash B(x) \text{ type}.$$

Кажемо да је B фамилија типова над A у контексту Γ. Алтернативно, кажемо да је $B(x)$ тип индексиран са $x : A$ у контексту Γ.

Дефиниција 2.2.2. Нека је B фамилија типова над A у контексут Γ. *Секција* фамилије B над типом A у контексту Γ је елемент типа $B(x)$ у контексту $\Gamma, x : A$, тј.

$$\Gamma, x : A \vdash b(x) : B(x).$$

Кажемо да је b секција фамилије B над A у контексту Γ. Алтернативно, кажемода да је $b(x)$ елемент типа $B(x)$ индексиран са $x : A$ у контексту $\Gamma, x : A$.

Дефиниција 2.2.3. Нека је B фамилија типова над A у контексту Γ, и нека је $a : A$. Кажемо да је $B[a/x]$ *влакно* од B за параметар a, где $B[a/x]$ представља замену свих појављивања x у B са a. Нит од B за параметар a краће записујемо као $B(a)$.

Дефиниција 2.2.4. Нека је b секција фамилије типова B над A у контексту Γ. Кажемо да је $b[a/x]$ *вредност* од b за параметар a, где $b[a/x]$ представља замену свих појављивања x у b са a. Такође, вредност од b за параметар a краће записујемо као $b(a)$.

2.3 Типови зависних функција

У математици заснованој на теорији скупова функција $f : A \to B$ дефинисана је над одређеним доменом A и кодоменом B. У теорији типова то не мора да буде случај, тј. кодомен може зависити од елемента над којим се функција

примељује. Прецизније, посматрајмо секцију b фамилије типова B над A у контексту Γ. Један начин је да b посматрамо као функцију $mapstob(x)$. Тада $b(x)$ настањује тип $B(x)$ који зависи од $x : A$. Због тога за разне елементе $x : A$ домена имамо разне кодомене, те има смисла говорити о типу *зависних функција* $\prod_{(x:A)} B(x)$.

Спецификација типа зависних функција $\prod_{(x:A)} B(x)$ је дата следећим правилима закључивања:

$$\frac{\Gamma, x : A \vdash B(x)\ \text{type}}{\Gamma \vdash \prod_{(x:A)} B(x)\ \text{type}}\ [\textstyle\prod\text{-form}] \qquad \frac{\Gamma, x : A \vdash b(x) : B(x)}{\Gamma \vdash \lambda x.b(x) : \prod_{(x:A)} B(x)}\ [\textstyle\prod\text{-intro}] \qquad \frac{\Gamma \vdash f : \prod_{(x:A)} B(x)}{\Gamma, x : A \vdash f(x) : B(x)}\ [\textstyle\prod\text{-elim}]$$

$$\frac{\Gamma, x : A \vdash b(x) : B(x)}{\Gamma \vdash (\lambda y.b(y))(x) \equiv b(x) : B(x)}\ [\textstyle\prod\text{-comp}_1] \qquad \frac{\Gamma \vdash f : \prod_{(x:A)} B(x)}{\Gamma \vdash \lambda x.f(x) \equiv f : \prod_{(x:A)} B(x)}\ [\textstyle\prod\text{-comp}_2]$$

Специјала случај типа зависних функција је тип (уобичајених) *функција* $A \to B$. Уколико су типови A и B у контексту Γ, тј. тип B не зависи од елемената типа A, тада $\prod_{(x:A)} B$ представља тип (уобичајених) функција.

Дефиниција 2.3.1. Тип (уобичајених) *функција* $A \to B$ дефинишемо као:

$$A \to B := \prod_{(x:A)} B.$$

Ако је $f : A \to B$ функција, тада је A *домен*, а B *кодомен* функције f.

Дефиниција 2.3.2. За сваки тип A дефинишемо *функцију идентитета* $\mathsf{id}_A : A \to A$ као $\mathsf{id}_A :\equiv \lambda x.x$.

Дефиниција 2.3.3. За свака три типа A, B, и C дефинишемо *композицију* $\mathsf{comp} : (B \to C) \to (A \to B) \to A \to C$ као $\mathsf{comp} :\equiv \lambda g.\lambda f.\lambda g(f(x))$.

Може се показати да је композиција асоцијативна, као и да је функција идентитета неутрал за композицију функција. Због сагласности типова имамо леви неутрал id_B и десни неутрал id_A.

2.4 Индуктивни типови

Поред типова зависних функција постоји и класа *индуктивних типова*. Сваки индуктивни тип се дефинише помоћу следеће спецификације:

(i) *Формирање* типа описује начин на који се дати тип формира.

(ii) *Конструисања* описује на који начин се уводе нови канонични термови датог типа.

(iii) *Индуктивни принцип* описује податке који су потреби да би се конструисала секција произвољне фамилије типова над датим типом.

(iv) *Правила израчунавања* захтевају да се индуктивно дефинисана секција произвољне фамилије типова над датим типом слаже по конструкторима који уводе нове каноничне термове.

Обично се, поред ових спецификације, уводи и *правило рекурзије* које је специјални случај правила индукције. Код правила рекурзије не конструишемо секцију произвољне фамилије типова над датим типом, већ само константну фамилију над датим типом.

Сада ће бити дате спецификације за неке уобичајене индуктивне типове.

Тип природних бројева

$$\frac{}{\vdash \mathbb{N}\ \text{type}}\ [\mathbb{N}\text{-form}] \qquad \frac{}{\vdash 0_{\mathbb{N}} : \mathbb{N}}\ [\mathbb{N}\text{-intro}_{0_{\mathbb{N}}}] \qquad \frac{}{\vdash \mathsf{succ}_{\mathbb{N}} : \mathbb{N} \to \mathbb{N}}\ [\mathbb{N}\text{-intro}_{\mathsf{succ}_{\mathbb{N}}}]$$

$$\frac{\begin{array}{l}\Gamma, n : \mathbb{N} \vdash P(n)\ \text{type} \\ \Gamma \vdash p_{0_{\mathbb{N}}} : P(0_{\mathbb{N}}) \\ \Gamma \vdash p_{\mathsf{succ}_{\mathbb{N}}} : \prod_{(n:\mathbb{N})} P(n) \to P(\mathsf{succ}_{\mathbb{N}}(n))\end{array}}{\Gamma \vdash \mathsf{ind}_{\mathbb{N}}(p_{0_{\mathbb{N}}}, p_{\mathsf{succ}_{\mathbb{N}}}) : \prod_{(n:\mathbb{N})} P(n)}\ [\mathbb{N}\text{-ind}]$$

$$\frac{\begin{array}{l}\Gamma, n : \mathbb{N} \vdash P(n)\ \text{type} \\ \Gamma \vdash p_{0_{\mathbb{N}}} : P(0_{\mathbb{N}}) \\ \Gamma \vdash p_{\mathsf{succ}_{\mathbb{N}}} : \prod_{(n:\mathbb{N})} P(n) \to P(\mathsf{succ}_{\mathbb{N}}(n))\end{array}}{\Gamma \vdash \mathsf{ind}_{\mathbb{N}}(p_{0_{\mathbb{N}}}, p_{\mathsf{succ}_{\mathbb{N}}}, 0_{\mathbb{N}}) \equiv p_{0_{\mathbb{N}}} : P(0_{\mathbb{N}})}\ [\mathbb{N}\text{-comp}^{\mathsf{ind}_{\mathbb{N}}}_{0_{\mathbb{N}}}]$$

$$\frac{\begin{array}{l}\Gamma, n : \mathbb{N} \vdash P(n)\ \text{type} \\ \Gamma \vdash p_{0_{\mathbb{N}}} : P(0_{\mathbb{N}}) \\ \Gamma \vdash p_{\mathsf{succ}_{\mathbb{N}}} : \prod_{(n:\mathbb{N})} P(n) \to P(\mathsf{succ}_{\mathbb{N}}(n))\end{array}}{\Gamma,\ n : \mathbb{N} \vdash \mathsf{ind}_{\mathbb{N}}(p_{0_{\mathbb{N}}}, p_{\mathsf{succ}_{\mathbb{N}}}, \mathsf{succ}_{\mathbb{N}}(n)) \equiv p_{\mathsf{succ}_{\mathbb{N}}}(n, \mathsf{ind}_{\mathbb{N}}(p_{0_{\mathbb{N}}}, p_{\mathsf{succ}_{\mathbb{N}}}, n)) : P(\mathsf{succ}_{\mathbb{N}}(n))}\ [\mathbb{N}\text{-comp}^{\mathsf{ind}_{\mathbb{N}}}_{\mathsf{succ}_{\mathbb{N}}}]$$

$$[\mathbb{N}\text{-rec}_{\mathbb{N}}]\ \frac{\begin{array}{l}\Gamma \vdash A\ \text{type} \\ \Gamma \vdash a_{0_{\mathbb{N}}} : A \\ \Gamma \vdash a_{\mathsf{succ}_{\mathbb{N}}} : \mathbb{N} \to A \to A\end{array}}{\Gamma \vdash \mathsf{rec}_{\mathbb{N}}(a_{0_{\mathbb{N}}}, a_{\mathsf{succ}_{\mathbb{N}}}) : \mathbb{N} \to A}$$

$$\frac{\begin{array}{l}\Gamma \vdash A\ \text{type} \\ \Gamma \vdash a_{0_{\mathbb{N}}} : A \\ \Gamma \vdash a_{\mathsf{succ}_{\mathbb{N}}} : \mathbb{N} \to A \to A\end{array}}{\Gamma \vdash \mathsf{rec}_{\mathbb{N}}(a_{0_{\mathbb{N}}}, a_{\mathsf{succ}_{\mathbb{N}}}, 0_{\mathbb{N}}) \equiv a_{0_{\mathbb{N}}} : A}\ [\mathbb{N}\text{-comp}^{\mathsf{rec}_{\mathbb{N}}}_{0_{\mathbb{N}}}]$$

$$\frac{\begin{array}{l}\Gamma \vdash A\ \text{type} \\ \Gamma \vdash a_{0_{\mathbb{N}}} : A \\ \Gamma \vdash a_{\mathsf{succ}_{\mathbb{N}}} : \mathbb{N} \to A \to A\end{array}}{\Gamma, n : \mathbb{N} \vdash \mathsf{rec}_{\mathbb{N}}(a_{0_{\mathbb{N}}}, a_{\mathsf{succ}_{\mathbb{N}}}, \mathsf{succ}_{\mathbb{N}}(n)) \equiv a_{\mathsf{succ}_{\mathbb{N}}}(n, \mathsf{rec}_{\mathbb{N}}(a_{0_{\mathbb{N}}}, a_{\mathsf{succ}_{\mathbb{N}}}, n)) : A}\ [\mathbb{N}\text{-comp}^{\mathsf{rec}_{\mathbb{N}}}_{\mathsf{succ}_{\mathbb{N}}}]$$

Тип природних бројева $\mathbb{N}$ може да се формира из празног контекста што нам говори правило $\mathbb{N}$-form. Другим речима, постојање тип природних бројева $\mathbb{N}$ не зависи од постојања других типова. Даље, имамо два конструктора помоћу којих конструишемо све каноничке термове типа $\mathbb{N}$. Први конструктор је константа $0_{\mathbb{N}} : \mathbb{N}$ и он говори да је $0_{\mathbb{N}}$ канонични терм типа $\mathbb{N}$. Други конструктор је функција $\mathsf{succ}_{\mathbb{N}} : \mathbb{N} \to \mathbb{N}$ и она говори да ће $\mathsf{succ}_{\mathbb{N}}(n)$ бити канонични терм тип $\mathbb{N}$ ако је $n : \mathbb{N}$ кононични терм. Због тога су $0_{\mathbb{N}}, \mathsf{succ}_{\mathbb{N}}(0_{\mathbb{N}}), \mathsf{succ}_{\mathbb{N}}(\mathsf{succ}_{\mathbb{N}}(0_{\mathbb{N}})), \ldots$ канонични термови који настањују тип $\mathbb{N}$.

Правила формирања и конструкције нам говоре о томе под којим условима ће нешто бити тип, и како конструисати канонične термове тог типа. Недостаје начин на који се тип и његови термови користе. Због тога се

уводи индуктивно правило и правила израчунавања. Да би конструисали $\mathsf{ind}_{\mathbb{N}}(p_{0_{\mathbb{N}}}, p_{\mathsf{succ}_{\mathbb{N}}}) : \prod_{(n:\mathbb{N})} P(n)$ потребно је конструисати $p_{0_{\mathbb{N}}} : P(0_{\mathbb{N}})$ (*база индукције*) і $p_{\mathsf{succ}_{\mathbb{N}}} : \prod_{n:\mathbb{N}} P(n) \to P(\mathsf{succ}_{\mathbb{N}}(n))$ (*индуктивни корак*). Даље, за сваки од конструктора треба увести правило израчунавања тако да се понаша у складу са зависном функцијом $\mathsf{ind}_{\mathbb{N}}(p_{0_{\mathbb{N}}}, p_{\mathsf{succ}_{\mathbb{N}}}) : \prod_{(n:\mathbb{N})} P(n)$. Због тога имамо правила два правила израчунавања $\mathbb{N}\text{-comp}_{0_{\mathbb{N}}}$ і $\mathbb{N}\text{-comp}_{\mathsf{succ}_{\mathbb{N}}}$.

Специјални случај индукције типа природних бројева је рекурзија типа природних бројева, у којој тип P не зависи од $\mathbb{N}$. Тада добијамо функцију $\mathsf{rec}_{\mathbb{N}}(a_{0_{\mathbb{N}}}, a_{\mathsf{succ}_{\mathbb{N}}}) : \mathbb{N} \to A$, под условом да имамо елементе $a_{0_{\mathbb{N}}} : A$ и $a_{\mathsf{succ}_{\mathbb{N}}} : \mathbb{N} \to A \to A$.

Правило индукције, заједно са правилом рекурзије, омогућава дефинисање разних функција над природним бројевима. Да би дефинисали операцију сабирања природних бројева $+_{\mathbb{N}} : \mathbb{N} \to \mathbb{N} \to \mathbb{N}$ можемо искористити правило рекурзије, тј. функцију $\mathsf{rec}_{\mathbb{N}} : A \to (\mathbb{N} \to A \to A) \to \mathbb{N} \to A$. За тип A узећемо $\mathbb{N}$. Због тога, сабирање природних бројева дефинишемо као

$$m +_{\mathbb{N}} n :\equiv \mathsf{rec}_{\mathbb{N}}(m, \lambda n.\lambda r.\mathsf{succ}_{\mathbb{N}}(r), n).$$

Заиста, за овако дефинисану операцију сабирања важи:

$$m +_{\mathbb{N}} 0_{\mathbb{N}} \equiv m;$$
$$m +_{\mathbb{N}} \mathsf{succ}_{\mathbb{N}}(n) \equiv \mathsf{succ}_{\mathbb{N}}(m +_{\mathbb{N}} n).$$

Слично, множење природних бројева $\times_{\mathbb{N}} : \mathbb{N} \to \mathbb{N} \to \mathbb{N}$ можемо дефинисати као

$$m \times_{\mathbb{N}} m :\equiv \mathsf{rec}_{\mathbb{N}}(0_{\mathbb{N}}, \lambda n.\lambda r.m +_{\mathbb{N}} r, n).$$

Такође, за овако дефинисану операцију множења важи:

$$m \times_{\mathbb{N}} 0_{\mathbb{N}} \equiv 0_{\mathbb{N}};$$
$$m \times_{\mathbb{N}} \mathsf{succ}_{\mathbb{N}}(n) \equiv (m +_{\mathbb{N}} (m \times_N n)).$$

Можемо приметити шаблон између дефинисања операција преко рекурзивног правила и правила која захтевамо да веже по конструкторима. Наиме, уколико желимо да дефинишемо функцију $f : \mathbb{N} \to A$ за коју важи:

$$f(0_{\mathbb{N}}) \equiv \Phi_{0_{\mathbb{N}}};$$
$$f(\mathsf{succ}_{\mathbb{N}}(n)) \equiv \Phi_{\mathsf{succ}_{\mathbb{N}}},$$

где је $\Phi_{0_{\mathbb{N}}}$ израз типа A, и $\Phi_{\mathsf{succ}_{\mathbb{N}}}$ израз типа A који може садржати n и $f(n)$. Тада функцију $f : \mathbb{N} \to A$ дефинишемо као

$$f :\equiv \mathsf{rec}_{\mathbb{N}}(\Phi_{0_{\mathbb{N}}}, \lambda n.\lambda r.\Phi'_{\mathsf{succ}_{\mathbb{N}}}),$$

где $\Phi'_{\mathsf{succ}_{\mathbb{N}}}$ добијемо из $\Phi_{\mathsf{succ}_{\mathbb{N}}}$ тако што сва појављивања $f(n)$ заменимо са r. Овај поступак дефинисања можемо уопштити и на индуктивно правило, и тада се он назива *упаривање шаблона* (енгл. *pattern matching*). hmm, da li se ovo ovako prevodi? pitanje za Čukića

Празни тип

$$[\mathbb{0}\text{-form}]\ \frac{}{\vdash \mathbb{0}\ \text{type}} \qquad [\mathbb{0}\text{-ind}]\ \frac{\Gamma, 0 \vdash P(x)\ \text{type}}{\Gamma \vdash \mathsf{ind}_{\mathbb{0}} : \prod_{(0)} P(x)} \qquad [\mathbb{0}\text{-rec}]\ \frac{\Gamma \vdash A\ \text{type}}{\Gamma \vdash \mathsf{rec}_{\mathbb{0}} : \mathbb{0} \to A}$$

Празан тип $\mathbb{0}$ је дегенерисани пример индуктивног типа који нема ни један конструктор, самим тим нема ни једно правило израчунавања. Може да се формира из празног контекста, а његово правило индукције тврди да за било коју фамилију типове P над $\mathbb{0}$ постоји $\mathsf{ind}_{\mathbb{0}} : \prod_{(x:\mathbb{0})} P(x)$. Чешће се користи правило рекурзије које тврди да уколико конструишемо елемент $x : \mathbb{0}$, онда можемо да конструишемо елемент $\mathsf{rec}_{\mathbb{0}}(x) : A$ било ког типа A. Правило рекурзије за празни тип $\mathbb{0}$ се обично назива и *правило контрадикције* или *противречност*.

Дефиниција 2.4.1. За сваки тип A дефинишемо тип *негације* од A као $\neg A := A \to \mathbb{0}$. Поред тога, кажемо да је тип A *празан* ако његову негацију настањује неки елемент, тј. $\mathsf{empty}(A) := A \to \mathbb{0}$.

Приметимо да је *дупла негација* од A дефинисана као $\neg\neg A := (A \to \mathbb{0}) \to \mathbb{0}$. Због тога, не мора да важи $\neg\neg A \to A$, те није могуће изводити доказе контрадикцијом.

Јединични тип

$$[\mathbb{1}\text{-form}]\ \frac{}{\vdash \mathbb{1}\ \text{type}} \qquad [\mathbb{1}\text{-intro}_\star]\ \frac{}{\vdash \star : \mathbb{1}}$$

$$[\mathbb{1}\text{-ind}]\ \frac{\Gamma, x : \mathbb{1} \vdash P(x)\ \text{type} \quad \Gamma \vdash p_\star : P(\star)}{\Gamma \vdash \mathsf{ind}_{\mathbb{1}}(p_\star) : \prod_{(x:\mathbb{1})} P(x)} \qquad [\mathbb{1}\text{-comp}]\ \frac{\Gamma, 1 \vdash P(x)\ \text{type} \quad \Gamma \vdash p_\star : P(\star)}{\Gamma \vdash \mathsf{ind}_{\mathbb{1}}(p_\star, \star) \equiv p_\star : P(\star)}$$

$$[\mathbb{1}\text{-rec}]\ \frac{\Gamma \vdash A\ \text{type} \quad \Gamma \vdash a : A}{\Gamma \vdash \mathsf{rec}_{\mathbb{1}}(a) : \mathbb{1} \to A}$$

Јединични тип $\mathbb{1}$ је индуктивни тип кога настањује само елемент $\star$. Може да се формира из празног контекста, а његово правило индукције тврди да за било коју фамилију типова P над $\mathbb{1}$ постоји $\mathsf{ind}_{\mathbb{1}}(p_\star) : \prod_{(x:\mathbb{1})} P(x)$ уколико постоји елемент $p_\star : P(\star)$. Како постоји само један конструктор $\star : \mathbb{1}$, имамо једно правило израчунавања које треба да се сложи са индуктивним правилом, и то као $\mathsf{ind}_{\mathbb{1}}(p_\star, \star) \equiv p_\star : P(\star)$.

Специјални случај правила индукције типа $\mathbb{1}$ је правило рекурзије типа $\mathbb{1}$, које добијамо када фамилија типова P над $\mathbb{1}$ не зависи од $x : \mathbb{1}$. Тада за сваки елемент $a : A$ имамо функцију $\mathsf{rec}_{\mathbb{1}}(a) : \mathbb{1} \to A$.

Дефиниција 2.4.2. За сваки тип A дефинишемо тип *јединствене функције* од A као $!\mathbb{1}(A) := A \to \mathbb{1}$. Специјално, јединствена функција од $\mathbb{0}$, тј. $\mathbb{0} \to \mathbb{1}$, се назива *вакумска функција*.

У хомотопној теоријити типова за вакумску функцију важи да је јединствена.

Типови копроизвода

$$\frac{\Gamma \vdash A \text{ type}, B \text{ type}}{\Gamma \vdash A + B \text{ type}}\ [+\text{-form}] \qquad \frac{}{\Gamma \vdash \mathsf{inl} : A \to A + B}\ [+\text{-intro}_{\mathsf{inl}}] \qquad \frac{}{\Gamma \vdash \mathsf{inr} : B \to A + B}\ [+\text{-intro}_{\mathsf{inr}}]$$

$$[+\text{-ind}]\quad \frac{\begin{array}{l}\Gamma, z : A + B \vdash P(z) \text{ type} \\ \Gamma \vdash p_{\mathsf{inl}} : \prod_{(a:A)} P(\mathsf{inl}(a)) \\ \Gamma \vdash p_{\mathsf{inr}} : \prod_{(b:B)} P(\mathsf{inr}(b))\end{array}}{\Gamma \vdash \mathsf{ind}_+(p_{\mathsf{inl}}, p_{\mathsf{inr}}) : \prod_{(z:A+B)} P(z)}$$

$$[+\text{-comp}]\quad \frac{\begin{array}{l}\Gamma, z : A + B \vdash P(z) \text{ type} \\ \Gamma \vdash p_{\mathsf{inl}} : \prod_{(a:A)} P(\mathsf{inl}(a)) \\ \Gamma \vdash p_{\mathsf{inr}} : \prod_{(b:B)} P(\mathsf{inr}(b))\end{array}}{\begin{array}{l}\Gamma, a : A \vdash \mathsf{ind}_+(p_{\mathsf{inl}}, p_{\mathsf{inr}}, \mathsf{inl}(a)) \equiv p_{\mathsf{inl}}(a) : P(\mathsf{inl}(a)) \\ \Gamma, b : B \vdash \mathsf{ind}_+(p_{\mathsf{inl}}, p_{\mathsf{inr}}, \mathsf{inr}(b)) \equiv p_{\mathsf{inr}}(b) : P(\mathsf{inr}(b))\end{array}}$$

$$[+\text{-rec}]\quad \frac{\begin{array}{l}\Gamma \vdash \mathit{type} \\ \Gamma \vdash f : A \to X \\ \Gamma \vdash g : B \to X\end{array}}{\Gamma \vdash \mathsf{rec}_+(f, g) : A + B \to X}$$

За типове A и B из контекста Γ можемо дефинисати тип копроизвода $A + B$ кога ће настањивати елементи из типа A или из типа B. Због тога тип копроизвода $A + B$ има два конструктора $\mathsf{inl} : A \to A + B$ і $\mathsf{inl} : B \to A + B$. Правило индукције тврди да за било коју фамилију типова P над $A + B$ постоји елемент $\mathsf{ind}_+(p_{\mathsf{inl}}, p_{\mathsf{inr}}) : \prod_{(z:A+B)} P(z)$ уколико постоје елементи $p_{\mathsf{inl}} : \prod_{(a:A)} P(\mathsf{inl}(a))$ и $p_{\mathsf{inr}} : \prod_{(b:B)} P(\mathsf{inr}(b))$. Како постоје два конструктора, имамо два правила израчунавања која треба да се сложе са правилом индукције, и то као $\mathsf{ind}_+(p_{\mathsf{inl}}, p_{\mathsf{inr}}, \mathsf{inl}(a)) \equiv p_{\mathsf{inl}}(a) : P(\mathsf{inl}(a))$ и $\mathsf{ind}_+(p_{\mathsf{inl}}, p_{\mathsf{inr}}, \mathsf{inr}(b)) \equiv p_{\mathsf{inr}}(b) : P(\mathsf{inr}(b))$.

Специјални случај правила индукције типа $A + B$ је правило рекурзије типа $A + B$, које добијамо када фамилија типова P над $A + B$ не зависи од $z : A + B$. Тада за сваку функцију $f : A \to X$ и за сваку функцију $g : B \to X$ имамо функцију $\mathsf{rec}_+(f, g) : A + B \to X$. Из правила индукције, за свако $f : A \to X$ и за свако $g : B \to Y$, имамо функцију $f + g : A + B \to X + Y$.

Специјални случај типа копроизвода је Буловски тип $2 := \mathbb{1} + \mathbb{1}$, чије једине елементе дефинишемо као $\mathsf{true} :\equiv \mathsf{inl}(\star)$, $\mathsf{false} :\equiv \mathsf{inr}(\star)$. Из спецификације типа

копроизвода можемо извући правило индукције и правило израчунавања, за Буловски тип 2. Правило индукције 2-ind се назива и *if-then-else*.

$$[2\text{-ind}]\quad \frac{\begin{array}{l}\Gamma, x : 2 \vdash P(x)\ \text{type}\\ \Gamma \vdash p_{\mathsf{true}} : P(\mathsf{true})\\ \Gamma \vdash p_{\mathsf{false}} : P(\mathsf{false})\end{array}}{\Gamma \vdash \mathsf{ind}_2(p_{\mathsf{true}}, p_{\mathsf{false}}) : \prod_{(x:2)} P(x)}$$

$$[2\text{-comp}]\quad \frac{\begin{array}{l}\Gamma, x : 2 \vdash P(x)\ \text{type}\\ \Gamma \vdash p_{\mathsf{true}} : P(\mathsf{true})\\ \Gamma \vdash p_{\mathsf{false}} : P(\mathsf{false})\end{array}}{\begin{array}{l}\Gamma \vdash \mathsf{ind}_2(p_{\mathsf{true}}, p_{\mathsf{false}}, \mathsf{true}) \equiv p_{\mathsf{true}} : P(\mathsf{true})\\ \Gamma \vdash \mathsf{ind}_2(p_{\mathsf{true}}, p_{\mathsf{false}}, \mathsf{false}) \equiv p_{\mathsf{false}} : P(\mathsf{true})\end{array}}$$

Типови зависних парова

$$\frac{[\textstyle\sum\text{-form}]\quad \Gamma, x : A \vdash B(x)\ \text{type}}{\Gamma \vdash \sum_{(x:A)} B(x)\ \text{type}} \qquad \frac{[\textstyle\sum\text{-intro}]\quad \Gamma, x : A \vdash y(x) : B(x)}{\Gamma \vdash (x, y(x)) : \sum_{(x:A)} B(x)}$$

$$[\textstyle\sum\text{-ind}]\quad \frac{\begin{array}{l}\Gamma, (x,y) : \sum_{(x:A)} B(x) \vdash P((x,y))\ \text{type}\\ \Gamma \vdash f : \prod_{(x:A)} \prod_{(y:B(x))} P((x,y))\end{array}}{\Gamma \vdash \mathsf{ind}_{\sum}(f) : \prod_{(p:\sum_{(x:A)} B(x))} P(p)}$$

$$[\textstyle\sum\text{-comp}]\quad \frac{\begin{array}{l}\Gamma, (x,y) : \sum_{(x:A)} B(x) \vdash P((x,y))\ \text{type}\\ \Gamma \vdash f : \prod_{(x:A)} \prod_{(y:B(x))} P((x,y))\end{array}}{\Gamma, (x,y) : \sum_{(x:A)} B(x) \vdash \mathsf{ind}_{\sum}(f, (x,y)) \equiv f(x,y) : P((x,y))}$$

Ако је B фамилија типова над A из контекста Γ, онда можемо формирати тип зависних парова $\sum_{(x:A)} B(x)$ кога ће настањивати парови $(x, y(x))$, где је $x : A$ и $y(x) : B(x)$. Правило индукције тврди да за било коју фамилију типова P над $\sum_{(x:A)} B(x)$ постоји елемент $\mathsf{ind}_{\sum}(f) : \prod_{p:\sum_{(x:A)} B(x)} P(p)$ уколико постоје елемент $f : \prod_{(x:A)} \prod_{(y:B(x))} P((x,y))$. Како постоји само један конструктор, имамо само једно правило израчунавања које треба да се сложи са правилом индукције, и то као $\mathsf{ind}_{\sum}(f, (x,y)) \equiv f(x,y) : P((x,y))$.

Правило индукције нам омогућава да дефинишемо следеће функције:

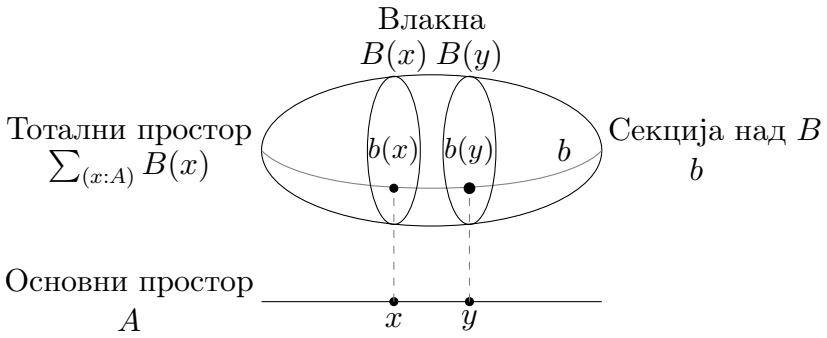
Дефиниција 2.4.3. Нека је B фамилија типова над A. Тада елемент $\mathsf{pr}_1 : \sum_{(x:A)} B(x) \to A$ *пројекције на први елемент* дефинишемо као:

$$\mathsf{pr}_1((a,b)) :\equiv\ a, \tag{2.1}$$

а елемент $\mathsf{pr}_2 : \prod_{p:\sum_{(x:A)} B(x)} B(\mathsf{pr}_1(p))$ *пројекције на други елемент* дефинишемо као:

$$\mathsf{pr}_2((a,b)) :\equiv b. \tag{2.2}$$

Ако претпоставимо да имамо елемент $f : \prod_{((x,y):\sum_{(x:A)} B(x))} P((x,y))$ тада конструишемо елемент типа $\prod_{(x:A)} \prod_{(y:B(x))} P((x,y))$ као $\lambda x.\lambda y.f((x,y))$. Ова конструкција се назива *каријевање*, и како је супртона правилу $\sum$-ind, правило $\sum$-ind често наивамо *одкаријевање* (je l je ovo uopste rec, kako se prevodi uncarry).



Слика 2.1: Геометријска репрезентација типа зависних парова.

Специјални случај типа зависних парова је тип (независних) *парова* или *(Декартов) производ* $A \times B := \sum_{(x:A)} B$.

$$[\times\text{-ind}]\quad \frac{\Gamma, (x,y) : A \times B \vdash P((x,y)) \text{ type} \qquad \Gamma \vdash f : \prod_{(x:A)} \prod_{(y:B)} P((x,y))}{\Gamma \vdash \mathsf{ind}_\times(f) : \prod_{(p:A\times B)} P(p)}$$

$$[\times\text{-comp}]\quad \frac{\Gamma, (x,y) : A \times B \vdash P((x,y)) \text{ type} \qquad \Gamma \vdash f : \prod_{(x:A)} \prod_{(y:B)} P((x,y))}{\Gamma, (x,y) : A \times B \vdash \mathsf{ind}_\times(f,(x,y)) \equiv f(x,y) : P((x,y))}$$

2.5 Искази као типови

Кари-Хавардова интерпретација неформално посматра исказе као типове, доказе као елементе типова, и предикате као фамилије типова. Да би показали да је исказ тачан у теорији типова треба конструисати елемент који настањује одговарајући тип. Прецизније, за дати исказ A (добро-формирани

Искази	Типови
$\bot$	$\mathbb{0}$
$\top$	$\mathbb{1}$
$A \vee B$	$A + B$
$A \wedge B$	$A \times B$
$A \implies B$	$A \to B$
$A \iff B$	$(A \to B) \times (B \to A)$
$\neg A$	$A \to \mathbb{0}$
$\forall x.P(x)$	$\prod_{(x:A)} P(x)$
$\exists x.P(x)$	$\sum_{(x:A)} P(x)$

Табела 2.1: Кари-Хавардова интерпретација

тип) уколико конструишемо елемент $x : A$ (кога често називамо и *сведок* за A) тада сматрамо да је исказ A тачан. Приметимо да исказ није тачан или нетачан, већ да представља колекцију својих сведока који сведоче о његовој истинитости. Због тога докази нису начин на који се комуницира о математичким објектима, већ су и они сами математички објекти. У табели 2.1 приказани су искази заједно са њиховим одговарајућом интерпретацијом у теорији типова.

Прокоментаришимо неке интерпретације из табеле 2.1. Да би показали да важи $A \implies B$ треба претпоставити да важи A и доказати да важи B. У теорији типова треба конструисати елемент типа $A \to B$, тј. треба конструисати елемент типа B који потенцијално користи претпостављени елемент типа A. Слично, да би показали $\exists x.P(x)$ у теорији типова треба конструисати елемент типа $\sum_{(x:A)} P(x)$. У овом случају теорија типова нам даје и више од тога. Наиме, P је фамилија типова, што значи да $P(x)$ не мора да буде типа 2, тј. P не мора да буде предикат. Поред тога, тип $\sum_{(x:A)} P(x)$ можемо схватити као тип свих елемената $x : A$ za koje $P(x)$.

2.6 Хијерархија универзума и универзум типови

Универзум типови се могу сматрати као типови кога настањују други типови. Универзум тип $\mathcal{U}$ омогућава да се исказ A type запише формално као $A : \mathcal{U}$. Поред тога, омогућава да се фамилија типова B над типом A

дефинише као функција $B : A \to \mathcal{U}$.

Желимо да типови који могу да се формирају из празног контекста настањују универзум $\mathcal{U}$ (то су, на пример, $\mathbb{0}$, $\mathbb{1}$, и $\mathbb{N}$). Штавише како универзум $\mathcal{U}$ настањују и други типова, желимо да универзум $\mathcal{U}$ буде затворен по свим конструкторима који користе типове универзума $\mathcal{U}$. На пример, ако је $A : \mathcal{U}$ и $B : A \to \mathcal{U}$, онда $\prod_{(x:A)} B(x) : \mathcal{U}$. Међутим, не сме доћи то тога да универзум настањује сам себе, тј. не сме да важи $\mathcal{U} : \mathcal{U}$ (Раселов парадокс).

У многим случајевима довољно је постојање једног универзума $\mathcal{U}$, међутим, некада желимо да универзум настањује неки други универзум. Како би избегли Раселов парадокс захтевамо постојање *хијерархије универзума*

$$\mathcal{U}_0, \quad \mathcal{U}_1, \quad \mathcal{U}_2, \quad \ldots \tag{2.3}$$

за коју важе следећа правила:

$$[\mathcal{U}\text{-intro}]\ \frac{}{\Gamma \vdash \mathcal{U}_i : \mathcal{U}_{i+1}} \qquad [\mathcal{U}\text{-cumul}]\ \frac{\Gamma \vdash A : \mathcal{U}_i}{\Gamma \vdash A : \mathcal{U}_{i+1}}$$

Универзум $\mathcal{U}_0$ називамо *базни универзум*. Базни универзум настањују типови који могу да се формирају из празног контекста, као и сви типови чији конструктори користе типове који се већ налазе у базном универзуму. За универзум $\mathcal{U}_i$ има смисла посматрати и $\mathcal{U}_{i+1}$ кога називамо и *универзум следбеник*. Често није битно знати редни број универзума у хијерархији, те се следбеник универзума $\mathcal{U}$ обележава са $\mathcal{U}^+$. За два универзума $\mathcal{U}$ и $\mathcal{V}$ можемо дефинисати њихову најмању горња границу $\mathcal{U} \sqcup \mathcal{V}$. На пример, за $\mathcal{U}_0$ і $\mathcal{U}_1$, најмања горња граница $\mathcal{U}_0 \sqcup \mathcal{U}_1$ је $\mathcal{U}_1$.

2.7 Типови идентитети

Шта значи да су елементи неког типа једнаки?

$$\frac{\Gamma \vdash A \text{ type} \qquad \Gamma \vdash x : A \qquad \Gamma \vdash y : A}{\Gamma \vdash x =_A y \text{ type}}\ [=\text{-form}] \qquad [=\text{-intro}]\ \frac{\Gamma \vdash A \text{ type} \qquad \Gamma \vdash x : A}{\Gamma \vdash \mathsf{refl}_x : x =_A x}$$

Indukcija putanje

$$\text{[=-ind]}\quad \frac{\begin{array}{l}\Gamma, x : A, y : A, p : x =_A y \vdash P(x, y, p)\ \text{type} \\ \Gamma \vdash f : \prod_{(x:A)} P(x, x, \mathsf{refl}_x)\end{array}}{\Gamma \vdash \mathsf{ind}_= : \prod_{(x,y:A)} \prod_{(p:x=_A y)} P(x, y, p)}$$

$$\text{[=-comp]}\quad \frac{\begin{array}{l}\Gamma, x : A, y : A, p : x =_A y \vdash P(x, y, p)\ \text{type} \\ \Gamma \vdash f : \prod_{(x:A)} P(x, x, \mathsf{refl}_x)\end{array}}{\Gamma, x : A \vdash \mathsf{ind}_=(x, x, \mathsf{refl}_x) \equiv f(x) : P(x, x, \mathsf{refl}_x)}$$

Transport

Akcije nad putanjama

2.8 Ekvivalentnosti

Functional extentionality

Ekvivalentnosti i univerzalna osobina

2.9 Aksioma univalentnosti

Neke posledice univalentnosti

Глава 3

Агда

Глава 4

Закључак

Литература

[1] Guillaume Brunerie и др. *Homotopy Type Theory in Agda*. URL: `https://github.com/HoTT/HoTT-Agda`.

[2] Cyril Cohen и др. *Cubical Type Theory: a constructive interpretation of the univalence axiom*. 2016. arXiv: `1611.02108 [cs.LO]`.

[3] Thierry Coquand, Simon Huber и Anders Mörtberg. *On Higher Inductive Types in Cubical Type Theory*. 2018. arXiv: `1802.01170 [cs.LO]`.

[4] Martín Hötzel Escardó. „Introduction to univalent foundations of mathematics with Agda". У: *arXiv preprint arXiv:1911.00580* (2019).

[5] Jackson Macor. *A brief introduction to type theory and the univalence axiom*. 2015.

[6] Per Martin-Löf и Giovanni Sambin. *Intuitionistic type theory*. Св. 9. Bibliopolis Naples, 1984.

[7] Ulf Norell. „Dependently typed programming in Agda". У: *Proceedings of the 4th international workshop on Types in language design and implementation*. 2009, стр. 1–2.

[8] Egbert Rijke. *Introduction to Homotopy Type Theory*. 2022. arXiv: `2212.11082 [math.LO]`.

[9] The Univalent Foundations Program. *Homotopy Type Theory: Univalent Foundations of Mathematics*. Institute for Advanced Study: `https://homotopytypetheory.org/book`, 2013.

[10] Andrea Vezzosi, Anders Mörtberg и Andreas Abel. „Cubical Agda: A dependently typed programming language with univalence and higher inductive types". У: *Journal of Functional Programming* 31 (2021), e8.

[11] Vladimir Voevodsky, Benedikt Ahrens, Daniel Grayson и др. *UniMath — a computer-checked library of univalent mathematics*. available at `http://unimath.org`. DOI: `10.5281/zenodo.7848572`. URL: `https://doi.org/10.5281/zenodo.7848572`.

[12] Philip Wadler. „Propositions as Types". У: *Commun. ACM* 58.12 (Нов. 2015), стр. 75–84. ISSN: 0001-0782. DOI: `10.1145/2699407`. URL: `https://doi.org/10.1145/2699407`.

Биографија аутора

Вук Стефановић Караџић (*Тршић, 26. октобар/6. новембар 1787. — Беч, 7. фебруар 1864.*) био је српски филолог, реформатор српског језика, сакупљач народних умотворина и писац првог речника српског језика. Вук је најзначајнија личност српске књижевности прве половине XIX века. Стекао је и неколико почасних доктората. Учествовао је у Првом српском устанку као писар и чиновник у Неготинској крајини, а након слома устанка преселио се у Беч, 1813. године. Ту је упознао Јернеја Копитара, цензора словенских књига, на чији је подстицај кренуо у прикупљање српских народних песама, реформу ћирилице и борбу за увођење народног језика у српску књижевност. Вуковим реформама у српски језик је уведен фонетски правопис, а српски језик је потиснуо славеносрпски језик који је у то време био језик образованих људи. Тако се као најважније године Вукове реформе истичу 1818., 1836., 1839., 1847. и 1852.