UNIVERZA V LJUBLJANI
FAKULTETA ZA MATEMATIKO IN FIZIKO

Matematika – 1. stopnja

Mabel Najdovski

OBRNLJIVE FUNKCIJE SO SFERE V SVETU

Delo diplomskega seminarja

Mentorica: izr. prof. dr. Ime Priimek
Somentor: doc. dr. Ime Priimek

Ljubljana, 2016

Kazalo

Obrnljive funkcije so sfere v svetu

Povzetek

V povzetku na kratko opišemo vsebinske rezultate dela. Sem ne sodi razlaga organizacije dela – v katerem poglavju/razdelku je kaj, pač pa le opis vsebine.

Invertible maps are spheres in the universe

Abstract

Prevod slovenskega povzetka v angleščino.

Math. Subj. Class. (2020): 74B05, 65N99

Ključne besede: naravni logaritem, nenaravni algoritem

Keywords: natural logarithm, unnatural algorithm

1 Osnove Martin-Löfove odvisne teorije tipov

Sledeč (TODO cite intro to hott) bomo v tem poglavju predstavili osnovne koncepte Martin-Löfove odvisne teorije tipov. Zaradi velike razsežnosti in dokajšnje tehničnosti formalne predstavitve teorije tipov se ne bomo do potankosti spustili v njene podrobnosti, vendar bomo koncepte predstavili v nekoliko neformalnem slogu, osredotočajoč se na analogije z bolj standardnimi matematičnimi koncepti iz formalizma teorije množic.

1.1 Sodbe in pravila sklepanja

Osnovni gradniki teorije tipov so *tipi*, ki jih bomo označevali z velikimi tiskanimi črkami, kot so A, B in C, ter njihovi *elementi*, ki jih bomo označevali z malimi tiskanimi črkami, kot so x, y, z in a, b, c. Vsak element x ima natanko določen tip A, kar izrazimo s tako imenovano *sodbo* $x : A$, ki jo preberemo kot "x je tipa A". Ta je v mnogih pogledih podobna relaciji $x \in A$ iz teorije množic, vendar se od nje razlikuje v nekaj pomembnih pogledih.

Prvič, v teoriji množic je relacija $\in$ v določenem smislu *globalna*; vsak matematični objekt je kodiran kot določena množica in za poljubni množici A in B se lahko vprašamo, ali velja $A \in B$. Tako je na primer vprašanje, ali za element g grupe $\mathbb{Z}/_5$ velja $g \in \mathcal{C}^1(\mathbb{R})$ povsem smiselno, vendar tudi povsem v nasprotju z matematično prakso. V nasprotju s tem so v teoriji tipov elementi nerazdružljivi od svojih tipov; če velja sodba $x : A$, potem vprašanja "Ali velja $x : B$?" sploh ne moremo tvoriti.

Drugič, sodba $x : A$ ni trditev, ki bi jo lahko dokazali, temveč je konstrukcija, do katere lahko pridemo po nekem končnem zaporedju *pravil sklepanja*. To so sintaktična pravila, ki nam omogočijo, da iz določenega nabora sodb tvorimo nove sodbe. Primer pravila sklepanja bi bilo pravilo, ki pravi, da lahko iz sodb $x : A$ in $f : A \to B$ sklepamo, da velja sodba $f(x) : B$, kar predstavlja uporabo funkcije f na elementu x.

Poleg sodb oblike $x : A$ v teoriji tipov obstajajo še tri vrste sodb. Prvič, obstaja sodba A type, s katero izrazimo, da je tip A dobro tvorjen. Primer pravila sklepanja s to sodbo bi bilo pravilo, ki pravi, da lahko iz sodb A type in B type sklepamo, da velja sodba $A \to B$ type, kar predstavlja tvoritev tipa funkcij med tipoma A in B. Drugič, obstajata še sodbi $x \doteq y : A$ in $A \doteq B$ type, s katerima izražamo *sodbeno enakost*. Za obe vrsti sodbene enakosti veljajo pravila sklepanja, preko katerih postaneta ekvivalenčni relaciji, veljajo pa še določena substitucijska pravila, ki omogočajo, da lahko v poljubnem izrazu elemente in tipe nadomeščamo s sodbeno enakimi elementi in tipi. Ker bomo kasneje spoznali še drugo vrsto enakosti, je vredno poudariti, da je sodbena enakost zelo stroga oblika enakosti. Sodbeno enake izraze lahko zato razumemo kot različne zapise za *isti* element.

Pomemben element teorije tipov, ki smo ga do sedaj izpuščali, je dejstvo, da vsaka sodba nastopa v določenem *konstekstu*. Konteksti so končni seznami deklariranih spremenljivk $x_k : A_k$, ki jih sodbe v tem kontekstu lahko vsebujejo, razumemo pa jih lahko tudi kot nabor predpostavk, pod katerimi določena sodba velja. To, da je $\mathcal{J}$ sodba v kontekstu Γ, zapišemo kot $\Gamma \vdash \mathcal{J}$. Primer pravila sklepanja s sodbo v konstekstu bi bilo pravilo, ki pravi, da lahko iz sodbe $x : A \vdash b : B$ sklepamo, da velja $\emptyset \vdash \lambda x.b(x) : A \to B$, kar predstavlja tvoritev anonimne funkcije z vezavo spremenljivke x.

Razpravo povzamemo v sledeči vzajemno rekurzivni definiciji sodb in kontekstov.

Definicija 1.1. V Martin-Löfovi odvisni teoriji tipov obstajajo štiri vrste sodb:

1. A je tip v kontekstu Γ, kar izrazimo z
$$\Gamma \vdash A \ \mathsf{type}.$$
2. A in B sta *sodbeno enaka tipa* tipa v kontekstu Γ, kar izrazimo z
$$\Gamma \vdash A \doteq B \ \mathsf{type}.$$
3. x je element tipa A v kontekstu Γ, kar izrazimo z
$$\Gamma \vdash x : A.$$
4. x in y sta *sodbeno enaka elementa* v kontekstu Γ, kar izrazimo z
$$\Gamma \vdash x \doteq y : A.$$

Kontekst je induktivno definiran seznam sodb, podan s praviloma:

1. Obstaja prazen kontekst $\emptyset$.
2. Če je Γ kontekst in lahko prek obstoječih pravil sklepanja izpeljemo sodbo $\Gamma \vdash A \ \mathsf{type}$, je tedaj $\Gamma, x : A$ kontekst.

Ekspliciten zapis kontekstov in sodb bomo od sedaj deloma izpuščali in o njih govorili v bolj naravnem jeziku. Ko bomo torej trditve začeli stavki kot je "Naj bo A tip in x element A...", smo pravzaprav predpostavili, da velja sodba $A \ \mathsf{type}$ in da delamo v kontestku $\Gamma = x : A$.

S temi definicijami lahko za konec tega podpoglavja predstavimo še specifikaciji funkcijskih in produktnih tipov. Velik del pravil sklepanja za funkcijske tipe smo že videli na primerih, tu pa jih še dopolnimo in zberemo v definicijo.

Definicija 1.2. Naj bosta A in B tipa. Tedaj lahko tvorimo *funkcijski tip* $A \to B$, za katerega veljajo sledeča pravila sklepanja:

1. Če veljata $x : A$ in $f : A \to B$, velja $f(x) : B$.
2. Če pod predpostavko $x : A$ velja $b(x) : B$, velja $\lambda x.b(x) : A \to B$.
3. Za vsak $a : A$ velja enakost $(\lambda x.b(x))(a) \doteq b(a)$.
4. Za vsak $f : A \to B$ velja enakost $\lambda x.f(x) \doteq f$.

Elemente funkcijskega tipa imenujemo *funkcije*.

Definicija 1.3. Naj bosta A in B tipa. Tedaj lahko tvorimo *produktni tip* $A \times B$, za katerega veljajo sledeča pravila sklepanja.

1. Če velja $t : A \times B$, veljata $\mathsf{pr}_1(t) : A$ in $\mathsf{pr}_2(t) : B$.
2. Če veljata $x : A$ in $y : B$, velja $(x, y) : A \times B$.
3. Za vsaka $x : A$ $y : B$ veljata enakosti $\mathsf{pr}_1(a, b) \doteq a$ in $\mathsf{pr}_2(a, b) \doteq b$.
4. Za vsak $t : A \times B$ velja enakost $(\mathsf{pr}_1(t), \mathsf{pr}_2(t)) \doteq t$.

Elemente produktnega tipa imenujemo *pari*.

1.2 Odvisne vsote in odvisni produkti

Pomemben element *odvisne* teorije tipov je dejstvo, da lahko sodbe A type tvorimo v poljubnem kontekstu, to pa pomeni da lahko izrazimo tudi sodbe oblike $x : A \vdash B(x)$ type. V tem primeru pravimo, da je B *odvisen tip*, oziroma, da je B *družina tipov nad A*. Odvisni tipi nam bodo omogočili, da izrazimo tipe, parametrizirane s spremenljivko nekega drugega tipa.

Za delo z odvisnimi tipi vpeljemo še dve pomembni konstrukciji. Prvič, definiramo *odvisni produkt tipov A in B*, elementi katerega so predpisi, ki vsakemu elementu $x : A$ priredijo element v pripadajočem tipu $B(x)$. Če na B gledamo kot družino tipov nad A, lahko na elemente odvisnega produkta gledamo kot na *prereze družine B*, pogosto pa jih imenujemo tudi *odvisne funkcije* ali *družine elementov B, parametrizirane z A*. Drugič, definiramo *odvisno vsoto tipov A in B*, elementi katere so pari (x, y), kjer je $x : A$ in $y : B(x)$. Z odvisno vsoto tako zberemo celotno družino tipov B v skupen tip, na pare (x, y) pa lahko gledamo kot na elemente $y : B(x)$, označene z elementom $x : A$, nad katerim ležijo. Drugače pogledano lahko na pare (x, y) gledamo tudi kot na elemente $x : A$, opremljene z dodatno strukturo y iz pripadajočega tipa $B(x)$.

Definicija 1.4. Naj bo A tip in B družina tipov nad A. Tedaj lahko tvorimo tip

$$\prod (x : A)\, B(x),$$

imenovan *odvisni produkt tipov A in B*. Zanj veljajo sledeča pravila sklepanja:

1. Če veljata $x : A$ in $f : \prod (x : A)\, B(x)$, velja $f(x) : B(x)$.
2. Če pod predpostavko $x : A$ velja $b(x) : B(x)$, velja $\lambda x.b(x) : \prod (x : A)\, B(x)$.
3. Za vsak $a : A$ velja enakost $(\lambda x.b(x))(a) \doteq b(a)$.
4. Za vsak $f : \prod (x : A)\, B(x)$ velja enakost $\lambda x.f(x) \doteq f$.

Definicija 1.5. Naj bo A tip in B družina tipov nad A. Tedaj lahko tvorimo tip

$$\sum (x : A)\, B(x),$$

imenovan *odvisna vsota tipov A in B*. Zanj veljajo sledeča pravila sklepanja

1. Če velja $t : \sum (x : A)\, B(x)$, veljata $\mathsf{pr}_1(t) : A$ in $\mathsf{pr}_2(t) : B(\mathsf{pr}_1(t))$.
2. Če veljata $x : A$ in $y : B(x)$, velja $(x, y) : \sum (x : A)\, B(x)$.
3. Za vsaka $x : A$ $y : B(x)$ veljata enakosti $\mathsf{pr}_1(a, b) \doteq a$ in $\mathsf{pr}_2(a, b) \doteq b$.
4. Za vsak $t : \sum (x : A)\, B(x)$ velja enakost $(\mathsf{pr}_1(t), \mathsf{pr}_2(t)) \doteq t$.

Opomba 1.6. Opazimo lahko podobnost med pravili sklepanja za odvisne produkte in funkcije ter za odvisne vsote in produkte. Ta podobnost ni naključje, saj če je A tip in B tip, neodvisen od A, lahko nanj kljub temu gledamo kot na tip, *trivialno* odvisen od A. V tem primeru veljata enakosti $\sum (x : A)\, B \doteq A \times B$ in $\prod (x : A)\, B \doteq A \to B$.

Primer 1.7. S pomočjo odvisnih produktov lahko podamo specifikacijo tipa *naravnih števil* $\mathbb{N}$, za katera veljajo sledeča pravila sklepanja.

1. Obstaja naravno število $\mathsf{0} : \mathbb{N}$.
2. Če velja $n : \mathbb{N}$, velja $\mathsf{S}(n) : \mathbb{N}$.
3. Velja sledeč *princip indukcije*. Naj bo B družina tipov nad $\mathbb{N}$ in denimo, da veljata $b_{\mathsf{0}} : B(\mathsf{0})$ in $b_{\mathsf{S}} : \prod\,(n : \mathbb{N})\,(B(n) \to B(\mathsf{S}(n)))$.

   Tedaj velja $\mathsf{ind}_{\mathbb{N}}(b_{\mathsf{0}}, b_{\mathsf{S}}) : \prod\,(n : \mathbb{N})\,B(n)$.
4. Veljata enakosti $\mathsf{ind}(b_{\mathsf{0}}, b_{\mathsf{S}}, \mathsf{0}) \doteq b_{\mathsf{0}}$ in $\mathsf{ind}(b_{\mathsf{0}}, b_{\mathsf{S}}, \mathsf{S}(n)) \doteq b_{\mathsf{S}}(n, \mathsf{ind}(b_{\mathsf{0}}, b_{\mathsf{S}}, n))$.

Princip indukcije nam pove, da če želimo za vsako naravno število $n : \mathbb{N}$ konstruirati element tipa $B(n)$, tedaj zadošča, da konstruiramo element tipa $B(\mathsf{0})$ in pa da znamo za vsak $n : \mathbb{N}$ iz elementov tipa $B(n)$ konstruirati elemente tipa $B(\mathsf{S}(n))$, v čemer lahko prepoznamo podobnost z običajnim principom indukcije v teoriji množic. $\diamondsuit$

1.3 Interpretacija *izjave kot tipi*

S tipi ne želimo predstaviti samo matematičnih objektov, temveč tudi matematične trditve, kar bomo dosegli preko tako imenovane interpretacije *izjave kot tipi* (ang. *propositions as types*). V tej interpretaciji vsaki trditvi P dodelimo tip z istim imenom, elementi katerega so dokazi oz. *priče*, da ta trditev velja. Konstrukcija elementa $p : P$ tako postane konstrukcija priče p, ki potrjuje veljavnost trditve P, to pa ustreza dokazu trditve P.

Če si sedaj iz vidika izjav kot tipov zopet ogledamo pravila sklepanja za produktne in funkcijske tipe, lahko v njih prepoznamo pravila sklepanja za konjunkcijo in implikacijo. Dokaz trditve $P \wedge Q$ je namreč sestavljen iz para dokazov posameznih trditev P in Q, dokaz implikacije $P \Rightarrow Q$ pa lahko razumemo kot predpis, ki iz dokazov P konstruira dokaze Q. Uporaba funkcije $P \to Q$ tako ustreza pravilu *modus ponens*, ki pravi, da lahko iz veljavnosti izjav P in $P \Rightarrow Q$ sklepamo veljavnost izjave Q.

Denimo sedaj, da je A tip in P družina tipov nad A, vsak od katerih predstavlja določeno izjavo. Tako družino izjav imenujemo tudi *predikat na* A. Tedaj lahko logično interpretacijo podamo tudi odvisni vsoti in odvisnemu produktu; ustrezata namreč eksistenčni in univerzalni kvantifikaciji. Elementi odvisnega vsote $\sum\,(x : A)\,P(x)$ so pari (x, p), kjer je p dokaz $P(x)$, zato lahko nanje gledamo kot na priče veljavnosti $\exists x : A . P(x)$, elementi odvisnega produkta $\prod\,(x : A)\,P(x)$ pa so predpisi, ki vsakemu elementu $x : A$ priredijo dokaz izjave $P(x)$, zato lahko nanje gledamo kot na priče veljavnosti $\forall x : A . P(x)$.

Definicije predikatov in formulacije trditev, ki jih bomo predstavili v preostanku dela, bomo izrazili v jeziku interpretacije *izjave kot tipi*, kljub temu pa se lahko eksplicitnemu zapisu prič pogosto izognemo. V dokazih trditev bomo tako pogosto uporabljali naravni jezik sklepanja, vedno pa pravzaprav konstruiramo element določenega tipa, ki predstavlja trditev, ki jo dokazujemo.

2 Sintetična homotopska teorija

2.1 Identifikacije in homotopije

Osnovna motivacija za vpeljavo tipa identifikacij je dejstvo, da je pojem sodbene enakosti *prestrog* in da prek nje pogosto ne moremo identificirati vseh izrazov, ki bi jih želeli. Poleg tega si v skladu z interpretacijo *izjave kot tipi* želimo tip, elementi katerega bi predstavljali dokaze enakosti med dvema elementoma.

Pomankljivost sodbene enakosti si lahko ogledamo na primeru komutativnosti naravnih števil.

Primer 2.1. Seštevanje naravnih števil bomo definirali kot funkcijo

$$\mathsf{sum} : \mathbb{N} \to (\mathbb{N} \to \mathbb{N}),$$

njeno evaluacijo $\mathsf{sum}(m, n)$ pa bomo kot običajno pisali kot $m + n$. Ker je tip $\mathbb{N} \to (\mathbb{N} \to \mathbb{N})$ po opombi 1.6 sodbeno enak tipu $\prod (n : \mathbb{N})\, \mathbb{N} \to \mathbb{N}$, lahko njegove elemente konstruiramo po indukcijskem principu naravnih števil. Izpuščajoč nekaj podrobnosti to pomeni, da želimo vrednost izraza $m + n$ definirati z indukcijo na n, torej moramo v baznem primeru podati vrednost izraza $m + \mathsf{0}$, v primeru koraka pa moramo pod predpostavko, da poznamo vrednost izraza $m + n$, podati vrednost izraza $m + \mathsf{S}(n)$. Naravno je, da izberemo $m + \mathsf{0} :\doteq m$ in $m + \mathsf{S}(n) :\doteq \mathsf{S}(m + n)$. Za poljubni *fiksni* naravni števili, recimo $\mathsf{4}$ in $\mathsf{5}$, lahko sedaj z uporabo pravil sklepanja za naravna števila pod točko 4 dokažemo, da velja $\mathsf{4} + \mathsf{5} \doteq \mathsf{5} + \mathsf{4}$. Če pa sta po drugi strani m in n *spremenljivki* tipa $\mathbb{N}$, tedaj zaradi asimetričnosti definicije seštevanja enakosti $n + m \doteq m + n$ ne moremo dokazati. Potrebujemo tip *identifikacij* $\mathsf{Id}_{\mathbb{N}}$, z uporabo katerega bi lahko z indukcijo pokazali, da velja

$$\prod (m, n : \mathbb{N})\, \mathsf{Id}_{\mathbb{N}}(n + m, m + n).$$

$\diamondsuit$

Definicija 2.2. Naj bo A tip in $a, x : A$. Tedaj lahko tvorimo tip $\mathsf{Id}_A(a, x)$, imenovan *tip identifikacij med elementoma a in x*. Zanj veljajo sledeča pravila sklepanja:

1. Za vsak element $a : A$ obstaja identifikacija $\mathsf{refl}_a : \mathsf{Id}_A(a, a)$.

2. Velja sledeč princip *eliminacije identifikacij*. Naj bo $a : A$, B družina tipov, indeksirana z $x : A$ ter $p : \mathsf{Id}_A(a, x)$ in denimo, da velja $r(a) : B(a, \mathsf{refl}_a)$. Tedaj velja
$$\mathsf{ind}_{\mathsf{Id}_A}(a, r(a)) : \prod (x : A) \prod (p : \mathsf{Id}_A(a, x))\, B(x, p)$$

3. Za vsak $a : A$ velja enakost $\mathsf{ind}_{\mathsf{Id}_A}(a, r(a))(a, \mathsf{refl}_a) \doteq r(a)$.

Tip identifikacij $\mathsf{Id}_A(a, x)$ bomo od sedaj pisali kar kot $a = x$ in za boljšo berljivost izpuščali ekspliciten zapis tipa A. Kadar je p identifikacija med a in x, bomo torej pisali $p : a = x$.

Konstruktor $\mathsf{refl}_a : a = a$ nam zagotavlja, da za vsak element $a : A$ obstaja kanonična identifikacija elementa samega s sabo. Princip eliminacije identifikacij nato zatrdi, da če želimo za vsako identifikacijo $p : a = x$ konstruirati element tipa $B(x, p)$, tedaj zadošča, da konstruiramo le element tipa $B(a, \mathsf{refl}_a)$. Pravimo tudi, da je družina tipov $a = x$ *induktivno definirana* s konstruktorjem refl_a. .

Resnična motivacija za takšno definicijo tipa identifikacij sega pregloboko, da bi se vanjo spustili v tem delu, povemo pa lahko, da lahko preko nje dokažemo vse željene lastnosti, ki bi jih od identifikacij pričakovali. Nekaj od teh vključuje:

1. Velja *simetričnost*: za vsaka elementa $x, y : A$ obstaja funkcija

   $$\mathsf{sym} : (x = y) \to (y = x).$$

   Identifikacijo $\mathsf{sym}(p)$ imenujemo *inverz* identifikacije p in ga pogosto označimo z p^{-1}.

2. Velja *tranzitivnost*: za vsake elemente $x, y, z : A$ obstaja funkcija

   $$\mathsf{concat} : (x = y) \to ((y = z) \to (x = z)),$$

   ki jo imenujemo *konkatenacija identifikacij*, $\mathsf{concat}(p, q)$ pa pogosto označimo z $p \cdot q$.

3. Funkcije ohranjajo identifikacije: za vsaka elementa $x, y : A$ in vsako funkcijo $f : A \to B$ obstaja funkcija

   $$\mathsf{ap}_f : (x = y) \to (f(x) = f(y)),$$

   ki jo imenujemo *aplikacija* funkcije f na identifikacije v A.

4. Družine tipov spoštujejo identifikacije: za vsaka elementa $x, y : A$, identifikacijo $p : x = y$ in družino tipov B nad A obstaja funkcija

   $$\mathsf{tr}_B(p) : B(x) \to B(y).$$

   To pomeni, da lahko elemente tipa $B(x)$ *transportiramo* vzdolž identifikacije $p : x = y$, da dobimo elemente $B(y)$. V interpretaciji *izjave kot tipi* to pomeni, da če je P predikat na A in velja identifikacija $x = y$, da tedaj $P(x)$ velja natanko tedaj kot $P(y)$, saj lahko transport uporabimo tudi na p^{-1}.

Konstrukcije naštetih funkcij lahko poiščemo v 5. poglavju (TODO cite intro to hott), vse izmed njih pa zahtevajo le preprosto uporabo eliminacije identifikacij. (TODO grupoidala struktura tipov, interpretacija identifikacij kot poti)

Izkaže se, da smo s tako definicijo identifikacij zelo omejeni v konstruiranju identifikacij med funkcijami. Namesto tega lahko identifikacije med funkcijami nadomestimo s t.i. *homotopijami*, ki v mnogih primerih zadoščajo, konstruiramo pa jih veliko lažje. Homotopija med funkcijami zatrdi, da se ti ujemata na vsakem elementu domene.

Definicija 2.3. Naj bosta A in B tipa ter f in g funkciji tipa $A \to B$. Definiramo tip

$$f \sim g := \prod (x : A)\, f(x) = g(x),$$

imenovan *tip homotopij med f in g*, njegove elemente pa imenujemo *homotopije*.

Opomba 2.4. Na kratko komentiramo o podobnosti med homotopijami v teoriji tipov in homotopijami v klasični topologiji. Naj bosta X in Y topološka prostora in $f, g : X \to Y$ zvezni funkciji. Z I označimo interval $[0, 1]$. Homotopijo med funkcijama f in g tedaj definiramo kot zvezno funkcijo $H : I \times X \to Y$, za katero velja $H(0, x) = f(x)$ in $H(1, x) = g(x)$. Ker pa je I lokalno kompakten prostor, so zvezne funkcije $H : I \times X \to Y$ v bijektivni korespondenci z zveznimi funkcijami $\hat{H} : X \to \mathcal{C}(I, Y)$, kjer je $\mathcal{C}(I, Y)$ prostor zveznih funkcij med I in Y, opremljen s kompaktno-odprto topologijo. Zvezne funkcije $\alpha : I \to Y$ ustrezajo potem v Y med točkama $\alpha(0)$ in $\alpha(1)$. Če je torej H homotopija med f in g, za pripadajočo funkcijo $\hat{H}$ velja, da je $\hat{H}(x)$ pot med točkama $f(x)$ in $g(x)$ za vsak $x \in X$. To je analogno definiciji homotopij med funkcijama f in g v teoriji tipov, ki vsaki točki $x : X$ priredijo identifikacijo $f(x) = g(x)$.

Operacije na identifikacijah lahko po elementih razširimo do operacij na homotopijah, za delo s homotopijami pa bomo potrebovali še operaciji (TODO whiskering?), s katerima lahko homotopije z leve ali z desne razširimo s funkcijo.

Trditev 2.5. *Naj bosta A in B tipa in $f, g, h : A \to B$ funkcije. Denimo, da obstajata homotopiji $H : f \sim g$ in $K : g \sim h$.*

Tedaj obstajata homotopiji $H^{-1} : g \sim f$ in $H \cdot K : f \sim h$.

Dokaz. Če homotopiji H in K evaluiramo na elementu $x : A$, dobimo identifikaciji $H(x) : f(x) = g(x)$ in $K(x) : g(x) = h(x)$. Homotopijo H^{-1} lahko torej definiramo kot $\lambda x.H(x)^{-1}$, homotopijo $H \cdot K$ pa kot $\lambda x.(H(x) \cdot K(x))$. □

Trditev 2.6. *Naj bodo A, B, C in D tipi ter $h : A \to B$, $f, g : B \to C$ in $k : C \to D$ funkcije. Denimo, da obstaja homotopija $H : f \sim g$. Tedaj obstajata homotopiji $Hh : f \circ h \sim g \circ h$ in $kH : k \circ f \sim k \circ g$.*

Dokaz. Homotopijo H zopet evaluiramo na elementu $y : B$ in dobimo identifikacijo $H(y) : f(y) = g(y)$. Če sedaj nanjo apliciramo funkcijo h, dobimo identifikacijo $\mathsf{ap}_h(H(y)) : h(f(y)) = h(g(y))$. Homotopijo kH lahko torej definiramo kot $\lambda y.\mathsf{ap}_k(H(y))$.

Naj bo sedaj $x : A$. Homotopijo H lahko tedaj evaluiramo tudi na elementu $h(x)$ in tako dobimo identifikacijo $H(h(x)) : f(h(x)) = g(h(x))$. Homotopijo Hh lahko torej definiramo kot $\lambda x.H(h(x))$. □

2.2 Obrnljivost in ekvivalenca

Definicija 2.7. Za funkcijo f pravimo, da je *obrnljiva* oziroma, da *ima inverz*, če obstaja element tipa

$$\mathsf{is\text{-}invertible}(f) := \sum (g : B \to A)\,(f \circ g \sim id) \times (g \circ f \sim id),$$

imenovanega *tip inverzov funkcije f*.

Funkcije, pripadajoče elementom $\mathsf{is\text{-}invertible}(f)$, imenujemo *inverzi funkcije f*.

Definicija 2.8. Za funkcijo f pravimo, da *ima prerez*, če obstaja element tipa

$$\mathsf{section}(f) := \sum (g : B \to A)\, f \circ g \sim id,$$

imenovanega *tip prerezov funkcije f.*

Funkcije, pripadajoče elementom tipa $\mathsf{section} f$ imenujemo *prerezi funkcije f.* Za funkcijo f pravimo, da *ima retrakcijo*, če obstaja element tipa

$$\mathsf{retraction}(f) := \sum (g : B \to A)\, g \circ f \sim id,$$

imenovanega *tip retrakcij funkcije f.*

Funkcije, pripadajoče elementom tipa $\mathsf{retraction} f$ imenujemo *retrakcije funkcije f.*

Definicija 2.9. Pravimo, da je funkcija f *ekvivalenca*, če ima tako prerez kot retrakcijo, torej, če obstaja element tipa

$$\mathsf{is\text{-}equiv}(f) := \mathsf{section}(f) \times \mathsf{retraction}(f).$$

Pravimo, da je tip A *ekvivalenten* tipu B, če obstaja ekvivalenca med njima, torej element tipa $A \simeq B := \sum (f : A \to B)\, \mathsf{is\text{-}equiv}(f)$. Funkcijo, pripadajočo elementu $e : A \simeq B$, označimo z $\mathsf{map} e$.

2.3 Osnovne lastnosti

V definiciji obrnljivosti smo zahtevali, da ima funkcija obojestranski inverz, v definiciji ekvivalence pa smo zahtevali le, da ima ločen levi in desni inverz. To bi nas lahko napeljalo k prepričanju, da je pojem obrnljivosti močnejši od pojma ekvivalence, vendar spodnja trditev pokaže da sta pravzaprav logično ekvivalentna. Pokazali bomo, da lahko prerez (ali simetrično, retrakcijo) ekvivalence f vedno izboljšamo do inverza, kar pokaže, da je f tudi obrnljiva.

Trditev 2.10. *Funkcija je ekvivalenca natanko tedaj, ko je obrnljiva.*

Dokaz. Denimo, da je funkcija f obrnljiva. Tedaj lahko njen inverz podamo tako kot njen prerez, kot njeno retrakcijo, kar pokaže, da je ekvivalenca.

Obratno denimo, da je funkcija f ekvivalenca. Podan imamo njen prerez s s homotopijo $H : f \circ s \sim id$ in njeno retrakcijo r s homotopijo $K : r \circ f \sim id$, s katerimi lahko konstruiramo homotopijo tipa $s \circ f \sim id$ po sledečem izračunu:

$$sf \overset{K^{-1}sf}{\sim} rfsf \overset{rHf}{\sim} rf \overset{K}{\sim} id. \qquad \square$$

Definicija 2.11. Naj bo $f : \prod (x : A)\, (Bx \to Cx)$ družina funkcij. Definiramo funkcijo $\mathsf{tot}(f) : \sum (x : A)\, Bx \to \sum (x : A)\, Cx$ s predpisom $\mathsf{tot}(f)(x, y) = (x, f(x)(y))$.

Trditev 2.12. *Naj bo $f : \prod (x : A)\, (Bx \to Cx)$ družina funkcij in denimo, da je $f(x)$ ekvivalenca za vsak $x : A$. Tedaj je ekvivalenca tudi funkcija $\mathsf{tot}(f)$.*

Dokaz. Ker je funkcija $f(x)$ ekvivalenca za vsak $x : A$, lahko tvorimo družino funkcij $s : \prod (x : A)\, (Cx \to Bx)$, kjer je $s(x)$ prerez funkcije $f(x)$. Trdimo, da je tedaj $\mathsf{tot}(s)$ prerez funkcije $\mathsf{tot}(f)$, saj za vsak $(x, y) : \sum (x : A)\, Cx$ velja enakost

$$\mathsf{tot}(f)(\mathsf{tot}(s)(x, y)) = \mathsf{tot}(f)(x, s(x)(y)) = (x, f(x)(s(x)(y))) = (x, y).$$

Analogno lahko konstruiramo retrakcijo funkcije $\mathsf{tot}(f)$, kar zaključi dokaz. $\square$

Zgornja trditev je pomembna, saj nam omogoči, da konstrukcijo ekvivalence med odvisnima vsotama z istim baznim tipom poenostavimo na konstrukcijo družine ekvivalenc, kar je pogosto veliko lažje. Uporabljali ga bomo v obliki sledeče posledice:

Posledica 2.13. *Naj bo A tip, B in C družini tipov nad A in denimo, da velja $Bx \simeq Cx$ za vsak $x : A$. Tedaj velja $\sum (x : A)\, Bx \simeq \sum (x : A)\, Cx$.*

2.4 Podtipi

Definicija 2.14. Naj bo A tip in P družina tipov nad A. Pravimo, da je P *predikat*, če velja $\mathsf{is\text{-}prop}(Px)$ za vsak $x : A$.

Trditev 2.15. *Naj bo A tip, P predikat na A, B pa družina tipov nad A. Denimo, da obstaja družina funkcij $s : \prod (x : A)\, (Bx \to Px)$. Tedaj velja ekvivalenca*

$$\sum (x : A)\, Bx \simeq \sum \Big(t : \sum (x : A)\, Px\Big)\, B(pr_1 t).$$

p

Dokaz. Po asociativnosti odvisne vsote je desna stran ekvivalence ekvivalentna tipu $\sum (x : A) \sum (p : Px)\, Bx = \sum (x : A)\, (Px \times Bx)$. Po posledici 2.13 torej zadošča pokazati, da za vsak $x : A$ obstaja ekvivalenca $Bx \simeq Px \times Bx$.

Funkcijo $f : Bx \to Px \times Bx$ definiramo kot $\lambda\, y.\, (s(x, y),\, y)$, za funkcijo $g : Px \times Bx \to Bx$ pa lahko vzamemo drugo projekcijo. Očitno velja enakost $g(f(y)) = y$, ker pa je P predikat, velja tudi enakost $f(g\,(p, y)) = (s(x, y), y) = (p, y)$. □

3 Karakterizacija obrnljivosti

Definicija 3.1. *Prosta zanka* na tipu A je sestavljena iz točke $a : A$ in identifikacije $a = a$. Tip vseh prostih zank na tipu A označimo s

$$\mathsf{free\text{-}loop}(A) := \sum (x : A)\, x = x.$$

Izrek 3.2. *Tip obrnljivih funkcij med A in B je ekvivalenten tipu prostih zank na tipu $A \simeq B$.*

Dokaz. Želimo konstruirati ekvivalenco tipa

$$\sum (f : A \to B)\, \mathsf{is\text{-}invertible}(f) \simeq \sum (e : A \simeq B)\, (e = e).$$

Ker je $\mathsf{is\text{-}equiv}$ predikat in ker po trditvi 2.10 za vsako funkcijo f obstaja funkcija $\mathsf{is\text{-}invertible}(f) \to \mathsf{is\text{-}equiv}(f)$, najprej opazimo, da po trditvi 2.15 velja ekvivalenca

$$\sum (f : A \to B)\, \mathsf{is\text{-}invertible}(f) \simeq \sum (e : A \simeq B)\, \mathsf{is\text{-}invertible}(\mathsf{map} e).$$

Po posledici 2.13 torej zadošča pokazati, da za vsak element $e : A \simeq B$ obstaja ekvivalenca

$$\mathsf{is\text{-}invertible}(\mathsf{map} e) \simeq (e = e).$$

Oglejmo si tip

$$\mathsf{is\text{-}invertible}(\mathsf{map}e) = \sum (g : B \to A)\,(\mathsf{map}e \circ g \sim id) \times (g \circ \mathsf{map}e \sim id).$$

Po asociativnosti odvisne vsote ta ekvivalenten tipu

$$\begin{aligned}&\sum \Big(H : \sum (g : B \to A)\, \mathsf{map}e \circ g \sim id\Big)\,(\mathsf{map}H \circ \mathsf{map}e \sim id) = \\ &\sum (H : \mathsf{section}(\mathsf{map}e))\,(\mathsf{map}H \circ \mathsf{map}e \sim id),\end{aligned}$$

ker pa imajo po trditvi TODO ekvivalence kontraktibilen tip prerezov, po trditvi (TODO kontraktibilen bazni prostor) velja še ekvivalenca

$$\sum (H : \mathsf{section}(\mathsf{map}e))\,(\mathsf{map}H \circ \mathsf{map}e \sim id) \simeq (\mathsf{sec}e \circ \mathsf{map}e \sim id)\,.$$

Sledi, da velja $\mathsf{is\text{-}invertible}(\mathsf{map}e) \simeq (\mathsf{sec}e \circ \mathsf{map}e \sim id)$, dokaz pa zaključimo še z zaporedjem ekvivalenc, ki jih argumentiramo spodaj.

$$\begin{aligned}&(\mathsf{sec}e \circ \mathsf{map}e \sim id) \simeq \\ &(\mathsf{map}e \circ \mathsf{sec}e \circ \mathsf{map}e \sim \mathsf{map}e) \simeq \\ &(\mathsf{map}e \sim \mathsf{map}e) \simeq \\ &(\mathsf{map}e = \mathsf{map}e) \simeq \\ &(e = e)\end{aligned}$$

- Ker je e ekvivalenca, je po trditvi TODO ekvivalenca tudi delovanje $\mathsf{map}e$ na homotopije.
- Funkcija $\mathsf{sec}e$ je prerez funkcije $\mathsf{map}e$.
- TODO funext
- Po trditvi **??** lahko zanko na funkciji $\mathsf{map}e$ dvignemo do zanke na pripadajoči ekvivalenci e.

□

Slovar strokovnih izrazov

universe svet