UNIVERZITET U NOVOM SADU
FAKULTET TEHNIČKIH NAUKA

Trg Dositeja Obradovića 6
21000 Novi Sad, Srbija

DIPLOMSKI - MASTER RAD

O nekim proširenjima lambda računa

Jelena Ivetić

MENTOR:
Prof. dr Silvia Gilezan

Novi Sad, 2008

Sadržaj

Glava 1

Uvod

Iako su pojedina pravila rezonovanja poznata i formalno uobličena još u antička vremena, logika kao matematička disciplina datira iz druge polovine XIX veka kada su objavljeni radovi Fregea i Peircea. Svoj intenzivni razvoj doživljava početkom XX veka, a najznačajniji logičari tog vremena su Gentzen, Hilbert, Russel i posebno Goedel. U tom periodu su nastali formalni logički sistemi dokazivanja - Hilbertov aksiomatski sistem i Gentzenovi sistemi: prirodna dedukcija i sekventni račun.
20-tih godina XX veka, u skladu sa tadašnjom težnjom matematičara ka formalizaciji matematike, počinje razvoj formalnih računa. U tom periodu nastaje kombinatorni račun Schonfinkela i Curryja, a nešto kasnije Church objavljuje λ-račun, čija je verzija sa tipovima postala osnovni formalizam za pisanje programa. Sa razvojem teorijskog računarstva javlja se težnja za konstruisanjem izvesnih proširenja λ-računa. Tako nastaju formalni računi čija sintaksa dopušta izražavanje odredjenih svojstava kao što su npr. eksplicitna supstitucija i uopštena aplikacija.
Prema Curry-Howardovoj korespondenciji, prirodna dedukcija za intuicionističku logiku i λ-račun sa tipovima su uzajamno odgovarajući sistemi, tako da pojednostavljivanje dokaza odgovara izvršavanju programa.
Kako se svaki dokaz napisan u sistemu prirodne dedukcije može prevesti u dokaz u sistemu sekventnog računa, bilo je jasno da se Curry-Howardova korespondencija može proširiti i na intuicionistički sekventni račun i neku modifikovanu verziju λ-računa. Medjutim, budući da se sekventni račun i λ-račun razlikuju po svojoj strukturi, modifikacija je morala biti izvršena i na sintaksnom nivou, a ne samo u vidu promenjenih pravila tipiziranja. Prvi formalni račun koji je u smislu Curry-Howardove korespondencije bio

odgovarajući jednoj restrikciji sekventnog računa je $\bar{\lambda}$-račun, koji je predložio Herbelin 1995. Korespondencija u ovom slučaju postoji izmedju normalnih formi $\bar{\lambda}$-računa i sekventnih dokaza koji ne sadrže pravilo sečenja. Nakon $\bar{\lambda}$-računa, nastali su i drugi formalni računi zasnovani na sekventnom računu, u kojima redukcija odgovara procesu eliminacije sečenja. Jedan od njih je i λ^{Gtz}-račun, koji je konstruisao Espirito Santo 2006. godine.
Da bi neki formalni račun mogao da bude pouzdano implementiran u programski jezik, neophodno je da zadovoljava pojedine osobine koje omogućavaju bezbedno izvršenje programa. Osobina jake normalizacije znači da ako term ima tip onda ima i osobinu jake normalizacije tj. svi termi sa tipovima imaju isključivo konačne redukcije. Ova osobina je kod većine formalnih računa zadovoljena već u osnovnom tipskom sistemu. Medjutim, obrnuti smer tvrdjenja, poznat kao karakterizacija jake normalizacije, zahteva tipski sistem sa presekom. Ova osobina znači da ako term ima osobinu jake normalizacije onda ima i tip, tj. svim termima sa samo konačnim redukcijama se može dodeliti tip. Ukoliko neki formalni račun ima tipski sistem u kome važe obe osobine, to nam daje jasan kriterijum za proveru terminacije nekog programa. Naime, ako možemo da skup jako normalizovanih terma opišemo skupom terma koji imaju tip, onda bismo svakom programu mogli da pridružimo term, i ako taj term ima tip, sigurni smo da se izvršenje programa uvek završava.

Intuicionistička logika

Logički okvir svih formalnih sistema i formalnih računa prezentovanih u ovom radu je intuicionistička logika, koja se prvi put pominje u radovima Brouwera početkom XX veka. Osnovna odlika intuicionizma je da se istinitim smatra samo ono što može biti dokazano. Dok u klasičnoj logici $A \vee B$ tumačimo kao "A je tačno ili B je tačno" u intuicionističkoj je tumačenje istog izraza "postoji dokaz za A ili dokaz za B". Samim tim, klasični zakon isključenja trećeg $A \vee \neg A$ u intuicionističkoj logici ne važi, kao ni zakoni duple negacije $\neg\neg A \Leftrightarrow A$ i čisto implikativni Piercov zakon $(A \Rightarrow B) \Rightarrow A \Rightarrow A$. Zbog svoje konstruktivne prirode, intuicionistička logika je postala osnova konstruktivne matematike i teorijskog računarstva.

Pregled rada

Osnovni cilj ovog rada je da predstavi λ-račun i neka njegova proširenja u kojima su osobine jake normalizacije i karakterizacije jake normalizacije zadovoljene u tipskom sistemu sa presekom. Prikazani su λx-račun, ΛJ-račun i λ^{Gtz}-račun. Posebna pažnja je posvećena λ^{Gtz}-računu preko kog se mogu predstaviti ostala tri predstavljena računa. Analizirane su sličnosti i razlike ovih računa putem posmatranja tipskih sistema i osobina poput karakterizacije jake normalizacije i očuvanja jake normalizacije. Zaključak rada je da je zbog sekventne prirode λ^{Gtz}-računa, koja ga razlikuje od λx-računa i ΛJ-računa, (oni, kao i λ-račun, odgovaraju prirodnoj dedukciji) ovaj račun u nekim aspektima pogodniji za analizu prirode i svojstava formalnih računa.

Rad je podeljen u 5 celina.

Glava 1 je uvodna i daje pregled celokupnog rada.

U Glavi 2 je predstavljen λ-račun. Prvo je predstavljen λ-račun bez tipova, a potom i dva tipska sistema - osnovni tipski sistem i sistem tipova sa presekom. Objašnjena je Curry-Howardova korespondencija, koja izražava povezanost ovog računa sa prirodnom dedukcijom. Navedeni su primeri računanja sa lambda termima i primeri tipiziranja u oba tipska sistema. Takodje, uvedena je terminologija koja će se koristiti u nastavku rada.

Glava 3 sadrži tri poglavlja u kojima su predstavljena tri proširenja λ-računa. U poglavljima 3.1 i 3.2 su predstavljeni λx-račun sa eksplicitnom supstitucijom i ΛJ-račun sa uopštenom aplikacijom. Za oba računa su prikazani sintaksa, pravila redukcije i tipski sistem sa presekom koji karakteriše jaku normalizaciju. Dati su primeri koji pokazuju da u ovim računima nije zadovoljeno očuvanje jake normalizacije λ-računa. Poglavlje 3.3 se bavi λ^{Gtz}-računom. Definisani su sintaksa i pravila redukovanja i naveden je primer mogućnosti simulacije različitih strategija izračunavanja. Potom je uveden osnovni tipski sistem $\lambda^{Gtz}\to$, a zatim i tipski sistem sa presekom $\lambda^{Gtz}\cap$. Navedena su tvrdjenja koja iskazuju osobine ovog računa, uključujući očuvanje tipa pri redukciji, očuvanje tipa pri ekspanziji, jaku normalizaciju i karakterizaciju jake normalizacije.

U Glavi 4 je uspostavljena relacija izmedju prethodno navedenih formalnih računa. U poglavlju 4.1 je pokazano kako se λx-termi i ΛJ-termi mogu prikazati kao podklase λ^{Gtz}-terma, i kako se u skladu sa tim mogu konstruisati jednostavniji tipski sistemi sa presekom pomoću kojih se mogu okarakterisati termi sa osobinom jake normalizacije. Takodje, analizirani su primeri terma iz prethodne glave, sada posmatrani kroz λ^{Gtz}-račun. Poglavlje 4.2 se bavi vezom izmedju λ-računa i λ^{Gtz}-računa. U njemu su i λ-termi prikazani kao podklasa λ^{Gtz}-terma, ali je prikazano i kako se λ^{Gtz}-termi mogu prevesti u λ-račun. Dokazana je osobina očuvanja jake normalizacije i ponudjen je novi tipski sistem za karakterizaciju jako normalizovanih λ-terma. Rezultati poglavlja 3.3 i Glave 4 su plod zajedničkog rada sa Silviom Gilezan, Jose Espirito Santom i Silviom Likavec i nalaze se u radovima [9], [15], [12] [17] i [13].

Glava 5 je zaključna i u njoj su sumirani rezultati ovog rada.

Na kraju rada se nalazi spisak korišćene literature.

Glava 2

λ-račun

λ-račun je formalizam koji je 1932. godine uveo Church, pokušavajući da napravi sistem koji bi formalizovao matematiku. Iako se pokazalo da je formalizacija matematike neizvodljiva, ispostavilo se da ovaj sistem ima drugu, veoma važnu primenu. Naime, Church je 1936. godine pokazao da je λ-račun uspešan model za predstavljanje izračunljivih funkcija. Sa razvojem teorije programskih jezika, λ-račun se našao u osnovi raznih funkcionalnih programskih jezika. Eksplicitno je implementiran u neke programske jezike (LISP, Haskell, Scheme...), a razne varijante i ekstenzije λ-računa su osnova za implementaciju i dizajn mnogih drugih kako funkcionalnih tako i objektno-orijentisanih programskih jezika. U skorašnje vreme, našao je i primenu u novom polju - izgradnji automatskih i poluautomatskih dokazivača teorema (npr. Isabelle, COQ) i programa za proveru dokaza.

2.1 λ-račun bez tipova

Sintaksa λ-računa je jednostavna - izrazi se zovu λ-termi, obeležavaju se slovima t, u, v, t'... i grade se na sledeći način:

$$\text{termi: } t \ ::= \ x \mid (\lambda x.t) \mid (tu)$$

Dakle, osnovni termi su promenljive $(x, y, z, x_1, ...)$ koje se spajaju operatorima *apstrakcije* i *aplikacije*. Intuitivno, operator apstrakcije pravi funkciju od terma t, tako što simbolom λ "vezuje" promenljivu koja postaje argument te funkcije, slično kao u izrazu $\int f \, dx$. Aplikacija je primena terma na term tj. funkcije na funkciju. Primeri λ-terma su sledeći:

$$x,\ \lambda x.(yz),\ \lambda x.(\lambda y.((yx)x)).$$

Radi jednostavnijeg pisanja, koriste se sledeći skraćeni zapisi:

- umesto $\lambda x_1.(\lambda x_2.(...(\lambda x_n.t)...))$, pišemo $\lambda x_1 x_2 ... x_n.t$;

- umesto $(...((t_1 t_2)t_3)...t_n)$ pišemo $t_1 t_2 t_3 ... t_n$.

Dakle, apstrakcije se grupišu na desno, a aplikacije na levo, tako da prethodno navedene terme u stvari pišemo ovako:

$$x,\ \lambda x.yz,\ \lambda xy.yxx.$$

U svakom λ-termu, razlikujemo *vezane* promenljive (one koje su izmedju simbola λ i .) i *slobodne* promenljive.

Definicija 1 *Skup slobodnih promenljivih terma t, u oznaci $Fv(t)$, je induktivno definisan na sledeći način:*

1. $Fv(x) = \{x\}$;
2. $Fv(tu) = Fv(t) \cup Fv(u)$;
3. $Fv(\lambda x.t) = Fv(t) \setminus \{x\}$.

Definicija 2 *λ-term koji nema slobodnih promenljivih se zove* zatvoren *term ili* kombinator*. Skup svih zatvorenih λ-terma se označava sa Λ°.*

Osnovno pravilo računanja sa termima je pravilo β-redukcije:

$$(\lambda x.t)u \ \to_\beta \ t[x := u]$$

β-redukcija u stvari predstavlja supstituciju, jer izraz $t[x := u]$ čitamo kao "u termu t, sva slobodna pojavljivanja promenljive x zameniti sa u". Ovo je meta-supstitucija, zato što ne predstavlja deo sintakse ovog računa, već se posebno definiše na sledeći način:

1. $x[x := u] = u$;
2. $y[x := u] = y$;

3. $(tv)[x := u] = t[x := u]v[x := u]$;
4. $(\lambda x.t)[x := u] = \lambda x.t$;
5. $(\lambda y.t)[x := u] = \lambda y.(t[x := u])$, ako $y \notin Fv(u)$;
6. $(\lambda y.t)[x := u] = (\lambda y'.t[y := y'])[x := u]$, ako $y \in Fv(u)$ a $y' \notin Fv(u)$.

Iako λ-račun sadrži još dve redukcije, upravo mu β-redukcija daje toliku ekspresivnost.

Kada je promenljiva vezana, njeno se ime može menjati a da term ostane isti, u smislu da i dalje predstavlja istu funkciju. Na primer, $\lambda x.xy$ i $\lambda z.zy$ su isti termi, baš kao što su i $f(x) = y + x$ i $f(z) = y + z$ iste funkcije, jer za iste vrednosti argumenta daju isti rezultat. Medjutim, kao što se funkcija $f(y) = y + y$ razlikuje od prethodne dve, tako ni u prethodnom λ-termu ne možemo preimenovanje vršiti tako da slobodna promenljiva postane vezana. Takodje, ako se u jednom termu nadje slobodna promenljiva koja bi posle β-redukcije postala vezana, potrebno je izvršiti preimenovanje jedne od njih. Ovo pravilo je poznato pod imenom *Barendregtova konvencija.*
Pomenuto preimenovanje vezanih promenljivih je formalizovano u vidu α-redukcije:

$$\lambda x.t \ \to_\alpha \ \lambda y.t[x := y], \quad y \notin Fv(t).$$

Preostala redukcija je η-redukcija, koja izražava osobinu da su dve funkcije jednake ako primenjene na isti argument vraćaju istu vrednost tj. da je u tom slučaju jasno o kojoj se funkciji radi i bez eksplicitnog navodjenja argumenta.

$$\lambda x.tx \ \to_\eta \ t, \quad x \notin Fv(t).$$

Dakle, ako posmatramo meta-zapis funkcija iz prethodnog pasusa, termi $\lambda x.\mathbf{add}\ y\ x$ i $\lambda z.\mathbf{add}\ y\ z$ se η-redukuju na term $\mathbf{add}\ y$, i svi oni predstavljaju istu funkciju, ali u slučaju terma $\lambda y.\mathbf{add}\ y\ y$ se η-redukcija ne može primeniti. Formalni razlog je to što je promenljiva y slobodna u termu $\mathbf{add}\ y$, a semantički je jasno da funkcije *"dodaj y"* i *"dodaj broju isti taj broj"* nisu iste.

Zapis $t \ \to \ t'$ označava da se term t nekom od tri redukcije redukuje na term t'. U tom slučaju, t se zove *redeks* a t' se zove *kontraktum.* Ukoliko se term t' dobija od t nakon niza redukcija, to ćemo označavati sa $t \twoheadrightarrow t'$.

Veoma važno svojstvo λ-računa ogleda se u tome da ćemo bez obzira na redosled kojim vršimo redukcije nekog terma, na kraju stići do istog kontraktuma. To znači da je, u skladu sa konkretnim slučajem, moguće primeniti različite *strategije računanja*, kao što su npr. *call-by-name* i *call-by-value*, bez bojazni da će to uticati na promenu rezultata. Ovo svojstvo se zove *konfluencija* i navedeno je u sledećoj teoremi:

Teorema (Church-Rosser) Ako $u \twoheadleftarrow t \twoheadrightarrow v$, onda postoji term t' takav da $u \twoheadrightarrow t' \twoheadleftarrow v$.

Drugo važno svojstvo je svojstvo *normalizacije*. Ako neki term ne može da se redukuje, on se zove *normalna forma*. Skup svih normalnih formi NF u λ-računu se induktivno definiše na sledeći način:

1. $x \in NF$, za svaku promenljivu x;
2. Ako $t_1, t_2, ..., t_n \in NF$, onda i $xt_1t_2...t_n \in NF$;
3. Ako $t \in NF$, onda i $\lambda x.t \in NF$.

Dakle, sve normalne forme su oblika $\lambda y_1...y_n.zN_1...N_k$, gde su termi $N_i,\ i \in \{1, ..., k\}$ takodje normalne forme, a z može biti jedna od promenljivih $y_j,\ j \in \{1, ..., n\}$. Ako se neki term redukuje do normalne forme, kažemo da ima osobinu normalizacije. Ako su sve redukcije nekog terma konačne, kažemo da ima osobinu *jake normalizacije*. Skup svih terma koji imaju ovu osobinu se obeležava sa SN i pripada mu samo mali deo od ukupnog broja svih λ-terma. Zbog primene u programskim jezicima, bilo je veoma važno pronaći način za opisivanje i izolovanje skupa SN, jer bi u protivnom moglo doći do tzv. *Halting* problema, tj. blokiranja programa. Rešenje ovog problema je pronadjeno u λ-računu sa tipovima.

2.2 λ-račun sa tipovima

Teoriju tipova su uveli Curry [1934] u svom radu o kombinatorima i Church [1940] za λ-račun. Njihovi pristupi se donekle razlikuju, jer Curry implicitno dodeljuje tipove, a Church eksplicitno (tako da razlikujemo "*à la Curry*"

i "*à la Church*" dodelu tipova). Neki autori nazivaju Curry-jev sistem "lambda račun sa dodeljivanjem tipova", a Church-ov sistem "lambda račun sa tipovima". Osnovna ideja je kod oba načina ista, tako da ćemo u nastavku primenjivati "*à la Curry*" princip dodele tipova.
Tipovi su, naime, sintaksni objekti, koji se dodeljuju λ-termima. Intuitivno, mogu se posmatrati kao domen funkcije koju λ-term reprezentuje. λ-račun sa tipovima omogućava mnogo dublju i detaljniju analizu programskih jezika od teorije bez tipova, tako da je moguće posmatrati λ-term kao program, a njemu odgovarajući tip kao specifikaciju programa.

Osnovni tipski sistem

Osnovni tipski sistem, u oznaci $\lambda \rightarrow$, je sistem u kom se kao jedini operator javlja strelica $\rightarrow$. U ovom sistemu, tipovi se prave od tipskih promenljivih koje se povezuju operacijom $\rightarrow$.

$$A, B \ ::= \ p \mid A \rightarrow B$$

U cilju smanjenja broja zagrada, usvaja se konvencija da se strelice grupišu prema desno tj. $A \rightarrow B \rightarrow C \ = \ A \rightarrow (B \rightarrow C)$.

Definicija 3

1) Dodela tipa je izraz oblika $t : A$, gde je t λ-term, a A tip, tj. formula.

2) Deklaracija je dodela tipa u kojoj je term promenljiva.

3) Baza $\Gamma = \{x_1 : A_1, ..., x_n : A_n\}$ je skup deklaracija u kojem je svaka promenljiva deklarisana najviše jednom. $dom\Gamma = \{x_1, ..., x_n\}$

Pravila ovog tipskog sistema su predstavljena u Tabeli 1.
U njemu aplikacija lambda terma dovodi do eliminacije strelice, dok je apstrakcija uvodi.

$(axiom)$	$$\frac{(x:A)\in\Gamma}{\Gamma\vdash x:A}$$
$(\rightarrow_{elim})$	$$\frac{\Gamma\vdash t:A\rightarrow B \quad \Gamma\vdash u:A}{\Gamma\vdash tu:B}$$
$(\rightarrow_{intr})$	$$\frac{\Gamma, x:A\vdash t:B}{\Gamma\vdash \lambda x.t:A\rightarrow B}$$

Tabela 1: Tipski sistem $\lambda\rightarrow$

Nakon uvodjenja tipova u lambda račun, postalo je jasno da postoji veza izmedju ovog formalnog računa i prirodne dedukcije (ND) za implikativni fragment intuicionističke logike. Ova veza je poznata pod imenom *Curry-Howardova korespondencija* i može se uočiti na više nivoa.

Kao prvo, svaki tip koji se može dodeliti zatvorenom lambda termu odgovara teoremi u intuicionističkoj implikativnoj logici i suprotno, za svaku takvu teoremu postoji term čiji je ona tip. Ova bijekcija se ponekad označava sintagmom *propositions-as-types*.
Kao drugo, postupak dodele tipa nekom termu odgovara upravo dokazu odgovarajuće teoreme u logičkom sistemu ND. Ukoliko posmatramo terme kao funkcije, postupak tipiziranja se može posmatrati i kao program kojim se odredjuje domen vrednosti koju funkcija izračunava, pa se zbog toga prethodna veza često pominje u formi *proofs are programs*. Ova formulacija je našla jako uporište u računarstvu, u domenu verifikacije programa na osnovu ekstrakcije odgovarajućeg dokaza.
Konačno, veza se može uspostaviti i izmedju postupka normalizacije tj. pojednostavljivanja dokaza u sistemu ND i postupka redukcije lambda terma. Ovu korespondenciju je prvi primetio Curry, 50-tih godina prošlog veka, doduše u malo izmenjenoj formi, razvijajući *teoriju kombinatora* koji na sličan, ali očigledniji način, odgovaraju Hilbertovom aksiomatskom sistemu. Nakon Prawitzovog rada o normalizaciji u prirodnoj dedukciji, Howard je formulisao ovu vezu 1969. godine, ali je rad objavljen tek 1980 godine u zborniku povodom Curryevog osamdesetog rodjendana [25]. Pre toga su ovaj rezultat intenzivno koristili de Bruijn i Lambek, koji je dao interpretaciju u

teoriji kategorija, tako da je najpravedniji naziv Curry-Howard-de Bruijn-Lambekova korespondencija.

Tipski sistem sa presekom

U osnovnom tipskom sistemu važi da ako term ima tip onda ima i osobinu jake normalizacije tj. svi termi sa tipovima imaju isključivo konačne redukcije. Sa aspekta računarstva, ovo je veoma korisna osobina. Medjutim, obrnuti smer tvrdjenja bi bio još korisniji zato što bi dao jasan kriterijum za proveru terminacije nekog programa. Naime, ako bismo mogli da skup jako normalizovanih terma opišemo skupom terma koji imaju tip, onda bismo svakom programu mogli da pridružimo term, i ako taj term ima tip, sigurni smo da se izvršenje programa uvek završava.
Na žalost, osnovni tipski sistem ne zadovoljava ovu osobinu. U njemu ne samo da nemaju tip svi termi sa osobinom jake normalizacije, već postoje i normalne forme koje se ne mogu tipizirati. Primer za to je term $\lambda x.xx$. Da bismo mogli da tipiziramo ovaj term tip prvog x-a u aplikativnom delu bi morao da ima jednu strelicu više od tipa drugog x-a, a u osnovnom tipskom sistemu je nemoguće deklarisati promenljivu sa dva različita tipa.

Upravo ideja da je termu moguće dodeliti više od jednog tipa je dovela do uvodjenja operatora $\cap$ u teoriju tipova. Prvi sistem sa tipovima sa presekom su uveli Coppo i Dezani, 1978. godine, u radu [6]. Sistem je poznat i pod imenom *Torino sistem.*

U ovom tipskom sistemu, tipovi se grade od tipskih promenljivih koje se povezuju operatorima $\to$ i $\cap$.

$$A, B \ ::= \ p \mid A \to B \mid A \cap B$$

Interpretacija operatora $\cap$ odgovara preseku u teoriji skupova. Dakle, ako neki term ima tipove A i B, onda ima i tip $A \cap B$. Još jedno svojstvo ovog sistema je postojanje univerzalnog tipa ω koji može biti dodeljen svakom termu. U cilju uredjenja prostora tipova uvodi se i relacija poretka $\leq$:

Definicija 4 *Relacija poretka $\leq$ definisana nad skupom tipova je najmanja relacija koja zadovoljava sledeće osobine:*

1. $A \leq A$;
2. $A \leq B$ *i* $B \leq C \;\;\Rightarrow\;\; A \leq C$;
3. $A \cap B \leq A, \;\; A \cap B \leq B$;
4. $A \leq B$ *i* $A \leq C \;\;\Rightarrow\;\; A \leq B \cap C$;
5. $(A \to B) \cap (A \to C) \leq A \to (B \cap C)$;
6. $A \leq A_1$ *i* $B \leq B_1 \;\;\Rightarrow\;\; A_1 \to B \leq A \to B_1$.

Definicija 5 *Dva tipa su ekvivalentna, u oznaci $A \sim B$, ako i samo ako je $A \leq B$ i $B \leq A$.*

Aksioma i pravila tipiziranja u tipskom sistemu $\lambda \cap \omega$ su navedeni u Tabeli 2.
Različitim kombinacijama ovih pravila dobijaju se restrikcije sistema $\lambda \cap \omega$. Ovi sistemi su dati sledećim aksiomama i pravilima:

- $\lambda \to$ - *osnovni tipski sistem*: (ax), $(\to E)$, i $(\to I)$;

- $\mathcal{D}$ - *osnovni tipski sistem sa presekom*: (ax), $(\to E)$, $(\to I)$, $(\cap E)$ i $(\cap I)$.

- $\mathcal{D}\Omega$: (ax), $(\to E)$, $(\to I)$, $(\cap E)$, $(\cap I)$ i (ω).

- $\lambda\cap$: (ax), $(\to E)$, $(\to I)$, $(\cap E)$, $(\cap I)$ i $(\leq)$.

Od posebnog značaja su tipski sistemi $\mathcal{D}$ i $\lambda\cap$ zato što upravo u ovim sistemima tipiziranost karakteriše jaku normalizaciju. Ovu osobinu su prvi put dokazali Pottinger u [23] i Coppo i Dezanni u [6]. Kasnije, ova tema je obradjivana u radovima Ghilezan [14] i Krivine [19], dok su kompletan dokaz dali Amadio i Curien u radu [2].

(ax)	$\Gamma, x : A \vdash x : A$
$(\to E)$	$\dfrac{\Gamma \vdash t : A \to B \quad \Gamma \vdash u : A}{\Gamma \vdash tu : B}$
$(\to I)$	$\dfrac{\Gamma, x : A \vdash t : B}{\Gamma \vdash \lambda x.t : A \to B}$
$(\cap E)$	$\dfrac{\Gamma \vdash t : A \cap B}{\Gamma \vdash t : A \quad \Gamma \vdash t : B}$
$(\cap I)$	$\dfrac{\Gamma \vdash t : A \quad \Gamma \vdash t : B}{\Gamma \vdash t : A \cap B}$
$(\leq)$	$\dfrac{\Gamma \vdash t : A \quad A \leq B}{\Gamma \vdash t : B}$
(ω)	$\dfrac{}{\Gamma \vdash t : \omega}$

Tabela 2: Tipski sistem $\lambda \cap \omega$

Ovo poglavlje završavamo primerima tipiziranja u tipskim sistemima $\lambda \to$ i $\mathcal{D}$. U nastavku ćemo koristiti skraćeni zapis $\Gamma \setminus \{x\}$ umesto $\Gamma \setminus \{x : A\}$.

Primer 6

- *Termu* $S \equiv \lambda xyz.xz(yz)$ *se može dodeliti tip* $(A \to (B \to C)) \to ((A \to B) \to (A \to C))$. *Neka je baza* $\Gamma = \{x : A \to (B \to C),\ y : A \to B,\ z : A\}$.

$$
\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\Gamma \vdash x : A \to (B \to C)}\,(ax) \quad \dfrac{}{\Gamma \vdash z : A}\,(ax)}{\Gamma \vdash xz : B \to C}\,(\to E) \quad \dfrac{\dfrac{}{\Gamma \vdash y : A \to B}\,(ax) \quad \dfrac{}{\Gamma \vdash z : A}\,(ax)}{\Gamma \vdash yz : B}\,(\to E)}{\Gamma \vdash xz(yz) : C}\,(\to E)}{\Gamma \setminus \{z\} \vdash \lambda z.xz(yz) : A \to C}\,(\to I)}{\dfrac{\Gamma \setminus \{y,z\} \vdash \lambda yz.xz(yz) : (A \to B) \to (A \to C)}{\vdash \lambda xyz.xz(yz) : (A \to (B \to C)) \to ((A \to B) \to (A \to C)).}\,(\to I)}\,(\to I)
$$

- *Termu $\lambda x.xx$ se u tipskom sistemu $\mathcal{D}$ može dodeliti tip $A \cap (A \to B) \to B$ na sledeći način:*

$$
\dfrac{\dfrac{\dfrac{\dfrac{}{x : A \cap (A \to B) \vdash x : A \cap (A \to B)}\,(ax)}{x : A \cap (A \to B) \vdash x : A \to B \quad x : A \cap (A \to B) \vdash x : A}\,(\cap E)}{x : A \cap (A \to B) \vdash xx : B}\,(\to E)}{\vdash \lambda x.xx : A \cap (A \to B) \to B.}\,(\to I)
$$

Glava 3

Proširenja λ-računa

3.1 λx-račun

λx-račun je račun sa eksplicitnom supstitucijom, tj. modifikacija standardnog λ-računa u kome supstitucija nije meta-operacija nad termima, već predstavlja operaciju ugradjenu u sintaksu jezika. U λ-računu možemo term sa više aplikacija izgraditi na više načina. Na primer, term $\lambda x.yz$ se može izgraditi na sledeća dva načina:

$$\begin{array}{lllll} u & \mapsto & yz & \mapsto & \lambda x.yz, \\ u & \mapsto & \lambda x.u & \mapsto & \lambda x.yz. \end{array}$$

Eksplicitnim uvodjenjem operatora supstitucije, izvodjenja iz prethodnog primera ne generišu više jedan, već dva različita terma: $\lambda x.(u\langle u = yz\rangle)$ i $(\lambda x.u)\langle u = yz\rangle$.

Prve formalne račune sa eks plicitnom supstitucijom su definisali Abadi et al. u radu [1] i de Bruijn [7]. Pokazalo se da svojstvo jake normalizacije nije zadovoljeno čak ni za verziju sa osnovnim tipskim sistemom Abadi et al. računa, i da razlog leži u postojanju kompozicije supstitucija. Da bi se postiglo svojstvo jake normalizacije, λx-račun koji ovde predstavljamo, a koji je detaljno proučen u radu Lengrand et al. [20], nema mogućnost kompozicije supstitucija. U tom smislu on predstavlja modifikaciju prvobitnog λx-računa koji su predložili Bloo i Rose [5].

3.1.1 λx-račun bez tipova

Sintaksa λx-računa je direktno proširenje sintakse λ-računa:

$$\text{termi: } t \ ::= \ x \mid \lambda x.t \mid tu \mid t\langle x = u\rangle$$

Termi mogu biti promenljive, apstrakcije, aplikacije i *zatvaranja*, kako zovemo terme oblika $t\langle x = u\rangle$. Term koji ne sadrži zatvaranje se zove *čist term*. Osim pojma slobodne i vezane promenljive, koji su definisani na uobičajen način, od ključnog značaja u prisustvu eksplicitne supstitucije je pojam *dostupne* promenljive.

Definicija 7 *Skup slobodnih promenljivih terma t, u oznaci $Fv(t)$, je induktivno definisan na sledeći način:*

1. $Fv(x) \ = \ \{x\}$;
2. $Fv(tu) \ = \ Fv(t) \ \cup \ Fv(u)$;
3. $Fv(\lambda x.t) \ = \ Fv(t) \setminus \ \{x\}$.
4. $Fv(t\langle x = u\rangle) \ = \ (Fv(t) \setminus \ \{x\}) \ \cup \ Fv(u)$.

Promenljive koje nisu slobodne u nekom termu se zovu vezane.

λx-računu postoje dva operatora koja vezuju promenljive - operator apstrakcije λ i operator eksplicitne supstitucije $\cdot\langle\cdot = \cdot\rangle$, i Barendregtova konvencija se treba primenjivati u oba slučaja.

Definicija 8 *Skup dostupnih promenljivih terma t, u oznaci $Av(t)$, je induktivno definisan na sledeći način:*

1. $Av(x) \ = \ \{x\}$;
2. $Av(tu) \ = \ Av(t) \ \cup \ Av(u)$;
3. $Av(\lambda x.t) \ = \ Av(t) \setminus \ \{x\}$.
4. $Av(t\langle x = u\rangle) \ = \begin{cases} (Av(t) \setminus \ \{x\}) \ \cup \ Av(u), & \textit{ako } x \in Av(t) \\ Av(t), & \textit{ako } x \notin Av(t) \end{cases}$

Može se pokazati da je skup dostupnih promenljivih podskup skupa slobodnih promenljivih, dok se kod čistih terma ovi skupovi podudaraju. Razliku možemo ilustrovati primerom terma $z\langle y = xx\rangle$, u kom je promenljiva x slobodna, ali ne i dostupna, zato što prilikom supstitucije potpuno nestaje.

U λx-računu postoje sledeća redukciona pravila:

$$
\begin{array}{rrcl}
(B) & (\lambda x.t)u & \to & t\langle x = u\rangle \\
(App) & (tv)\langle x = u\rangle & \to & t\langle x = u\rangle v\langle x = u\rangle \\
(Abs) & (\lambda y.t)\langle x = u\rangle & \to & \lambda y.(t\langle x = u\rangle) \\
(VarI) & x\langle x = u\rangle & \to & u \\
(VarK) & y\langle x = u\rangle & \to & y \\
(gc) & t\langle x = u\rangle & \to & t,\ akox \notin Av(t)
\end{array}
$$

Primetimo da je Barendregtova konvencija o preimenovanju slobodnih promenljivih ponovo od značaja, posebno u pravilu (Abs). Takodje, pravilo (gc), tzv. *garbage collection* je ovde definisano na nov način, budući da koristi pojam dostupne promenljive.

3.1.2 λx-račun sa tipovima sa presekom

U cilju karakterizacije skupa terma sa osobinom jake normalizacije, u λx-račun je uveden tipski sistem sa tipovima sa presekom. Presek je uveden na isti način kao u [6] za λ-račun, i koristi definiciju 4 za relacije poretka i ekvivalencije. Tipski sistem, u oznaci $\mathcal{E}$ je predstavljen u Tabeli 3. Definicije skupa tipova, dodele tipova i baze su iste kao u λ-računu, pa neće biti ponovo navodjene. Osobenost ovog sistema je postojanje 3 pravila za tipiziranje eksplicitne supstitucije: (cut), $(drop)$ i $(K - cut)$. Ova pravila se ne isključuju medjusobno, tj. kada su zadovoljeni odredjeni uslovi, moguće je birati izmedju upotrebe pravila (cut) i jednog od pravila $(drop)$ i $(K - cut)$. Štaviše, može se pokazati da pravila $(drop)$ i $(K - cut)$ ne moraju istovremeno biti u sistemu, tj. da je bez jednog od njih moguće izvesti isti skup tipiziranih terma. Motivaciju za uvodjenje ovih dodatnih pravila nalazimo u sledećem primeru, koji pokazuje da tipskim sistem bez pravila $(drop)$ i $(K - cut)$, koji ćemo označavati sa $\mathcal{E}^-$ ne može da tipizira sve terme sa osobinom jake normalizacije.

$$\frac{x : A \in \Gamma}{\Gamma \vdash x : A}\,(start) \qquad \frac{\Gamma, x : A \vdash t : B}{\Gamma \vdash \lambda x.t : A \to B}\,(\to I)$$

$$\frac{\Gamma, x : A \vdash t : B \quad \Gamma \vdash u : A}{\Gamma \vdash t\langle x = u\rangle : B}\,(cut) \qquad \frac{\Gamma \vdash t : A \to B \quad \Gamma \vdash u : B}{\Gamma \vdash tu : B}\,(\to E)$$

$$\frac{\Gamma, x : A \vdash t : B \quad \Delta \vdash u : A \quad x \notin Av(M)}{\Gamma \vdash t\langle x = u\rangle : B}\,(drop) \qquad \frac{\Gamma \vdash t : A \quad \Gamma \vdash t : B}{\Gamma \vdash t : A \cap B}\,(\cap I)$$

$$\frac{\Gamma, x : A \vdash t : B \quad \Delta \vdash u : A \quad x \notin dom\Gamma}{\Gamma \vdash t\langle x = u\rangle : B}\,(K - cut) \qquad \frac{\Gamma \vdash t : A_1 \cap A_2}{\Gamma \vdash t : A_i, \;\; i \in \{1, 2\}}\,(\cap E)$$

Tabela 3: Tipski sistem $\mathcal{E}$

Primer Posmatrajmo terme:

$$t_1 \equiv ((\lambda y.z)(xx))\langle x = \lambda a.aa\rangle$$

$$t_2 \equiv z\langle y = xx\rangle\langle x = \lambda a.aa\rangle$$

Term t_1 se može redukovati na t_2, ali se može i beskonačno redukovati preko redeksa $((\lambda y.z)((\lambda a.aa)(\lambda a.aa))$. Dakle, t_1 nema osobinu jake normalizacije, pa stoga nema ni tip u tipskom sistemu $\mathcal{E}^-$ (ovaj tipski sistem zadovoljava osobinu jake normalizacije, što je dokazano u [8]. t_2, medjutim, ima osobinu jake normalizacije, ali zbog toga što se dobija β-redukcijom od terma t_1, a β-redukcija ne menja tip termu (Teorema o očuvanju tipa pri redukciji je dokazana za sistem $\mathcal{E}^-$ takodje u [8]) zaključujemo da ni t_2 nema tip u ovom tipskom sistemu.
U tipskom sistemu sa pravilima $(drop)$ i $(K - cut)$ term t_2, kao i svi termi sa osobinom jake normalizacije, ima tip, i to sledi iz sledećih tvrdjenja, koja će biti navedena bez dokaza (dokazi se nalaze u [20]).

Teorema 9 *Ako λx-term ima osobinu jake normalizacije, onda ima tip u tipskom sistemu $\mathcal{E}$.*

Teorema 10 *Ako u tipskom sistemu $\mathcal{E}$ za λx-term važi $\Gamma \vdash t : A$, za neku bazu Γ i neki tip A, onda t pripada skupu terma sa osobinom jake normalizacije.*

3.2 ΛJ-račun

ΛJ-račun su uveli Joachimski i Matthes u radu [18], kao proširenje λ-računa sa *uopštenom aplikacijom*. Motivacija se nalazi u von Plato-vom sistemu prirodne dedukcije, u kom se nalazi logičko pravilo uopštene eliminacije.

3.2.1 ΛJ-račun bez tipova

Sintaksa ΛJ-računa je definisana na sledeći način:

$$\text{termi: } t \ ::= \ x \mid \lambda xt \mid t(u, x.v)$$

Termi mogu biti promenljive, apstrakcije i uopštene aplikacije. Obična aplikacija, i samim tim i λ-račun, se dobijaju restrikcijom uopštenih aplikacija na formu $t(u, x.x)$ koju onda označavamo sa tu. Obrnuto, uopštena aplikacija se može simulirati u okviru λ-računa na sledeći način: $t(u, x.v) := v[x := tu]$. Promenljive u ovom računu vezuju oba operatora, tako da je x vezana operatorom apstrakcije u termu λxr i operatorom uopštene aplikacije u termu $t(u, x.v)$. Barendregtova konvencija se treba primenjivati u oba slučaja.

Definicija 11 *Skup slobodnih promenljivih terma t, u oznaci $Fv(t)$, je induktivno definisan na sledeći način:*

1. $Fv(x) \ = \ \{x\}$;
2. $Fv(\lambda xt) \ = \ Fv(t) \setminus \ \{x\}$.
3. $Fv(t(u, x.v)) \ = \ (Fv(t) \cup Fv(u) \cup Fv(v)) \setminus \ \{x\}$.

U ΛJ-računu postoje sledeća 2 redukciona pravila:

$$\begin{array}{lrcl} (\beta) & (\lambda xt)(u, y.v) & \rightarrow & v[y := t[x := u]] \\ (\pi) & t(u, x.v)S & \rightarrow & t(u, x.vS) \end{array}$$

Oznaka S kod π-redukcije stoji umesto sintaktičke forme $(s, y.p)$. Supstitucija koja se javlja kod β-redukcije je u ovom računu meta-operator.
Skup normalnih formi se induktivno definiše na sledeći način:

$$\text{normalne forme: } t_{nf} \ ::= \ x \mid \lambda xt_{nf} \mid y(u_{nf}, x.v_{nf})$$

3.2.2 ΛJ-račun sa tipovima sa presekom

Kao i u slučaju λx-računa, i ovde je polazna osnova za konstrukciju tipskog sistema sa presekom koji bi karakterisao skup terma sa osobinom jake normalizacije tipski sistem $\mathcal{D}$, uveden u [6] za λ-račun. Osnovna razlika je u pravilu za tipiziranje aplikacije, koje je ovde promenjeno (i liberalizovano) kako bi moglo da se primeni na uopštenu aplikaciju. Tipski sistem je prikazan u Tabeli 4. Definicije skupa tipova, dodele tipova i baze su iste kao u λ-računu.

$$\frac{}{\Gamma, x : A \vdash x : A}\,(V) \qquad \frac{\Gamma, x : A \vdash t : B}{\Gamma \vdash \lambda x t : A \to B}\,(\lambda)$$

$$\frac{\Gamma \vdash t : A \quad \Gamma \vdash t : B}{\Gamma \vdash t : A \cap B}\,(\cap I) \qquad \frac{\Gamma \vdash t : A_1 \cap A_2}{\Gamma \vdash t : A_i, \quad i \in \{1,2\}}\,(\cap E)$$

$$\frac{\Gamma \vdash t : A \quad \Gamma \vdash u : B \quad \Gamma, x : C \vdash v : D \quad x \in Fv(v) \Rightarrow A \equiv B \to C}{\Gamma \vdash t(u, x.v) : D}\,(app)$$

Tabela 4: Tipski sistem sa presekom za ΛJ-račun

Pravilo (app) se može posmatrati kao skraćeni i ujedinjeni zapis pravila (app_1) i (app_2), prikazanih u Tabeli 5. Pravilo $(app)_1$ je očigledno uopštenje pravila za dodelu tipa aplikaciji u sistemu $\mathcal{D}$, dok pravilo $(app)_2$ odgovara-pravilu (app) u slučaju kada $x \notin Fv(v)$.

$$\frac{\Gamma \vdash t : A \to B \quad \Gamma \vdash u : A \quad \Gamma, x : B \vdash v : C}{\Gamma \vdash t(u, x.v) : C}\,(app_1)$$

$$\frac{\Gamma \vdash t : A \quad \Gamma \vdash u : B \quad \Gamma \vdash v : D \quad x \notin Fv(v)}{\Gamma \vdash t(u, x.v) : C}\,(app_2)$$

Tabela 5: Dekompozicija pravila (app)

Iako je na neki način pravilo (app_2) veštački dodato sistemu, što rezultira robusnim pravilom (app), Matthes je primerom pokazao da je prisustvo pravila (app_2) ključno tj. da sistem bez tog pravila ne može da tipizira sve terme sa osobinom jake normalizacije. Naime, term $t_0 := (\lambda x.x(x, z.z))(\lambda x.x(x, z.z), y.z)$ ima osobinu jake normalizacije, ali mu se ne može dodeliti tip ako pravilo (app_2) izostavimo iz sistema, tj. pravilo (app) zamenimo pravilom (app_1). Sa druge strane, za sistem sa pravilom (app), važe i Teorema o jakoj normalizaciji i Teorema o karakterizaciji jake normalizacije (formulacija ovih tvrdjenja je analogna formulaciji u λx-računu, a dokazi se nalaze u [22]).

3.3 λ^{Gtz}-račun

λ^{Gtz}-račun je jedno od skorije uvedenih proširenja λ-računa. Konstruisao ga je Espirito Santo u radu [11], 2006. godine. Njegova osnovna karakteristika je što odgovara intuicionističkom sekventnom računu, za razliku od prethodna dva proširenja koja, kao i λ-račun, odgovaraju intuicionističkoj prirodnoj dedukciji. Prvi intuicionistički sekventni formalni račun je $\bar{\lambda}$-račun, koji je uveo Herbelin u [16] 1995. godine. λ^{Gtz}-račun predstavlja pojednostavljenu i u izvesnom smislu upotpunjenu modifikaciju Herbelin-ovog računa.

3.3.1 λ^{Gtz}-račun bez tipova

Sintaksa λ^{Gtz}-računa se sastoji od dve vrste izraza: *terma* i *konteksta*.

$$\begin{array}{llcl} \text{(Termi)} & t, u, v & ::= & x \,|\, \lambda x.t \,|\, tk \\ \text{(Konteksti)} & k & ::= & \hat{x}.t \,|\, u :: k \end{array}$$

Termi se, kao i u običnom λ-računu, grade od termskih promenljivih (koje pripadaju prebrojivom skupu), pomoću operatora apstrakcije i aplikacije. Medjutim, aplikacija se razlikuje od aplikacije λ-računa po tome što se term primenjuje na kontekst. Konteksti su u stvari *liste* (u uobičajenoj analogiji sa teorijom funkcija se mogu posmatrati kao liste argumenata), i grade se *selekcijom* promenljive u termu, ili *konkatenacijom* tj. povezivanjem terma i konteksta (operator :: čitamo kao *kons*). Elementarni kontekst je selekcija promenljive u sebi samoj - $\hat{x}.x$, koja u stvari predstavlja praznu listu [ ]. U izrazima $\lambda x.t$ i $\hat{x}.t$, t je domen operatora λx i $\hat{x}$, respektivno. U cilju smanjenja broja zagrada, domen ovih operatora se proširuje na desno koliko god je to moguće. Terminom *izraz* označavamo elemente unije skupova terma i konteksta. Izraze obeležavamo sa $G, G', ..., G_1, G_2, ...$

Definicija 12 *Skup slobodnih promenljivih nekog terma ili konteksta, u oznaci $Fv(\ )$, je definisan na sledeći način:*

$$\begin{array}{lcl} Fv(x) & = & \{x\ \} \\ Fv(\lambda x.t) & = & Fv(t) \setminus \{x\}; \\ Fv(tk) & = & Fv(t) \cup Fv(k); \\ Fv(\hat{x}.t) & = & Fv(t) \setminus \{x\}; \\ Fv(t :: k) & = & Fv(t) \cup Fv(k). \end{array}$$

Napomena Može se uočiti da su slobodne promenljive u λ^{Gtz}-računu one koje nisu vezane operatorima apstrakcije ni selekcije i Barendregtova konvencija se treba primenjivati u oba slučaja.

U λ^{Gtz}-računu postoje sledeća četiri redukciona pravila:

$$\begin{array}{llcl} (\beta) & (\lambda x.t)(u :: k) & \to & u\hat{x}.(tk) \\ (\pi) & (tk)k' & \to & t(k@k') \\ (\sigma) & t\hat{x}.v & \to & v\langle x := t\rangle \\ (\mu) & \hat{x}.xk & \to & k, \text{ ako } x \notin k \end{array}$$

gde su meta-operatori $\langle := \rangle$ i @ definisani pravilima:

- $x\langle x := u\rangle = u$;
- $y\langle x := u\rangle = y$;
- $(\lambda y.t)\langle x := u\rangle = \lambda y.t\langle x := u\rangle$;
- $(tk)\langle x := u\rangle = t\langle x := u\rangle k\langle x := u\rangle$;
- $(\hat{y}.v)\langle x := u\rangle = \hat{y}.v\langle x := u\rangle$;
- $(v :: k)\langle x := u\rangle = v\langle x := u\rangle :: k\langle x := u\rangle$;
- $(u :: k)@k' = u :: (k@k')$;
- $(\hat{x}.v)@k' = \hat{x}.vk'$.

β-redukcija ima isti smisao kao i u λ-računu. π-redukcija je u stvari permutacija, čija je namena da "dotera" tj. spoji liste pre aplikacije na term, što odgovara call-by-value strategiji izračunavanja. σ-redukcija je supstitucija i realizuje se preko meta-operatora.

Najzad, μ-redukcija služi za brisanje uzastopnih pojavljivanja dve suprotne operacije - aplikacije i selekcije iste promenljive, ukoliko ona nije sadržana u

kontekstu.
Napomena Budući da se supstitucija u λ^{Gtz}-računu realizuje preko meta-operatora, tj. pošto nije deo sintakse, λ^{Gtz}-račun nije račun sa eksplicitnom supstitucijom. Medjutim, ovaj račun zahvaljujući definiciji β-redukcije sadrži mogućnost odlaganja supstitucije, što je osnovno svojstvo računa sa eksplicitnom supstitucijom. Takodje, supstitucija je "vidljiva", tj. može se pratiti u toku računa. Zbog toga ovaj račun možemo smatrati računom sa "implicitno-eksplicitnom" supstitucijom.

Normalne forme u λ^{Gtz}-računu su sledećeg oblika:

$$\begin{array}{llcl} (\text{Termi}) & t_{nf}, u_{nf}, v_{nf} & = & x_{nf} \mid \lambda x.t_{nf} \mid x(u_{nf} :: k_{nf}) \\ (\text{Konteksti}) & k_{nf} & = & \hat{x}.t_{nf} \mid t_{nf} :: k_{nf}. \end{array}$$

Dakle, redukcije su usmerene ka eliminisanju cut-ova (aplikacija) tj. samo su trivijalni cut-ovi dozvoljeni u normalnim formama.

Primer 13 *Posmatrajmo term $T : (\lambda x.y)(y(\hat{z}.z) :: \hat{x}.\lambda y.x)$ i moguće načine njegovog redukovanja. Budući da je promenljiva x vezana u $\hat{x}.\lambda y.x$, na početku ćemo primeniti Barendregtovu konvenciju i term pisati sa preimenovanom promenljivom - $T : (\lambda x.y)(y(\hat{z}.z) :: \hat{u}.\lambda y.u)$.*

I način:

$$\begin{array}{lll} T & \to_\beta & (y\hat{z}.z)\hat{x}.(y\hat{u}.\lambda y.u) \\ & \to_\pi & y\ ((\hat{z}.z)@\hat{x}.(y\hat{u}.\lambda y.u)) \\ & \leftrightarrow & y\hat{z}.(z\hat{x}.(y\hat{u}.\lambda y.u) \\ & \to_\sigma & y\hat{z}.\ y\hat{u}.(\lambda y.u)\langle x := z\rangle \\ & \leftrightarrow & y\hat{z}.(y\hat{u}.\lambda y.u) \\ & \to_\sigma & y\hat{z}.\ (\lambda y.u)\langle u := y\rangle \\ & \leftrightarrow & y\hat{z}.\ (\lambda y'.u)\langle u := y\rangle \\ & \leftrightarrow & y\hat{z}.\lambda y'.y \\ & \to_\sigma & (\lambda y'.y)\langle z := y\rangle \\ & \leftrightarrow & \lambda y'.y. \end{array}$$

II način:

$$\begin{array}{lll} T & \rightarrow_\beta & (y\hat{z}.z)\hat{x}.(y\hat{u}.\lambda y.u) \\ & \rightarrow_\sigma & (y\hat{u}.\lambda y.u)\langle x := y\hat{z}.z\rangle \\ & \leftrightarrow & y\hat{u}.\lambda y.u \\ & \rightarrow_\sigma & (\lambda y'.u)\langle u := y\rangle \\ & \leftrightarrow & \lambda y'.y. \end{array}$$

Napomena Može se primetiti da dva navedena načina odgovaraju različitim strategijama izračunavanja. Tako, prvi način odgovara call-by-value (prvo se "izračunava", tj. sredjuje lista π-redukcijom, a potom se zamenjuje u term), a drugi call-by-name strategiji (prvo σ-redukcijom uvrštavamo listu u term, u ovom slučaju nema ni potrebe za kasnijim sredjivanjem liste). Iako je u ovom slučaju konačan rezultat isti, u opštem slučaju to ne mora da važi tj. λ^{Gtz}-račun nije konfluentan, kao ni ostali formalni računi koji se zasnivaju na sekventnom računu. Konfluentnost se, medjutim, može obezbediti davanjem prioriteta jednoj od dve navedene strategije.

3.3.2 Osnovni tipski sistem

Osnovni tipski sistem za tipiziranje λ^{Gtz}-terma je sistem sa strelicom kao jedinim veznikom i biće označavan sa $\lambda^{Gtz} \rightarrow$.

Definicija 14 *Skup tipova, čije elemente označavamo sa $A, B, C, A', ...,$ je induktivno definisan na sledeći način:*

$$A, B \quad ::= \quad T \mid A \rightarrow B$$

gde je T proizvoljni element konačnog ili prebrojivog skupa tipskih promenljivih.

Definicija 15

(i) *Osnovna dodela tipa je deklaracija tj. izraz oblika $x : A$, gde je x termska promenjljiva a A je tip.*

(ii) *Baza Γ je skup osnovnih dodela tipova u kome su sve termske promenljive različite.*

(iii) *Postoje dve vrste dodele tipova:*

- $\Gamma \vdash t : A$ *za tipiziranje terma;*

- $\Gamma; B \vdash k : A$ *za tipiziranje konteksta.*

Po uzoru na $\bar{\lambda}$ i ovaj tipski sistem sadrži tzv. *stup* - posebno mesto sa leve strane sekventa koji dodeljuje tip kontekstu. Stup sadrži selektovanu formulu, koja se u pravilima ovog računa ponaša drugačije od ostalih. Tip, odnosno formula, u stup dolazi selekcijom, i sa takvom formulom se dalje nastavlja izračunavanje.
Iz definicije baze sledi da leva strana sekventa $\Gamma, x : A \vdash t : B$ implicira da promenljiva x nije deklarisana u bazi Γ, što će skraćeno biti označeno sa $x \notin dom\Gamma$.
Pravila tipskog sistema $\lambda^{Gtz} \rightarrow$ su prikazana u Tabeli 6.

$$\frac{}{\Gamma, x : A \vdash x : A}\,(Ax)$$

$$\frac{\Gamma, x : A \vdash t : B}{\Gamma \vdash \lambda x.t : A \rightarrow B}\,(\rightarrow_R)$$

$$\frac{\Gamma \vdash t : A \quad \Gamma; B \vdash k : C}{\Gamma; A \rightarrow B \vdash t :: k : C}\,(\rightarrow_L)$$

$$\frac{\Gamma \vdash t : A \quad \Gamma; A \vdash k : B}{\Gamma \vdash tk : B}\,(Cut)$$

$$\frac{\Gamma, x : A \vdash t : B}{\Gamma; A \vdash \hat{x}.t : B}\,(Sel)$$

Tabela 6: Osnovni tipski sistem $\lambda^{Gtz} \rightarrow$

Primer 16 *U λ-računu, term $\lambda xy.xy$ ima tip $(A \rightarrow B) \rightarrow (A \rightarrow B)$.*
Njemu odgovarajući term u λ^{Gtz}-računu je $\lambda x.\lambda y.x(y :: \hat{z}.z)$, i evo njegovog tipiziranja u sistemu $\lambda^{Gtz} \rightarrow$:

$$
\dfrac{\dfrac{\dfrac{}{x : A \to B, y : A \vdash x : A \to B}\,(Ax) \qquad \dfrac{\dfrac{}{x : A \to B, y : A \vdash y : A}\,(Ax) \qquad \dfrac{\dfrac{}{x : A \to B, z : B \vdash z : B}\,(Ax)}{x : A \to B; B \vdash \hat{z}.z : B}\,(Sel)}{x : A \to B, y : A; A \to B \vdash y :: \hat{z}.z : B}\,(\to_L)}{x : A \to B, y : A \vdash x(y :: \hat{z}.z) : B}\,(Cut)}{\dfrac{x : A \to B \vdash \lambda y.x(y :: \hat{z}.z) : A \to B}{\vdash \lambda x.\lambda y.x(y :: \hat{z}.z) : (A \to B) \to (A \to B)}\,(\to_R)}\,(\to_R)
$$

Napomena: Vidimo da postoji vrlo jednostavno preslikavanje izmedju λ-terma i λ^{Gtz}-terma. Medjutim, korespondencija se na ovaj način može uspostaviti samo izmedju skupova normalnih formi u oba sistema. Preslikavanje koje definiše sliku proizvoljnog λ-terma u λ^{Gtz}-računu je složenije forme, i biće definisano kasnije.

Espirito Santo je za ovako tipizirani λ^{Gtz}-račun dokazao svojstvo jake normalizacije [11], prevodjenjem u obični λ-račun posredstvom λ-računa sa eksplicitnom supstitucijom. Medjutim, kao i u λ-računu, osnovni tipski sistem ne može da obuhvati skup svih terma sa osobinom jake normalizacije. Primer za to je term $\lambda x.x(x :: \hat{y}.y)$ koji, iako je normalna forma, nema tip u sistemu $\lambda^{Gtz} \to$. U cilju karakterisanja jake normalizacije, bilo je potrebno uvesti tipski sistem sa presekom.

3.3.3 Tipski sistem sa presekom - $\lambda^{Gtz}\cap$

Tipski sistem za tipiziranje λ^{Gtz}-izraza koji osim veznika $\to$ sadrži i veznik $\cap$, uveden je u radu Espirito Santo et al. [9]. Detaljan prikaz ovog sistema i sistema koji su prethodili njegovom nastanku se može naći u [15, 12, 13]. Osnovna karakteristika predloženog sistema je da je presek ugradjen u već postojeća pravila osnovnog tipskog sistema, tako da nema posebnih pravila za uvodjenje ovog veznika.

Definicija 17 *Skup tipova, čiji elementi su označeni sa $A, B, C, A', ...$, je induktivno definisan na sledeći način:*

$$A, B \;::=\; T \mid A \to B \mid A \cap B$$

gde je T proizvoljni element konačnog skupa tipskih promenljivih.

Kao i u slučaju sistema $\mathcal{D}_{\leq}$, i ovaj sistem koristi relaciju poretka nad skupom tipova, ali ona nije eksplicitno uvedena u pravila tipiziranja, već je njena uloga jedino u definisanju ekvivalencije. Dakle, u ovom tipskom sistemu, ekvivalentni tipovi mogu zameniti jedni druge u dokazu. Posebnu ulogu igra sledeća ekvivalencija, koja sledi iz definicije 4.

Lema 18 $\cap(A \to B_i) \sim A \to \cap B_i, \;\; i = 1, ..., n.$

Tipski sistem koji uvodimo je prikazan u Tabeli 7.

$$\frac{}{\Gamma, x : \cap A_i \vdash x : A_i \;\; i \geq 1}\,(Ax)$$

$$\frac{\Gamma, x : A \vdash t : B}{\Gamma \vdash \lambda x.t : A \to B}\,(\to_R)$$

$$\frac{\Gamma \vdash t : A_i \;\; i = 1, ..., n \;\; \Gamma; B \vdash k : C}{\Gamma; \cap A_i \to B \vdash t :: k : C}\,(\to_L)$$

$$\frac{\Gamma \vdash t : A_i, \; \forall i \;\; \Gamma; \cap A_i \vdash k : B}{\Gamma \vdash tk : B}\,(Cut)$$

$$\frac{\Gamma, x : A \vdash t : B}{\Gamma; A \vdash \hat{x}.t : B}\,(Sel)$$

Tabela 7: Tipski sistem sa presekom $\lambda^{Gtz}\cap$

Predloženi tipski sistem zadovoljava osobine koje ga čine "dobrim" tipskim sistemom, a koje su artikulisane u tvrdjenjima koja bez dokaza navodimo u nastavku ovog poglavlja. Dokazi ovih tvrdjenja se mogu naći u [9, 12, 13, 17].

Uvodna tvrdjenja

Tvrdjenja koja slede su osnovna, uvodna tvrdjenja koja predstavljaju neophodan alat za dokazivanje važnih osobina nekog tipskog sistema. Medju njima se po značaju posebno ističe tzv. *Lema o generisanju* koja ispituje kako odredjeni izrazi u toku izvodjenja dokaza (tj. dodele tipa) mogu nastati.

Tvrdjenje 19 (Proširenje baze)

(i) $\Gamma \vdash t : A \;\Leftrightarrow\; \Gamma, x : B \vdash t : A$ *i* $x \notin Fv(t)$.

(ii) $\Gamma; C \vdash k : A \;\Leftrightarrow\; \Gamma, x : B; C \vdash k : A$ *i* $x \notin Fv(k)$.

Definicija 20 *Presek dve baze se definiše na sledeći način:*

$$\begin{array}{rcl} \Gamma_1 \cap \Gamma_2 & = & \{x : A | x : A \in \Gamma_1 \ \&\ x \notin dom\Gamma_2\} \\ & \cup & \{x : A | x : A \in \Gamma_2 \ \&\ x \notin dom\Gamma_1\} \\ & \cup & \{x : A \cap B | x : A \in \Gamma_1 \ \&\ x : B \in \Gamma_2\}. \end{array}$$

Tvrdjenje 21 (Presek baza) *Za proizvoljne baze* Γ_1, Γ_2 *važi:*

$$\Gamma_1 \vdash t : A \;\Rightarrow\; \Gamma_1 \cap \Gamma_2 \vdash t : A.$$

Tvrdjenje 22 (Pravilo $(\cap_L)$**)**

(i) Ako $\Gamma, x : A \vdash t : C$ *onda* $\Gamma, x : A \cap B \vdash t : C$.

(ii) Ako $\Gamma, x : A; D \vdash k : C$ *onda* $\Gamma, x : A \cap B; D \vdash k : C$.

Tvrdjenje 23 (Lema o generisanju - GL)

(i) $\Gamma \vdash x : A$ *ako i samo ako* $x : A \cap A_i \in \Gamma,\ i = 1, ..., n$ *za neko* $n \geq 0$.

(ii) $\Gamma \vdash \lambda x.t : A$ *ako i samo ako* $A \equiv \cap B_i \to C$ *i* , $\Gamma, x : \cap B_i \vdash t : C$.

(iii) $\Gamma; A \vdash \hat{x}.t : B$ *ako i samo ako* $\Gamma, x : A \vdash t : B$.

(iv) $\Gamma \vdash tk : A$ *ako i samo ako postoji tip* $\cap B_i,\ i = 1, ..., n$ *takav da za svako i važi* $\Gamma \vdash t : B_i$ *i* $\Gamma; \cap B_i \vdash k : A$.

(v) $\Gamma; T \vdash t :: k : C$ *ako i samo ako* $T \equiv \cap A_i \to B$, *i* $\Gamma; B \vdash k : C$ *i za svako i*, $\Gamma \vdash t : A_i$.

Tvrdjenje 24 *Normalne forme imaju tip.*

Očuvanje tipa prilikom redukcije

U cilju dokazivanja ovog svojstva za $\lambda^{Gtz}\cap$ sistem, potrebno je ispitati kako se meta-operatori $\langle := \rangle$ i @ ponašaju pri redukcijama.
Tvrdjenje koje sledi je analogno sa Lemom o zameni u običnom λ-računu.

Lema 25 (Lema o zameni)

(i) *Ako* $\Gamma, x : \cap A_i \vdash t : B$ *i* $\Gamma \vdash u : A_i$, *za svako* i, *onda* $\Gamma \vdash t\langle x := u\rangle : B$.

(ii) *Ako* $\Gamma, x : \cap A_i; C \vdash k : B$ *i* $\Gamma \vdash u : A_i$, *za svako* i, *onda* $\Gamma; C \vdash k\langle x := u\rangle : B$.

Sledeća lema je analogna prethodnoj, ali tretira meta-operator @.

Lema 26 (Append lemma) *Ako* $\Gamma; C \vdash k : B$ *i* $\Gamma; B \vdash k' : A$, *tada* $\Gamma; C \vdash k@k' : A$.

Sada možemo formulisati teoremu o očuvanju tipa pri redukciji za sve četiri redukcije λ^{Gtz}-računa.

Teorema 27 (Subject Reduction) *Ako* $\Gamma \vdash t : A$ *and* $t \to t'$, *onda* $\Gamma \vdash t' : A$.

Očuvanje tipa pri ekspanziji

Očuvanje tipa pri ekspanziji (proces obrnutog smera od redukcije) se pokazuje analogno očuvanju tipa pri redukciji. Zbog prisustva meta-operatora u izvršavanju σ i π ekspanzije, prvo moramo formulisati tvrdjenja inverzna Lemi o zameni i Append lemi.

Lema 28 (Inverzna lema o zameni)

(i) *Neka* $\Gamma \vdash v\langle x := t\rangle : A$, *i neka* t *ima tip. Tada postoje baza* Γ' *i tip* $B \equiv \cap B_i$, *takvi da* $\Gamma', x : \cap B_i \vdash v : A$ *i za svako* i, $\Gamma' \vdash t : B_i$.

(ii) *Neka* $\Gamma; C \vdash k\langle x := t\rangle : A$ *i neka* t *ima tip. Tada postoje baza* Γ' *i tip* $B \equiv \cap B_i$, *takvi da* $\Gamma', x : \cap B_i; C \vdash k : A$ *i za svako* i, $\Gamma' \vdash t : B_i$.

Lema 29 (Inverzna append lema) *Neka* $\Gamma; B \vdash k@k' : A$. *Tada postoji tip* $C \equiv \cap C_i$ *takav da za svako* i $\Gamma; B \vdash k : C_i$, *i* $\Gamma; \cap C_i \vdash k' : A$.

Tvrdjenje 30 (Subject expansion) *Ako* $t \to t'$, t *je redeks i* t' *ima tip u sistemu* $\lambda^{Gtz}\cap$, *onda i* t *ima isti tip u* $\lambda^{Gtz}\cap$.

Jaka normalizacija i karakterizacija jake normalizacije

Ove dve osobine, ako su istovremeno prisutne u nekom tipskom sistemu, omogućavaju da se na relativno jednostavno proverljiv način izoluju termi kod kojih ne postoji mogućnost beskonačnog izračunavanja, a to su termi sa kojima je bezbedno raditi. Zbog toga je ovo preduslov za implementaciju nekog formalnog računa u programski jezik. Teorema o karakterizaciji jake normalizacije 31 je posledica tvrdjenja 24 i tvrdjenja 30. Dokaz Teoreme o jakoj normalizaciji 32 zahteva nekolicinu pomoćnih, vrlo tehničkih lema koje u ovom radu neće biti navedene.

Teorema 31 (SN $\Rightarrow$ tipiziranost) *Svi termi sa osobinom jake normalizacije imaju tip u* $\lambda^{Gtz}\cap$ *sistemu.*

Teorema 32 (Tipiziranost $\Rightarrow$ SN) *Ako neki* λ^{Gtz}*-term* t *ima tip u* $\lambda^{Gtz}\cap$, *onda je* t $\beta\pi\sigma\mu$*-SN.*

Konačno, kao posledicu prethodne dve teoreme dobijamo glavnu teoremu:

Teorema 33 (Tipiziranost $\Leftrightarrow$ SN) λ^{Gtz}*-term* t *ima tip u* $\lambda^{Gtz}\cap$ *ako i samo ako je* t $\beta\pi\sigma\mu$*-SN.*

Glava 4

Klase λ^{Gtz}-terma

U ovoj glavi ćemo pokazati kako se λx- termi, ΛJ-termi i λ-termi mogu prikazati kao klase λ^{Gtz}-terma. Ovaj pristup omogućava dokazivanje očuvanja jake normalizacije λ-terma u λ^{Gtz}-računu (poznato kao PSN osobina). Takodje, predstavićemo tipske sisteme sa presekom koji ne sadrže dodatna pravila tipiziranja kao sistemi ponudjeni u [22] i [20], a koji takodje karakterišu terme sa osobinom jake normalizacije.

4.1 Uopštena aplikacija i eksplicitna supstitucija

U prethodnoj glavi je pokazano da su tzv. "prirodni" tipski sistemi za tipiziranje terma sa uopštenom aplikacijom ili sa eksplicitnom supstitucijom morali biti modifikovani tj. dodata su im u neku ruku "sintetička" pravila (pravilo app_2 u ΛJ; pravila *drop* i $K - Cut$ u λx) da bi se obezbedilo da se svakom termu sa osobinom jake normalizacije može dodeliti tip. Prirodno, nameću se dva pitanja: prvo, zbog čega "prirodna" pravila tipiziranja nisu dovoljna za obuhvatanje skupa jako normalizovanih terma; drugo, da li se taj cilj može postići promenom pravila redukcije posmatranih računa? U nastavku ćemo pokazati da je λ^{Gtz}-račun i njegov tipski sistem $\lambda^{Gtz}\cap$ od koristi za davanje odgovora na postavljena pitanja.

Definicija 34 *Neka je t* λ^{Gtz}*-term.*

1. *t je* λJ-term *ako je svako sečenje u t oblika* $t(u :: \hat{x}.v)$.

2. *t je* λx-term *ako je svako sečenje u t oblika $t(u :: \hat{x}.x)$ ili $t(\hat{x}.v)$.*

Usvojićemo oznaku "λJ-term" (umesto "ΛJ-term") zbog ujednačavanja terminologije. Koristićemo sledeće skraćenice:

$t(u, x.v)$	će označavati	$t(u :: \hat{x}.v)$;
$t(u)$	će označavati	$t(u :: \hat{x}.x)$;
$\langle t/x \rangle v$	će označavati	$t(\hat{x}.v)$.

Koristeći novouvedene oznake, dobijamo sledeću induktivnu karakterizaciju odredjenih klasa terma (koja se podudara sa sintaksom iz prethodne glave):

$$
\begin{array}{lll}
(\lambda J\text{-termi}) & t, u, v & ::= \; x \,|\, \lambda x.t \,|\, t(u, x.v) \\
(\lambda x\text{-termi}) & t, u, v & ::= \; x \,|\, \lambda x.t \,|\, t(u) \,|\, \langle t/x \rangle v
\end{array}
$$

Kada onim pravilima tipiziranja sistema $\lambda^{Gtz}\cap$ u kojima se ne pojavljuju konteksti dodamo sledeća pravila za tipiziranje uopštene aplikacije, aplikacije i eksplicitne supstitucije, prikazana u Tabeli 8, dobijamo tipske sisteme definisane sledećim pravilima:

Definicija 35

1. $\lambda J\cap := (Ax) + (\rightarrow_R) + (Gen.Elim)$.
2. $\lambda x\cap := (Ax) + (\rightarrow_R) + (Elim) + (Subst)$.

$\lambda J\cap$ se može smatrati "prirodnim" sistemom za tipiziranje λJ-terma, na dva načina. Kao prvo, pravila u sistemu $\lambda J\cap$ odgovaraju prirodnoj dedukciji. Iz sistema $\lambda^{Gtz}\cap$ su preuzeta samo ona pravila koja deluju na desnoj strani formule svakog sekventa, dok su pravila koja deluju na levoj strani formule (($\rightarrow_l$), (Sel) i (Cut)) zamenjena eliminacionim pravilom, u duhu prirodne dedukcije. Kao drugo, $\lambda J\cap$ sadrži osnovno, uobičajeno pravilo za tipiziranje uopštene aplikacije, a ne neka njegova proširenja, kao u radu [22]. Slično tome, i $\lambda x\cap$ se može smatrati "prirodnim" sistemom za tipiziranje λx-terma. U njemu su ponovo zadržana samo ona pravila iz $\lambda^{Gtz}\cap$ koja deluju na desnoj strani sekventa, a ostala pravila su zamenjena uobičajenim pravilima za eliminaciju i substitucuju. U ovom sistemu dodatna pravila za tipiziranje supstitucije i pravila sečenja nisu potrebna, za razliku od sistema predloženog u radu [20].

$$\frac{j \in \{1, \cdots, n\}}{\Gamma, x : \cap A_i \vdash x : A_j} \ (Ax)$$

$$\frac{\Gamma, x : A \vdash t : B}{\Gamma \vdash \lambda x.t : A \to B} \ (\to_R)$$

$$\frac{\Gamma \vdash t : \cap A_k \to B \quad \Gamma \vdash u : A_k \ , \forall k \in \{1, \cdots, n\}}{\Gamma \vdash t(u) : B} \ (Elim)$$

$$\frac{\Gamma \vdash t : A_k, \ \forall k \in \{1, \cdots, n\} \quad \Gamma, x : \cap A_k \vdash v : B}{\Gamma \vdash \langle t/x \rangle v : B} \ (Subst)$$

$$\frac{\Gamma \vdash t : \cap A_k \to B_i \ , \forall i \in \{1, \cdots, m\} \quad \Gamma \vdash u : A_k \ , \forall k \in \{1, \cdots, n\} \quad \Gamma, x : \cap B_i \vdash v : C}{\Gamma \vdash t(u, x.v) : C} \ (Gen.Elim)$$

Tabela 8: Pravila tipiziranja u sistemima $\lambda J \cap$ i $\lambda x \cap$

Za uvedene sisteme ćemo dokazati odredjena tvrdjenja, od kojih je prvo dodatak Lemi o generisanju [23], a ujedno i njena posledica.

Tvrdjenje 36 (Dopuna leme o generisanju) *U tipskom sistemu* $\lambda^{Gtz}\cap$ *važi:*

1. $\Gamma \vdash t(u, x.v) : C$ *ako i samo ako postoje* A_i, B *takvi da* $\Gamma \vdash t : \cap A_i \to B$, *i* $\Gamma \vdash u : A_i$, *za svako* i, *i* $\Gamma, x : B \vdash v : C$.

2. $\Gamma \vdash t(u) : B$ *ako i samo ako postoje* A_i, B_j *takvi da* $\Gamma \vdash t : \cap A_i \to B \cap B_j$, *i* $\Gamma \vdash u : A_i$, *za svako* i.

3. $\Gamma \vdash \langle t/x \rangle v : B$ *ako i samo ako postoji* A *takvo da* $\Gamma \vdash t : A$ *i* $\Gamma, x : A \vdash v : B$.

Dokaz: Kao i u Lemi o generisanju, i ovde dokaz direktno sledi iz činjenice da su pravila tipiziranja sintaksno usmerena tj. da postoji tačno jedno pravilo za tipiziranje svake klase izraza.

Sada možemo dokazati sledeću osobinu uvedenih tipskih sistema, poznatu pod imenom konzervativnost.

Tvrdjenje 37 (Konzervativnost)

1. *Neka je* t λJ*-term. U sistemu* $\lambda^{Gtz}\cap$ *možemo dokazati* $\Gamma \vdash t : A$ *ako i samo ako to možemo i u sistemu* $\lambda J\cap$.

2. *Neka je* t λx*-term. U sistemu* $\lambda^{Gtz}\cap$ *možemo dokazati* $\Gamma \vdash t : A$ *ako i samo ako to možemo i u sistemu* $\lambda x\cap$.

Dokaz: Kod oba dela ovog tvrdjenja, smer $(\Rightarrow)$ se dokazuje indukcijom po izvodjenju $\Gamma \vdash t : A$ u tipskim sistemima $\lambda J\cap$ i $\lambda x\cap$, respektivno, koristeći činjenicu da su pravila $(Gen.Elim)$, $(Elim)$, i $(Subst)$ u stvari izvedena pravila u tipskom sistemu $\lambda^{Gtz}\cap$.
Smer $(\Leftarrow)$ se dokazuje indukcijom po konstrukciji terma t, i oslanja se na Lemu o generisanju [23] i njenu dopunu [36].

Kao posledicu prethodnih tvrdjenja dobijamo karakterizaciju svojstva tipiziranosti terma u tipskim sistemima $\lambda J\cap$ i $\lambda x\cap$. Naglašavamo da je od suštinske važnosti posmatranje terma kao sekventnih tj. da pojam jake normalizacije ovde podrazumeva uključivanje sva 4 pravila redukcije λ^{Gtz}-računa.

Korolar 38

1. *Neka je* t λJ*-term. Tada,* t *je* $\beta\pi\sigma\mu - SN$ *ako i samo ako mu se može dodeliti tip u sistemu* $\lambda J\cap$.

2. *Neka je* t λx*-term. Tada,* t *je* $\beta\pi\sigma\mu - SN$ *ako i samo ako mu se može dodeliti tip u sistemu* $\lambda x\cap$.

Zahvaljujući tvrdjenjima dokazanim za λ^{Gtz}-račun sa tipovima sa presekom dokazanim u prethodnoj glavi, sada možemo tvrditi da "prirodni" tipski sistemi $\lambda J\cap$ i $\lambda x\cap$ obuhvataju tačno one terme sa osobinom jake normalizacije. Vratimo se sada na primere terma sa kraja prethodne glave,

$$\begin{aligned} t_0 &:= (\lambda x.x(x, w.w))(\lambda z.z(z, w.w), y.y'),\ \ y' \neq y\ , \\ t_2 &:= \langle \lambda z.zz/x\rangle\langle xx/y\rangle y'\ , \end{aligned}$$

Vidimo da iako oni zadovoljavaju osobinu jake normalizacije u računima ΛJ i λx, respektivno, oni to nisu kada se u obzir uzmu sve redukcije računa λ^{Gtz}, tj. kada se posmatraju kao sekventni termi. Zaista, posle jednog koraka β-redukcije, t_0 postaje $(\lambda z.z(z, w.w))\hat{x}.((x(x, w.w))\hat{y}.y')$, što je korišćenjem odgovarajućih skraćenica $\langle \lambda z.z(z)/x\rangle\langle x(x)/y\rangle y'$, tj. upravo t_2!. Nakon jednog koraka σ-redukcije, t_2 postaje term $\langle(\lambda z.z(z))(\lambda z.z(z))/y\rangle y'$, koji se može beskonačno redukovati uzastopnim primenama β i σ redukcija, tj. ne zadovoljava osobinu jake normalizacije. Stoga je sasvim u redu što se u tipskim sistemima $\lambda J\cap$ i $\lambda x\cap$ (baš kao i u sistemima iz radova [22] i [20] *bez* dodatnih pravila app_2, $drop$, i $K - Cut$) ovim termima ne mogu dodeliti tipovi, jer ovi termi ni nemaju osobinu jake normalizacije. Dakle, u ovom slučaju problem ne postoji. Ali zbog čega nije tako u osnovnim redukcijskim sistemima računa λx i ΛJ? Odgovor leži u tome što kod ΛJ računa ne postoji mogućnost odlaganja substitucije (veća β-redukcija) a kod λx računa ne postoji mogućnost kompozicije supstitucija, pa je zbog toga blokirano izvršenje spoljašnje supstitucije u termu t_2.

4.2 Lambda račun

Kako je λ-račun direktna restrikcija λx-računa koja se dobija eliminisanjem eksplicitne supstitucije, λ-termi se takodje mogu predstaviti kao podklasa λ^{Gtz}-terma, na sledeći način:

Definicija 39 *Neka je t λ^{Gtz}-term. t je* λ-term *ako je svaka njegova aplikacija oblika $t(u :: \hat{x}.x)$.*

Primenom ranije uvedene skraćenice, dobijamo induktivnu karakterizaciju:

$$(\lambda\text{-termi}) \qquad t, u, v \quad ::= \quad x \,|\, \lambda x.t \,|\, t(u)$$

Slično, polazeći od sistema $\lambda x\cap$, dobijamo tipski sistem $\lambda\cap$, prikazan u Tabeli 9.

Sledeća tvrdjenja su analogna prethodno formulisanim tvrdjenjima za sisteme $\lambda J\cap$ i $\lambda x\cap$, pa će biti navedena bez dokaza.

Tvrdjenje 40 (Konzervativnost) *Neka je t λ-term. U tipskom sistemu $\lambda^{Gtz}\cap$ je moguće dokazati $\Gamma \vdash t : A$ ako i samo ako je isto moguće u tipskom sistemu $\lambda\cap$.*

Korolar 41 *Neka je t λ-term. t je $\beta\pi\sigma\mu - SN$ ako i samo ako mu se može dodeliti tip u sistemu $\lambda\cap$.*

$$\frac{j \in \{1, \cdots, n\}}{\Gamma, x : \cap A_i \vdash x : A_j}\ (Ax)$$

$$\frac{\Gamma, x : A \vdash t : B}{\Gamma \vdash \lambda x.t : A \to B}\ (\to_R)$$

$$\frac{\Gamma \vdash t : \cap A_k \to B \quad \Gamma \vdash u : A_k \ , \forall k \in \{1, \cdots, n\}}{\Gamma \vdash t(u) : B}\ (Elim)$$

Tabela 9: tipski sistem $\lambda\cap$

Videli smo da postoje termi koji zadovoljavaju osobinu jake normalizacije u originalnim tipskim sistemima sistemima λx i ΛJ, ali je nemaju kada se posmatraju kao sekventni termi tj. kao podklase λ^{Gtz} terma. Sada ćemo pokazati da se to ne može desiti u λ računu, odnosno dokazaćemo da važi osobina Očuvanja jake normalizacije (skraćeno PSN - Preservation of Strong Normalisation). U cilju dokazivanja PSN, prvo moramo u λ-račun uvesti pomoćno pravilo redukcije. Prvo ćemo definisati preslikavanje F koje slika skup λ^{Gtz}-terma u skup λ-terma (ovo preslikavanje se koristi u dokazu jake normalizacije λ^{Gtz}-računa 32). Njegova ideja je u sledećem: ako je, na primer, $F(t) = M$, $F(u_i) = N_i$ i $F(v) = P$, onda se λ^{Gtz}-term $t(u_1 :: u_2 :: \hat{x}.v)$ preslikava u λ-term $(\lambda x.P)MN_1N_2$. Formalno, preslikavanje je definisamo istovremeno sa pomoćnim preslikavanjem F', na sledeći način:

$$\begin{aligned}
1.\ F(x) &= x \\
2.\ F(\lambda x.t) &= \lambda x.F(t) \\
3.\ F(tk) &= F'(F(t), k) \\
\\
4.\ F'(N, \hat{x}.t) &= (\lambda x.F(t))N \\
5.\ F'(N, u :: k) &= F'(NF(u), k)
\end{aligned}$$

Lema 42

$$F(t\langle x := u\rangle) = F(t)[x := F(u)],$$

gde izraz sa desne strane jednakosti označava meta-supstituciju u običnom λ-računu.

Dokaz: Indukcijom po strukturi terma t.

Primer: Neka je $T \equiv (\lambda x.x)(y :: x :: \hat{z}.(\lambda z.z))$. Tada je

$$\begin{aligned}
F(T) &= F'(F(\lambda x.x), y :: x :: \hat{z}.\lambda z.z) \\
&= F'(\lambda x.x, y :: x :: \hat{z}.\lambda z.z) \\
&= F'((\lambda x.x)y, x :: \hat{z}.\lambda z.z) \\
&= F'(((\lambda x.x)y)x, \hat{z}.\lambda z.z) \\
&= (\lambda z.F(\lambda z.z)))((\lambda x.x)y)x \\
&= (\lambda z.(\lambda z.z))((\lambda x.x)y)x.
\end{aligned}$$

Sledeće tvrdjenje sakuplja već poznate činjenice o λ-računu, potrebne za dokaz PSN osobine. Dokazi su u [9] i [10].

Tvrdjenje 43

1. *Ako $t \to_\beta u$ u λ, onda $t \to^+ u$ u λ^{Gtz}.*

2. *Ako je $F(t)$ $\beta\pi$-SN u λ, onda je t $\beta\pi\sigma\mu$-SN u λ^{Gtz}.*

3. *U λ, ako je t β-SN, onda je t $\beta\pi$-SN.*

Neka je u λ-računu β_0 pravilo koje redukuje $It \to t$, gde je $I := \lambda x.x$ a t proizvoljni term. Dakle, $\beta_0 \subset \beta$. Za λ-term t, imamo da je $F(t) \to^*_{\beta_0} t$, zbog toga što je $F(t(u)) = I(F(t)F(u))$.

Lema 44 *U λ računu važi:*

1. *Ako $t \to_\beta t_1$ i $t \to_{\beta_0} t_2$, onda postoji t_3 takav da $t_1 \to^*_{\beta_0} t_3$ i $t_2 \to_\beta t_3$.*

2. *Ako $t \to_\beta t_1$ i $t \to^*_{\beta_0} t_2$, onda postoji t_3 takav da $t_1 \to^*_{\beta_0} t_3$ i $t_2 \to_\beta t_3$.*

Dokaz:

1. Potrebno je razmotriti samo 3 situacije: (i) β_0-redeks je IM i β-redeks je u M; (ii) β-redeks je $(\lambda x.P)Q$ i β_0-redeks je u P; (iii) β-redeks je $(\lambda x.P)Q$ i β_0-redeks je u Q. U sva tri slučaja, željena komutacija je očigledna.

2. Dokaz direktno sledi iz prethodnog tvrdjenja.

Tvrdjenje 45 *Za proizvoljni λ-term t važi: t je β-SN u λ-računu ako i samo ako je $F(t)$ β-SN u λ-računu.*

Dokaz: ($\Rightarrow$): zato što $F(t) \to^*_{\beta_0} t$. ($\Leftarrow$): zbog toga što nam deo 2 Leme 44 dozvoljava da preslikamo beskonačni niz redukcija koji počinje od $F(t)$ u beskonačan niz redukcija koji počinje od t (kao $F(t) \to^*_{\beta_0} t$).

Teorema 46 (PSN) *Za proizvoljni λ-term t važi: t je β-SN u λ-računu ako i samo ako je $\beta\pi\sigma\mu$-SN u λ^{Gtz}-računu.*

Dokaz: ($\Rightarrow$): sledi direktno iz Tvrdjenja 43 (1).
($\Leftarrow$): neka je t β-SN. Na osnovu Tvrdjenja 45 dobijamo da je $F(t)$ β-SN. Zatim, iz Tvrdjenja 43 (3) sledi da je $F(t)$ $\beta\pi$-SN. Najzad, iz Tvrdjenja 43 (2) dobijamo da je t $\beta\pi\sigma\mu$-SN u λ^{Gtz}-računu.

Za razliku od Korolara 38, Korolar 41 se sada može kombinovati sa PSN teoremom, dajući tako novu karakterizaciju β-jake normalizacije u λ-računu.

Korolar 47 *Za proizvoljni λ-term t važi: t je β-SN u λ-računu ako i samo ako mu se može dodeliti tip u sistemu $\lambda\cap$.*

Dokaz: Iz Korolara 41 i PSN teoreme 46.

Glava 5

Zaključak

U ovom radu su predstavljeni neki formalni računi koji po Curry-Howardovoj korespodenciji odgovaraju intuicionističkoj logici. U centru interesovanja su bili tipski sistemi sa presekom pomoću kojih se karakteriše skup jako normalizovanih terma u ovim računima, zbog toga što je ova osobina od velike važnosti pri implementaciji nekog formalnog računa.
Prvo je ukratko predstavljen osnovni formalni račun - λ-račun. Data je njegova sintaksa, pravila redukcije i definisani su osnovni pojmovi teorije tipova. Prikazan je osnovni tipski sistem, a potom i tipski sistem sa presekom.
Zatim su date osnove dva proširenja λ-računa koji se takodje zasnivaju na prirodnoj dedukciji - λx-račun i ΛJ-račun. Predstavljene su verzije bez tipova i tipski sistemi sa presekom. Kroz odgovarajuće primere je objašnjen razlog uvodjenja pojedinih pravila tipiziranja, koja na prvi pogled ne deluju "prirodno".
Nešto detaljnije je predstavljen λ^{Gtz}-račun, koji se, za razliku od ostalih, zasniva na sekventnom računu za intuicionističku logiku. Objašnjena je verzija bez tipova, osnovni tipski sistem i tipski sistem sa presekom. Formulisana su tvrdjenja potrebna za dokaz glavnog svojstva ovog tipskog sistema, a to je osobina da se u njemu skup terma sa tipovima i skup terma sa osobinom jake normalizacije podudaraju. Budući da je ovaj račun relativno nov i ne u širokoj upotrebi, dati su primeri računanja i tipiziranja.
Najzad, pokazano je kako se ostala tri formalna računa mogu prikazati kroz λ^{Gtz}-račun, i kako ovaj ugao gledanja pomaže da se objasne pojedina svojstva posmatranih računa. Na kraju je dokazana osobina očuvanja jako normalizovanih λ-terma u λ^{Gtz}-računu, i predložen je novi tipski sistem pomoću kog se može okarakterisati skup β-jako normalizovanih λ-terma.

Zaključak ovog rada je da λ^{Gtz}-račun, zbog svoje sekventne prirode, ima finije redukcije (izračunavanja u manjim koracima) što ga čini dobrim sredstvom za ispitivanje svojstava formalnih računa.

Rezultati vezani za tipski sistem $\lambda^{Gtz}\cap$ su originalni, i predstavljaju rezultat zajedničkog rada sa Silviom Gilezan, Jose Espirito Santom i Silviom Likavec. Sumirani su u radovima [9], [15], [12] [17] i [13].

Literatura

[1] Abadi, M., Cardelli, L., Curien, P.-L. and Levy., J.-J.: Explicit substitutions. *Journal of Functional Programming*, Vol. 1(4):375-416, 1991.

[2] Amadio, R.M. i Curien, P.L.: Domains and lambda calculi. Cambridge University Press, Cambridge (1998).

[3] Barendregt, H.P.: The Lambda Calculus - Its Syntax and Semantics. North-Holland, Amsterdam (1984).

[4] Barendregt, H.P.: Lambda calculi with types. U: Abramsky, S., Gabbay, D.M., Maibaum, T.S.E. (eds.): *Handbook of Logic in Computer Science,* Vol. 2. Oxford University Press, Oxford (1992) 117–309.

[5] Bloo, R. i Rose, K.H.: Preservation of strong normalization in named lambda calculi with explicit substitution and garbage collection. In *CNS'95*, pages 62-72, 1995.

[6] Coppo, M. i Dezani-Ciancaglini, M.: A new type-assignment for lambda terms. U:*Archiv fr Mathematische Logik*, 19:139-156, 1978.

[7] de Bruijn, N.G.: A namefree lambda calculus with facilities for internal definition of expressions and segments. TH-report 78-WSK-03, Department of Mathematics, Technological University Eindhoven, Netherlands, 1978.

[8] Dougherty, D.J. and Lescanne, P.: Reductions, intersection types and explicit substitutions. U:*Mathematical structures in computer science* 13(1):55-85, 2003.

[9] Espírito Santo, J., Ghilezan, S. i Ivetić, J.: Characterising strongly normalising intuitionistic sequent terms. TYPES 2007, Lecture Notes in Computer Science 4941: 85-99 (2007)

[10] Espírito Santo, J.:Delayed substitutions. U: Baader,F.(ed):*Lecture Notes in Computer Science - Proceedings of RTA'07.* Vol. 4533. Springer-Verlag (2007).

[11] Espírito Santo, J.: Completing Herbelins programme. U: S. Ronchi Della Rocca (ed.): *Proceedings of 6th International Conference on Typed Lambda Calculi and Applications (TLCA 2007), Lecture Notes in Computer Science.* Vol. 4583. Springer-Verlag (2007).

[12] Espirito Santo, J., Ivetić, J. i Likavec, S.: "Intersection types for intuitionistic sequent terms". Intersection Types and Related Systems Workshop proceedings, Torino, 2008.

[13] Espirito Santo, J., Ivetić, J. i Likavec, S.: "Characterizing strongly normalising intuitionistic terms". Neobjavljen.

[14] Ghilezan, S.: Strong normalization and typability with intersection types. *Notre Dame Journal of Formal Logic* 37 (1996) 44–53.

[15] Ghilezan, S. i Ivetić, J.: Intersection types for intuitionistic λ^{Gtz}-calculus. Publications de l'Institute Mathematique, SANU, 82 (96) 159-164 (2007)

[16] Herbelin, H.:A lambda calculus structure isomorphic to Gentzen-style sequent calculus structure. U:*Computer Science Logic,CSL 1994*, Vol. 933. Springer-Verlag (1995) 61–75.

[17] Ivetić, J.:"Formalni računi za intuicionističku logiku". Magistarski rad, Fakultet tehničkih nauka, Univerzitet u Novom Sadu (2008)

[18] Joachimski, F. i Matthes, R.: Short Proofs of Normalization for the simply-typed lambda-calculus, permutative conversions and Godel's T. U:*Archive for Mathematical Logic* Vol. 42 (2003) 1, 59-87.

[19] Krivine, J. L.: Lambda-calcul: types et modèles. Masson, Paris (1990).

[20] Lengrand, S., Lescanne, P., Dougherty, D.J., Dezani-Ciancaglini, M. i Bakel, S. van: Intersection types for explicit substitutions. U: Longo, G. (ed):*Information and Computation,* Vol. 189. Elsevier (2003) 17–42.

[21] Likavec, S.: Metod redukcije u lambda računu sa tipovima sa presekom. Magistarski rad. Fakultet tehničkih nauka, Univerzitet u Novom Sadu (2005).

[22] Matthes, R.: Characterizing strongly normalizing terms of a calculus with generalized applications via intersection types. U: *ICALP Satellite Workshops*, pages 339354, 2000.

[23] Pottinger, G.: A type assignment for the strongly normalizable $\lambda-$terms. U: [25] 561–577.

[24] Ronchi Della Rocca, S.: Principal Type Scheme and Unification for Intersection Type Discipline.U: *Theoretical Computer Science* Vol. 59. Elsevier (1988) 181–209.

[25] Seldin J.P. i J.R. Hindley (eds.): To H.B. Curry: Essays on Combinatory Logic, Typed Lambda Calculus and Formalism. Academic Press, London, (1980).

[26] Wadler, Ph.: Proofs are Programs: 19th Century Logic and 21st Century Computing. U: *DR. Dobbs Journal (2000).*