Magistarski rad

Teorija unifikacije i
primena u eliminaciji kvantifikatora

Autor: Mirna Udovičić , Prirodno-matematički fakultet Beograd

Mentor: prof. dr. Aleksandar Jovanović

Godina: 2007

Sadržaj

1. Matematička logika
1.1. Jezik teorije predikatskog računa prvog reda....3
1.2. Termi i formule....3
1.3.Elementi iskazne logike....6
1.4. Teorije....8
1.5. Primeri teorija....13

2.Teorija unifikacije
2.1. Definicije....17
2.2. Unifikacija terma....18

3. Eliminacija kvantifikatora
3.1.Uvod....38
3.2.Teorija gustog uređenja sa prvim i zadnjim elementom....44
3.3.Teorija algebarskih zatvorenih polja....46
3.4.Teorija realnih zatvorenih polja....50
3.5.Teorija diskretnog uređenja bez prvog ili poslednjeg elementa....58

4. Literatura
4.1. Spisak referenci....64

Uvod

1. Matematička logika

1.1. Jezik teorije predikatskog računa prvog reda

def Jezik L teorije predikatskog računa prvog reda je bilo koji skup konstantnih simbola, funkcijskih simbola i relacijskih simbola:

$$L = Fnc_L \cup \mathrm{Re}l_L \cup Const_L,$$

gde je:

$Fnc_L = \{\, s \in L \mid s$ je funkcijski simbol u L $\}$
$\mathrm{Re}l_L = \{\, s \in L \mid s$ je relacijski simbol u L $\}$
$Const_L = \{\, s \in L \mid s$ je konstantni simbol u L $\}$.

Svi gore navedeni skupovi su međusobno disjunktni, i svaki od njih može biti prazan skup. Mi ćemo se samo dalje baviti logikom sa jednakošću.

Funkcija $ar : L \to N_0$ dodeljuje svakom simbolu $s \in L$ njegovu dužinu, odnosno broj mesnih argumenata. Ako važi da $s \in Const_L$, definišemo ar(s) = 0, dok je za $s \in Fnc_L$ Y $\mathrm{Re}l_L$, ar(s) ≥ 1.

Naprimer, $L = \{+, \cdot, -, \leq, 0, 1\}$ je jezik teorije uređenih polja, gde je:
$Fnc_L = \{+, -, \cdot\}$, ar(+) = 2, ar(−) = 1,
$\mathrm{Re}l_L = \{\leq\}$, ar($\leq$) = 2,
$Const_L = \{0, 1\}$.

1.2. Termi i formule

Termi i formule teorije predikatskog računa prvog reda su specijalni konačni nizovi simbola jezika L, i logičkih simbola predikatskog računa prvog reda (koji ćemo dalje označavati PR^1). Logički simboli u PR^1 su ustvari simboli osnovnih logičkih operacija i simboli promenljivih : $\wedge$ (i), $\vee$ (ili), $\Rightarrow$ (implikacija), $\Leftrightarrow$ (ekvivalencija), $\neg$ (negacija), znak semantičke ekvivalencije (odnosno jednakosti) $\equiv$, kvantifikatori $\forall$ (univerzalni kvantifikator), $\exists$ (egzistencijalni kvantifikator), i beskonačan niz promenljivih $v_0, v_1, v_2, \dots$.

def1 Termi, ili algebarski izrazi jezika L definišu se induktivno:

1. Promenljive i konstantni simboli su termi.

2. Ako je $F \in Fnc_L$ dužine n, i $t_1, \dots, t_n$ su termi jezika L, tada je $F(t_1, \dots, t_n)$ term jezika L.

3. Svaki term jezika L može se dobiti primenjujući pravila 1. i 2. konačan broj puta.

Termi jezika L mogu se i formalnije definisati na sledeći način:

<u>def2</u> $Term^0 = Var \cup Const_L$,

$Term^{m+1} = \{F(t_1, \dots, t_n) \mid n \in N_0 , F \in Fnc_L , ar(F) = n , t_1 , \dots, t_n \in Term^m\}, m \in N_0$

$Term_L = \cup_n Term^n$.

Vidimo da smo definicijom2 uveli oznaku skupa svih terma jezika L: $Term_L$.
Dalje, uvodimo meru složenosti terma:

<u>def</u> Funkcija složenosti terma $co : Term_L \to N_0$ je mera složenosti terma, i definiše se na sledeći način:

Ako $t \in Term^0$, onda je $co(t) = 0$.
Ako $t \in Term^n \setminus Term^{n-1}$, onda je $co(t) = n$, $n \in N_0$.

Na sličan način, možemo definisati formule jezika teorije predikatskog računa prvog reda L. Najpre definišimo atomične formule:

<u>def</u> Niz simbola φ je atomična formula jezika L, akko φ ima jedan od sledeća dva oblika:

$u \equiv v$, u, v su termi u L,

$R(t_1, t_2, \dots, t_n)$,

R je $n-$arni relacijski simbol jezika L, i $t_1, t_2, \dots, t_n$ su termi u L.

Dalje navodimo opštu definiciju formula jezika L:

<u>def</u> Formule jezika L definišemo induktivno na sledeći način:
(At_L je oznaka skupa svih atomičnih formula jezika L)

$$For^0 = At_L$$

$$
\begin{aligned}
For^{n+1} = For^{n} \cup & \{(\varphi \wedge \psi) \mid \varphi,\psi \in For^{n}\} \cup \\
& \{(\varphi \vee \psi) \mid \varphi,\psi \in For^{n}\} \cup \\
& \{\neg\varphi \mid \varphi \in For^{n}\} \cup \\
& \{(\varphi \Rightarrow \psi) \mid \varphi,\psi \in For^{n}\} \cup \\
& \{(\varphi \Leftrightarrow \psi) \mid \varphi,\psi \in For^{n}\} \\
& \{\forall x\varphi \mid x \in Var,\ \varphi \in For^{n}\} \cup \\
& \{\exists x\varphi \mid x \in Var,\ \varphi \in For^{n}\},
\end{aligned}
$$

$$For_L = \cup_n For^{n}.$$

Dakle, skup For_L je skup svih formula jezika L.

Na skupu For_L takođe možemo uvesti meru složenosti, i to proširujući funkciju složenosti co na formule.

<u>def</u> Funcija $co: For_L \to N_0$ definiše se induktivno na sledeći način:

Ako $\varphi \in At_L$, onda je $co(\varphi) = 0$,

Ako $\varphi \in For^{n} \setminus For^{n-1}$, $n \in \mathbf{N}$, onda je $co(\varphi) = n$.

Promenljive koje se nalaze u sklopu kvantifikatora zovu se kvantifikovane (ili vezane promenljive). Takođe, možemo definisati promenljive koje nisu u sklopu kvantifikatora (one se zovu slobodne promenljive).

<u>def</u> Skup $F_V(\varphi)$ je skup slobodnih promenljivih koje se pojavljuju u formuli φ jezika L, i definiše se indukcijom po meri složenosti φ:

1. Ako Ako $\varphi \in At_L$, onda je $F_V(\varphi)$ je skup promenljivih koje se pojavljuju u formuli φ.
2. $F_V(\neg\varphi) = F_V(\varphi)$.
3. $F_V(\varphi \wedge \psi) = F_V(\varphi \vee \psi) = F_V(\varphi \Rightarrow \psi) = F_V(\varphi \Leftrightarrow \psi) = F_V(\varphi) \cup F_V(\psi)$
4. $F_V(\exists x\varphi) = F_V(\forall x\varphi) = F_V(\varphi) \setminus \{x\}$.

Promenljive formule φ koje se ne nalaze u skupu je $F_V(\varphi)$ zovu se vezane promenljive formule φ. Naprimer, ako je $\varphi = (\neg x \equiv 0 \Rightarrow \exists y(x \cdot y \equiv 1))$, onda je $F_V(\varphi) = \{x\}$, pa je x slobodna, a y vezana promenljiva u φ.

Ako $\varphi \in For_L$, onda se koristi oznaka $\varphi(x_0, x_1, \ldots, x_n)$ koja označava da su sve slobodne promenljive u φ promenljive $x_0, x_1, \ldots, x_n$.

<u>def</u> Formule φ koje ne sadrže slobodne promenljive zovu se rečenice.

Formula φ je rečenica ako za nju važi: je $F_V(\varphi) = \emptyset$.

Naprimer, neka je zadan jezik $L = \{\cdot, 0, 1\}$, gde je $\cdot$ binarni funkcijski simbol.

Sledeće formule su primeri rečenica jezika L:

$0 \equiv 1$, $\forall x(\neg x \equiv 0 \Rightarrow \exists y(x \cdot y \equiv 1))$.

Skup svih rečenica jezika L označavamo $Sent_L$.

1.3.Elementi iskazne logike

def1 Skup $A = \{true, false\}$ snabdeven samo operacijama $\neg, \wedge, \vee$ nazivamo Boolova algebra.

def2 Za dve iskazne formule F_1, F_2 kažemo da su semantički ekvivalentne akko je formula $F_1 \Leftrightarrow F_2$ tautologija. U tom slučaju pišemo $F_1 \equiv F_2$ i kažemo da su F_1, F_2 logički ravnopravne (logički jednake).

Dakle, $F_1 \equiv F_2$ ako je $\tau(F_1) = \tau(F_2)$.

Naprimer, $(p \Rightarrow q) \equiv (\neg p \vee q)$.

Semantički ekvivalentne formule možemo u radu zamenjivati jednu drugom.

Relacija "$\equiv$" na skupu svih iskaza **I** je refleksivna, simetrična i tranzitivna, tj. "$\equiv$" je relacija ekvivalencije na **I**. Ona deli skup **I** na klase ekvivalencije. Ako je $A \in \mathbf{I}$, tada je:

$$[A] = \{ F \mid F \in \mathbf{I} \wedge F \equiv A \}$$

jedna klasa ekvivalencije određena relacijom "$\equiv$".

def3 Iskaznoj formuli $F = F(p_1, \ldots, p_n)$, $p_k \in \mathbf{I}$ ($k = 1,2,\ldots, n$) možemo pridružiti funkciju:

$$\tau(F) = \varphi_F(x_1, \ldots, x_n) \in \{true, false\},$$

gde je $x_k = \tau(p_k) \in \{true, false\}$ ($k = 1,2,\ldots, n$). Takve funkcije nazivamo Boolovim ili iskaznim funkcijama (ili tablicom istinitosti) formule F.

def4 Iskazna formula F je u normalnoj formi, ako ona pretstavlja najprostiji oblik formule iz klase $[F]$.

Normalna forma formule F može imati konjunktivni ili disjunktivni oblik.

def5 (KNF) Formula $F = F(i_1, i_2, \ldots, i_n)$ je u konjunktivnoj normalnoj formi ako F ima oblik:

$$F_1 \wedge F_2 \wedge \cdots \wedge F_m$$

gde je svaka od formula F_k ($k = 1, 2, ..., m$) oblika: $p_1 \vee p_2 \vee \cdots \vee p_s$, $s = s(k)$, a svaki od p_j ($j = 1, 2, ..., s$) je ili neki od i_k ($k = 1, 2, ..., n$), ili njegova negacija $\neg i_k$.

def6: (DNF) Formula $F = F(i_1, i_2, ..., i_n)$ je u disjunktivnoj normalnoj formi ako F ima oblik:

$$D_1 \vee D_2 \vee \cdots \vee D_m$$

gde je svaka od formula D_k ($k = 1, 2, ..., m$) oblika $p_1 \wedge p_2 \wedge \cdots \wedge p_s$, $s = s(k)$, a svaki od p_j ($j = 1, 2, ..., s$) je ili neki od i_k ($k = 1, 2, ..., n$), ili njegova negacija $\neg i_k$.

Za svođenje iskazne formule F na konjunktivnu ili disjunktivnu normalnu formu koristimo odgovarajuće potrebne tautologije, a ako je potrebno koristimo i De Morganove zakone i zakone komutativnosti, asocijativnosti i distributivnosti.

Primenom zakon komutativnosti, asocijativnosti i distributivnosti (ako je neophodno i više puta) možemo formulu F dovesti do tzv. savršene konjunktivne (savršene disjunktivne) normalne forme.

def7 (SKNF) Formula $F = F(i_1, i_2, ..., i_n)$ je u savršenoj konjunktivnoj normalnoj formi F ako ako F ima onu formu KNF u kojoj svaki konjunktivni član F_k ($k = 1, 2, ..., m$) sadrži svaku od razmatranih promenljivih i_k ($k = 1, 2, ..., n$), tačno jedanput (sa jednostrukom negacijom ili bez nje).

def7 (SDNF) Formula $F = F(i_1, i_2, ..., i_n)$ je u savršenoj disjunktivnoj normalnoj formi F ako ako F ima onu formu DNF u kojoj svaki disjunktivni član D_k ($k = 1, 2, ..., m$) sadrži svaku od razmatranih promenljivih i_k ($k = 1, 2, ..., n$), tačno jedanput (sa jednostrukom negacijom ili bez nje).

Pri prevođenju formule u SKNF koristimo (ako je potrebno):
$p \equiv p \vee (q \wedge \neg q) \equiv (p \vee q) \wedge (p \vee \neg q)$.
Pri prevođenju formule u SKNF koristimo (ako je potrebno):
$p \equiv p \wedge (q \vee \neg q) \equiv (p \wedge q) \vee (p \wedge \neg q)$.
Dve iskazne formule su semantički ekvivalentne (tj. logički jednake) akko imaju istu SKNF (odnosno istu SDNF).

primer1.3.1: Naći SDNF i SKNF formule:
$$F = \neg p \wedge (q \Rightarrow r).$$

Zadatak ćemo uraditi pomoću tabele istinitosti.

p	Q	r	$\neg r$	$q \Rightarrow \neg r$	$\neg p$	F
true	*True*	*true*	*false*	*false*	*false*	*False*
true	*True*	*false*	*true*	*true*	*false*	*False*
true	*False*	*true*	*false*	*true*	*false*	*False*
true	*False*	*false*	*true*	*true*	*false*	*False*
false	*True*	*true*	*false*	*false*	*true*	*False*
false	*True*	*false*	*true*	*true*	*true*	*True*
false	*False*	*true*	*false*	*true*	*true*	*True*
false	*False*	*false*	*true*	*true*	*true*	*True*

$F \equiv (\neg p \wedge q \wedge \neg r) \vee (\neg p \wedge \neg q \wedge r) \vee (\neg p \wedge \neg q \wedge \neg r)$, što pretstavlja SDNF fomule F.

$$F \equiv (\neg p \vee \neg q \vee \neg r) \wedge (\neg p \vee \neg q \vee r) \wedge (\neg p \vee q \vee \neg r) \wedge$$
$$\wedge (\neg p \vee q \vee r) \wedge (p \vee \neg q \vee \neg r),$$ što pretstavlja SKNF formule F.

1.4. Teorije

def Teorija T predikatskog računa prvog reda jezika L je bilo koji skup rečenica u L.

Elementi skupa T zovu se aksiome u T. Postoji više pristupa u formalizaciji dokaza. Mi ćemo ovde koristiti Hilbertove formalne sisteme, koji su bazirani na aksiomama. Pravila izvođenja jezika teorije predikatskog računa prvog reda su sledeća:

1. Rečenične aksiome
   Ove aksiome se dobijaju iz iskaznih tautologija zamenom iskaznih slova formulama u L.

2. Aksiome identiteta
   Ako $\varphi \in For_L$, $t \in Term_L$, $x \in Var$, onda $\varphi(t \setminus x)$ označava formulu dobijenu iz φ substitucijom svake slobodne pojave promenljive x termom t. Možemo koristiti i jednostavniju oznaku: φt umesto $\varphi(t \setminus x)$. Navedimo aksiome identiteta:
   $x = x$
   $x_1 = y_1 \wedge \cdots \wedge x_n = y_n \Rightarrow t(x_1, \ldots, x_n) = t(y_1, \ldots, y_n)$, $n \in N_0$, $\varphi \in At_L$
   $x_1 = y_1 \wedge \cdots \wedge x_n = y_n \Rightarrow (\varphi x_1 \cdots x_n \Leftrightarrow \varphi y_1 \cdots y_n)$, $\varphi \in At_L$.

3. Aksiome koje sadrže kvantifikatore

$\forall x \varphi x \Rightarrow \varphi t$, $\quad \varphi \in For_L, t \in Term_L, x \in Var$.

$\varphi t \Rightarrow \exists x \varphi x$,

gde je φt dobijeno iz φx zamenom svake slobodne pojave promenljive x u φx termom t.

Pravila izvođenja:

Neka su φ i ψ formule u L.

1. Modus Ponens: $\dfrac{\varphi, \quad \varphi \Rightarrow \psi}{\psi}$

3. Pravila generalizacije: $\dfrac{\varphi \Rightarrow \psi}{\varphi \Rightarrow \forall x \psi}$, moguće je primeniti ovo pravilo ako se x ne pojavljuje slobodno u φ

   $\dfrac{\psi \Rightarrow \varphi}{\exists x \psi x \Rightarrow \varphi}$, moguće je primeniti ovo pravilo ako se x ne pojavljuje slobodno u φ

def Dokaz rečenice φ teorije T jezika L je svaki niz $\psi_1, \psi_2, \ldots, \psi_n$ formula jezika L takav da je $\varphi = \psi_n$, i svaka formula ψ_i, $i \leq n$, je ili logička aksioma, ili aksioma teorije T, ili je dobijena pomoću pravila izvođenja koja su primenjena redom na članove niza.

Ako postoji dokaz rečenice φ teorije T, onda se φ zove teorema u T, i u ovom slučaju koristimo oznaku $T \vdash \varphi$. Ako je T = Ø, onda jednostavno pišemo $\vdash \varphi$, i φ nazivamo teoremom predikatskog računa prvog reda.

Formula oblika $\varphi \wedge \neg \varphi$ naziva se kontradikcija.

def Teorija T je konzistentna ako ne postoji kontradikcija ψ takva da važi: $T \vdash \psi$.

Druga važna osobina teorija je kompletnost.

def Teorija T jezika L je kompletna ako za svaku rečenicu φ jezika L važi ili $T \vdash \varphi$, ili $T \vdash \neg \varphi$.

def Teorija T jezika L je deduktivno zatvorena ako T sadrži sve svoje teoreme.

teorema 1.4.1. (teorema dedukcije): Pretpostavimo da je T teorija jezika L i $T \vdash \varphi$, gde $\varphi \in For_L$. Tada, postoje rečenice $\theta_0, \theta_1, ..., \theta_n \in$ T, takve da važi:

$$\vdash \theta_0 \wedge \theta_1 \wedge \cdots \wedge \theta_n \Rightarrow \varphi .$$

Kao posledica teoreme dedukcije sledi činjenica da je teorija iskaznog računa prvog reda konzistentna akko je svaki konačan podskup od T konzistentan.

<u>def</u> Formula φ jezika iskaznog računa prvog reda L je u preneks normalnoj formi, ako φ ima oblik $Q_1x_1Q_2x_2...Q_nx_n\psi$, gde je ψ formula bez kvantifikatora, i Q_1, Q_2, ..., Q_n su neki od kvantifikatora $\forall, \exists$.

<u>teorema1.4.2.</u> Za svaku formulu φ jezika iskaznog računa prvog reda jezika L, postoji formula ψ jezika L u preneks normalnoj formi takva da važi: $\vdash \varphi \Leftrightarrow \psi$.

<u>Algoritam prevođenja formule u KNF</u>

Datu formulu prevodimo u njoj ekvivalentnu formulu, koja je u KNF, koristeći redom sledeće korake:

1. Pojednostavljenje formule

Ovaj korak izvodimo koristeći poznate tautologije:

$\varphi \wedge \varphi \to \varphi$
$\varphi \vee \varphi \to \varphi$
...
$\forall x\varphi \to \varphi$, ako x nije slobodna promenljiva u φ
$\exists x\varphi \to \varphi$, ako x nije slobodna promenljiva u φ

2. Negirana normalna forma

Formula je u negiranoj normalnoj formi (NNF), ako ona ne sadrži simbole ekvivalencije ili implikacije, i ako se svaki simbol negacije pojavljuje direktno ispred atoma. Transformacija u NNF se izvodi pomoću sledećih pravila:

$\neg(\varphi \wedge \psi) \to \neg\varphi \vee \neg\psi$
$\neg(\varphi \vee \psi) \to \neg\varphi \wedge \neg\psi$
$\neg(\forall x\varphi) \to \exists x\neg\varphi$
$\neg(\exists x\varphi) \to \forall x\neg\varphi$

$\varphi \Rightarrow \psi \to \neg\varphi \vee \psi$
$\neg\neg\varphi \to \varphi$

Naprimer, razmotrimo prevođenje formule $\neg(\varphi_1 \Leftrightarrow \varphi_2)$.

$$\neg(\varphi_1 \Leftrightarrow \varphi_2) \to$$
$$\neg((\varphi_1 \Rightarrow \varphi_2) \wedge (\varphi_2 \Rightarrow \varphi_1)) \to$$
$$\neg((\neg\varphi_1 \vee \varphi_2) \wedge (\neg\varphi_2 \vee \varphi_1)) \to$$
$$\neg(\neg\varphi_1 \vee \varphi_2) \vee \neg(\neg\varphi_2 \vee \varphi_1) \to$$
$$(\varphi_1 \wedge \neg\varphi_2) \vee (\varphi_2 \wedge \neg\varphi_1) \to$$
$$(\varphi_1 \vee \varphi_2) \wedge (\varphi_1 \vee \neg\varphi_1) \wedge (\neg\varphi_2 \vee \varphi_2) \wedge (\neg\varphi_2 \vee \neg\varphi_1).$$

3. Pomeranje kvantifikatora ka unutrašnjosti formula

Cilj pravila u ovom delu je pomeranje kvantifikatora što više ka unutrašnjosti formule. Pravila su sledeća:

$\exists x(\varphi \wedge \psi) \to \exists x\varphi \wedge \psi$, ako x nije slobodna promenljiva u ψ
$\exists x(\varphi \vee \psi) \to \exists x\varphi \vee \psi$, ako x nije slobodna promenljiva u ψ
$\forall x(\varphi \wedge \psi) \to \forall x\varphi \wedge \psi$, ako x nije slobodna promenljiva u ψ
$\forall x(\varphi \vee \psi) \to \forall x\varphi \vee \psi$, ako x nije slobodna promenljiva u ψ
$\forall x(\varphi \wedge \psi) \to \forall x\varphi \wedge \forall x\psi$,
ako je x slobodna promenljiva u φ i x slobodna promenljiva u ψ

$\exists x(\varphi \vee \psi) \to \exists x\varphi \vee \exists x\psi$
ako je x slobodna promenljiva u φ i x slobodna promenljiva u ψ.

Navedimo primer koji opisuje ove transformacije:

$$\forall x\exists y\forall z(R(x,x) \wedge (P(y) \vee R(x,y) \vee Q(z))) \to$$

$$\forall x\exists y(R(x,x) \wedge \forall z(P(y) \vee R(x,y) \vee Q(z))) \to$$

$$\forall x\exists y(R(x,x) \wedge (P(y) \vee R(x,y) \vee \forall zQ(z))) \to$$

$$\forall x(R(x,x) \wedge \exists y(P(y) \vee R(x,y) \vee \forall zQ(z))) \to$$

$$\forall x(R(x,x) \wedge (\exists yP(y) \vee \exists yR(x,y) \vee \forall zQ(z))) \to$$

$$\forall xR(x,x) \wedge \forall x(\exists yP(y) \vee \exists yR(x,y) \vee \forall zQ(z)) \to$$

$$\forall x R(x,x) \wedge (\exists y P(y) \vee \forall x \exists y R(x,y) \vee \forall z Q(z)).$$

4. Reimenovanje promenljivih

U ovom koraku vrši se reimenovanje promenljivih, ako je neophodno, tako da ne mogu postojati dve različite promenljive sa istim imenom. Naprimer, ako je data formula:

$$\forall x R(x,x) \wedge (\exists y P(y) \vee \forall x \exists y R(x,y) \vee \forall z Q(z)),$$

kada preimenujemo promenljive dobijamo formulu:

$$\forall y_1 R(y_1,y_1) \wedge (\exists y_2 P(y_2) \vee \forall y_3 \exists y_4 R(y_3,y_4) \vee \forall y_5 Q(y_5)).$$

5. Standardna Skolemizacija

U ovom koraku se oslobađamo egzistencijalnog kvantifikatora. U prethodnom primeru, ako primenimo pravilo Skolemizacije dva puta dobijamo formulu:

$$\forall y_1 R(y_1,y_1) \wedge (P(a) \vee \forall y_3 R(y_3, f(y_3)) \vee \forall y_5 Q(y_5)),$$

gde je a Skolemova konstanta i f je Skolemova funkcija. Skolemova funkcija f je funkcija koja zavisi od univerzalnog kvantifikatora.

6. Konjunktivna normalna forma

U ovom koraku vrši se pomeranje svih univerzalnih kvantifikatora ispred formule i prevođenje formule u konjunktivnu normalnu formu. Formule koje koristimo su sledeće:

$\forall x \varphi \wedge \psi \rightarrow \forall x(\varphi \wedge \psi)$, ako x nije slobodna promenljiva u ψ
$\forall x \varphi \vee \psi \rightarrow \forall x(\varphi \vee \psi)$, ako x nije slobodna promenljiva u ψ
$$\varphi \vee (\psi_1 \wedge \psi_2) \rightarrow (\varphi \vee \psi_1) \wedge (\phi \vee \psi_2)$$

Za primer koji smo razmatrali dobijamo sledeći niz transformacija:

$$\forall y_1 R(y_1,y_1) \wedge (P(a) \vee \forall y_3 R(y_3, f(y_3)) \vee \forall y_5 Q(y_5)) \rightarrow$$

$$\forall y_1 \forall y_3 \forall y_5 (R(y_1,y_1) \wedge (P(a) \vee R(y_3, f(y_3)) \vee Q(y_5))) \rightarrow$$

$$\forall y_1 \forall y_3 \forall y_5 ((R(y_1,y_1) \vee P(a)) \wedge (R(y_1,y_1) \vee R(y_3, f(y_3))) \wedge (R(y_1,y_1) \vee Q(y_5)))$$

7. Eliminacija univerzalnog kvantifikatora

Ovaj korak izvodimo jednostavno uklanjanjem univerzalnih kvantifikatora i dobijamo skup konjunkta. Uvek pretpostavljamo da su sve promenljive univerzalno kvantifikovane i da različiti konjunkti nemaju iste promenljive.
U prethodnom primeru, konačna formula koju dobijamo je sledeća:

$$\{\{R(y_1, y_1), P(a)\}, \{R(y_6, y_6), R(y_3, f(y_3))\}, \{R(y_7, y_7), Q(y_5)\}\},$$

pri čemu smo reimenovali pojavu promenljive y_1 redom sa y_6 i y_7.

1.5. Primeri teorija

primer1.5.1: Teorija čistog predikatskog računa sa identitetom. Za ovu teoriju imamo: $L = \emptyset$, $T = \emptyset$.
Teoreme ove teorije su tačno teoreme predikatskog računa prvog reda koje sadrže samo logičke simbole.

primer1.5.2: Teorija linearnog uređenja, LO. Jezik ove teorije je: $L = \{\leq\}$, gde je $\leq$ binarni relacijski simbol. Aksiome teorije T su:

1. $(\forall x)(x \leq x)$ refleksivnost
2. $(\forall x)(\forall y)(\forall z)(x \leq y \wedge y \leq z \Rightarrow x \leq z)$ tranzitivnost
3. $(\forall x)(\forall y)(x \leq y \wedge y \leq x \Rightarrow x = y)$ antisimetričnost
4. $(\forall x)(\forall y)(x \leq y \vee y \leq x)$ linearnost .

Teorija PO čije su aksiome 1.-3. zove se teorija parcijalnog uređenja. Binarni relacijski simbol $<$ može se definisati:
$x < y \Leftrightarrow x \leq y \wedge \neg x \equiv y$.

primer1.5.3: Teorija gustog linearnog uređenja bez krajnjih tačaka, DLO. Jezik ove teorije je: $L = \{\leq\}$. Aksiome su:

1. $(\forall x)(x \leq x)$ refleksivnost
2. $(\forall x)(\forall y)(\forall z)(x \leq y \wedge y \leq z \Rightarrow x \leq z)$ tranzitivnost
3. $(\forall x)(\forall y)(x \leq y \wedge y \leq x \Rightarrow x = y)$ antisimetričnost
4. $(\forall x)(\forall y)(x \leq y \vee y \leq x)$ linearnost
5. $(\forall x)(\exists y)(x < y)$
6. $(\forall x)(\exists y)(y < x)$
7. $(\forall x)(\forall y)(\exists z)(x < y \Rightarrow x < z \wedge z < y)$
8. $(\exists x)(\exists y)\neg(x = y)$

<u>primer1.5.4</u>: Teorija Abelovih grupa , Ab. U ovoj teoriji jezik je L = $\{+ , - , 0\}$, gde je + binarni funkcijski simbol, i $-$ je unarni funkcijski simbol. Aksiome su:

1. $\forall x \forall y \forall z((x+y)+z = x+(y+z))$	asocijativnost
2. $\forall x \forall y(x+y = y+x)$	komutativnost
3. $\forall x(x+0 = x)$	postojanje neutralnog elementa
4. $\forall x(x+(-x) = 0)$	postojanje inverznog elementa.

Lako se dokazuje indukcijom po složenosti terma: Ako $t \in Term_L$, tada postoji $k \in \mathbf{N}$ i brojevi $m_1, \ldots, m_k$ takvi da važi:

Ab $\vdash t \equiv m_1 x_1 + \cdots + m_k x_k$, gde su $x_1, \ldots x_k$ promenljive.

<u>primer1.5.5:</u> Teorija polja, F. Jezik ove teorije je L = $\{+ , - , \cdot , 0 , 1\}$. Aksiome su:

1. $\forall x \forall y \forall z(x+(y+z) = (x+y)+z)$
2. $\forall x \forall y(x+y = y+x)$
3. $\forall x(x+0 = x)$
4. $\forall x(x+(-x) = 0)$
5. $\forall x \forall y \forall z(x(yz) = (xy)z)$
6. $\forall x \forall y(xy = yx)$
7. $\forall x(x \cdot 1 = x)$
8. $\forall x \exists y(x = 0 \vee xy = 1)$
9. $\forall x \forall y \forall z(x(y+z) = xy + xz)$
10. $0 \neq 1$

Možemo uvesti novi funkcijski simbol $^{-1}$ u teoriju F pomoću aksiome:

$\forall x \forall y(x \neq 0 \Rightarrow (x \cdot y = 1 \Leftrightarrow y = x^{-1}))$. Tada se iz teorije F može izvesti:

$\forall x(x \neq 0 \Rightarrow x \cdot x^{-1} = 1)$.

<u>primer1.5.6:</u> Teorija uređenih polja, FO. Jezik ove teorije je L = $\{\leq , + , - , \cdot , 0 , 1\}$. Aksiome su:

1. $\forall x \forall y \forall z(x+(y+z) = (x+y)+z)$
2. $\forall x \forall y(x+y = y+x)$
3. $\forall x(x+0 = x)$
4. $\forall x(x+(-x) = 0)$
5. $\forall x \forall y \forall z(x(yz) = (xy)z)$
6. $\forall x \forall y(xy = yx)$

7. $\forall x(x \cdot 1 = x)$
8. $\forall x \exists y(x = 0 \vee xy = 1)$
9. $\forall x \forall y \forall z(x(y+z) = xy + xz)$
10. $0 \neq 1$
11. $\forall x \forall y \forall z(x \leq y \Rightarrow (x + z \leq y + z)$
12. $\forall x \forall y \forall z(x \leq y \wedge 0 < z \Rightarrow x \cdot y \leq y \cdot z)$.

Sledeća teorema je formula teorije FO:

$$x_1^2 + \cdots + x_n^2 = 0 \Rightarrow x_1 = 0 \wedge \cdots \wedge x_n = 0.$$

<u>primer1.5.7:</u> Teorija Boolovih algebri, BA. Jezik ove teorije je $L = \{+, \cdot, ', \leq, 0, 1\}$. Aksiome su:

1. $\forall x \forall y \forall z(x + (y + z) = (x + y) + z)$
2. $\forall x \forall y \forall z(x(yz) = (xy)z)$
3. $\forall x \forall y(x + y = y + x)$
4. $\forall x \forall y(xy = yx)$
5. $\forall x(x + 0 = x)$
6. $\forall x(x \cdot 1 = x)$
7. $\forall x(x + x' = 1)$
8. $\forall x(x \cdot x' = 0)$
9. $0 \neq 1$
10. $\forall x \forall y(x \leq y \Leftrightarrow x = x \cdot y)$

Lako se dokazuje da u BA važi:

1. Relacijski simbol $<$ zadovoljava aksiome parcijalnog uređenja. U odnosu na ovo uređenje, važi:

$$\sup\{x_1, \ldots, x_n\} = \sum_{i \leq n} x_i,$$
$$\inf\{x_1, \ldots, x_n\} = \prod_{i \leq n} x_i.$$

2. Za svaki $t \in Term_L$,

$$BA \vdash t(x_1, \ldots, x_n) = \sum_{\alpha \in 2^n} t(\alpha_1, \ldots, \alpha_n) x_1^{\alpha_1} \cdots x_n^{\alpha_n},$$

gde je $2^n = \{\alpha \mid \alpha : \{1, \ldots, n\} \to \{0,1\}\}$, $\alpha_i = \alpha(i)$, $1 \leq i \leq n$, i $x^0 = x'$, $x^1 = x$. Ova činjenica se dokazuje indukcijom po složenosti terma. Umesto operacijskih simbola $+$ i $\cdot$ mogu se koristiti simboli $\wedge$ i $\vee$.

primer1.5.8: Peano aritmetika , PA. Jezik ove teorije je $L = \{+, \cdot, ', \leq, 0, 1\}$. Aksiome su sledeće:

1. $\neg\exists x(x' = 0)$
2. $\forall x \forall y(x' = y' \Rightarrow x = y)$
3. $\forall x(x + 0 = x)$
4. $\forall x \forall y(x + y' = (x + y)')$
5. $\forall x(x \cdot 0 = 0)$
6. $\forall x \forall y(x \cdot y' = (x \cdot y) + x$
7. $\neg\exists x(x < 0)$
8. $\forall x \forall y(x < y' \Rightarrow x < y \vee x = y)$
9. $\forall x \forall y(x < y \vee x = y \vee y < x)$
10. $1 = 0'$

(I) Indukcijska šema: Neka je $\varphi x y_1 \cdots y_n$ formula jezika L. Tada je univerzalno zatvorenje:

$\varphi 0 y_1 \cdots y_n \wedge \forall x(\varphi x y_1 \cdots y_n \Rightarrow \varphi x' y_1 \cdots y_n) \Rightarrow \forall x \varphi x y_1 \cdots y_n$ aksioma teorije PA.

Ova se teorija takođe zove formalna aritmetika.

2.Teorija unifikacije

2.1.Definicije

Signatura je (konačna ili prebrojivo beskonačna) skup funkcijskih simbola F. Term algebra T(F,V) generisana je pomoću skupa F i (prebrojivo) beskonačnog skupa promenljivih V. Terme ćemo označavati slovima *l,r,s,t,u* i *v*, a promenljive slovima x,y,z i w. Skup promenljivih koje se pojavljuju u termu *t* označićemo Vars(*t*).

Substitucija je funkcija iz skupa promenljivih u skup terma koja je skoro svugde jednaka funkciji identiteta (identična substitucija se označava *Id*). Primena substitucije σ na term *t*, u oznaci *t*σ, definiše se indukcijom po strukturi terma:

$$t\sigma := \begin{cases} x\sigma, & ako\ t = x \\ f(t_1\sigma, \ldots t_n\sigma), & ako\ t = f(t_1, \ldots t_n) \end{cases}.$$

U drugom slučaju ove definicije, n = 0 je dozvoljeno: u ovom slučaju, f je konstantni simbol i *f*σ = *f*. Substitucija takođe može da se primeni na skupove terma, na jednakosti i skupove jednakosti.

Za substituciju σ, domen je skup promenljivih

$$Dom(\sigma) := \{x | x\sigma \neq x\},$$

rang je skup terma

$$Ran(\sigma) := \bigcup_{x \in Dom(\sigma)} \{x\sigma\}.$$

Substitucija može biti predstavljena eksplicitno kao funkcija pomoću skupa ugradnji promenljivih u domen:

$$\{x_1 \to s_1, \ldots, x_n \to s_n\}.$$

Restrikcija substitucije θ na skup promenljivih X (označava se $\theta|x$, je substitucija koja je jednaka funkciji identiteta svugde osim na $X \cap Dom(\sigma)$, gde je jednaka σ. Kompozicija dve substitucije se piše $\sigma\theta$, i definiše se na sledeći način:

$$t\sigma\theta = (t\sigma)\theta.$$

Algoritam za konstrukciju kompozicije $\sigma\theta$ dve substitucije predstavljene pomoću skupa ugradnji je sledeći:

1.Primenimo θ na svaki term u $Ran(\sigma)$ da dobijemo σ_1;

2.Uklonimo iz θ sve ugradnje $x \to t$, gde $x \in Dom(\sigma)$, da dobijemo θ_1;

3.Uklonimo iz σ_1 sve trivijalne ugradnje $x \to x$, da dobijemo σ_2;

4.Uzmimo uniju dva skupa ugradnji σ_2 i θ_1.

Substitucija je idempotentna ako $\sigma\sigma = \sigma$.
Dve substitucije su jednake, označeno $\sigma = \theta$, ako $x\sigma = x\theta$ za svaku promenljivu x. Kažemo da je σ opštija od θ, ako postoji preslikavanje η takvo da je $\theta = \sigma\eta$.

definicija: Substitucija σ je unifikator dva terma s i t ako $s\sigma = t\sigma$; ona je najopštiji unifikator (mgu), ako je za svaki drugi unifikator θ za s i t, σ opštiji od θ.Problem unifikacije dva terma s i t predstavljamo pomoću: $s =^? t$.

2.2. Unifikacija terma

Unifikacija pomoću rekurzivnog spusta

Ovaj algoritam je prvi opisao Robinson [1965], i univerzalno je korišćen u simboličkim računskim sistemima.

```
global σ : substitution;  { Initialized to Id }

Unify( s : term; t : term )
  begin
    if s is a variable then   { Instantiate variables }
      s := sσ ;
    if t is a variable then
      t := tσ;
    if s is a variable and s = t then
      { Do nothing }
    else if s = f(s_1,...s_n) and t = g(t_1,....,t_m) for n,m ≥ 0 then begin
      if f = g then
        for i := 1 to n do
          Unify(s_i,t_i);
```

```
    else Exit with failure   { Symbol clash }
  end
  else if s is not a variable then
    Unify( t,s );
  else if s occurs in t then
    Exit with failure;  { Occurs check }
  else σ := σ{s → t};
end;
```

Pristup zasnovan na pravilima *U*

Ovde je predstavljen jednostavan sistem izvođenja koji se koristi za rešavanje unifikacijskih problema.

Idempotentna substitucija $\{x_1 \to t_1,....,x_n \to t_n\}$ može biti predstavljena pomoću skupa jednakosti $\{x_1 \approx t_1,....,x_n \approx t_n\}$ u rešenoj formi, što znači da svaki x_i ima jednu pojavu u skupu. Za bilo koju idempotentnu substituciju σ, odgovarajući skup u rešenoj formi označavamo $[\sigma]$, i za bilo koji skup jednakosti S u rešenoj formi, odgovarajuću substituciju označavamo σ_S.

Sistem je ili simbol *false*(koji predstavlja neuspeh), ili par koji se sastoji od nepraznog skupa P unifikacijskih problema i skupa S jednakosti u rešenoj formi. Koristićemo Γ da označimo sistem. Kažemo da je substitucija unifikator (ili rešenje) sistema P;S ako ona unifikuje svaku jednakost iz P i S; sistem false nema unifikatore.

Sistem izvođenja *U* sastoji se iz sledećih transformacija sistema:

Trivial:

$$\{s =^? s\} \cup P';S \Rightarrow P';S$$

Decomposition:

$$\{f(s_1,....,s_n) =^? f(t_1,....,t_n)\} \cup P';S \Rightarrow \{s_1 =^? t_1,....,s_n =^? t_n\} \cup P';S$$

(Primetimo da može biti n = 0)

Symbol Clash:

$$\{f(s_1,....,s_n) =^? g(t_1,....,t_m)\} \cup P';S \Rightarrow \mathit{false}$$

ako $f \neq g$

Orient:

$$\{t =^? x\} \cup P';S \Rightarrow \{x =^? t\} \cup P';S$$

ako t nije promenljiva

Occurs Check:

$$\{x =^? t\} \cup P';S \Rightarrow \mathit{false}$$

ako $x \in Vars(t)$ ali $x \neq t$.

Variable Elimination:

$$\{x =^? t\} \cup P';S \Rightarrow P'\{x \to t\};S\{x \to t\} \cup \{x \approx t\}$$

ako $x \notin Vars(t)$.

U cilju da unifikujemo s i t, kreiramo inicijalni sistem $\{s =^? t\};\emptyset$ i primenjujemo redom pravila iz U. Pokazaćemo kasnije da taj proces mora da se završi, proizvodeći terminalni sistem (onaj na koji se ne primenjuju pravila) u formi *false* ili Ø;S, gde je S sistem u rešenoj formi i predstavlja mgu za s i t.

Sistem U može simulirati akcije algoritma rekurzivnog spusta, i može se koristiti da se dokaže njegova korektnost. Ustvari, U se može posmatrati kao abstraktna verzija algoritma rekurzivnog spusta.

Neki tehnički rezultati u vezi *U*

lema2.2.1. Za bilo koji konačan skup jednakosti P, svaki niz transformacija iz U

$$P;\emptyset \Rightarrow P_1;S_1 \Rightarrow P_2;S_2 \Rightarrow \cdots$$

završava ili simbolom *false* ili u obliku Ø;S, gde je S u rešenoj formi.

dokaz: Definišimo meru složenosti $\langle n_1, n_2, n_3 \rangle$ na skupovima jednakosti, gde je:

n_1 = broj različitih promenljivih u P;
n_2 = broj simbola u P; i
n_3 = broj jednakosti u P forme $t =^? x$, gde t nije promenljiva.

Svako pravilo iz U smanjuje složenost problema P. Dalje, svaka jednakost mora odgovarati jednom od datih pravila sa leve strane, tako da pravilo uvek može da se primeni na sistem sa nepraznim P. Znači, sistem na koji se pravila ne mogu primeniti mora biti u obliku *false* ili Ø;S. Pošto kadgod se jednakost doda u S, promenljiva s leve strane se eliminiše iz ostatka sistema, pa svaki od sistema $S_1, S_2, \ldots, S$ mora biti u rešenoj formi.

Druga interesantna činjenica je da je rešenje σ proizvedeno pomoću *U* uvek idempotentno.

teorema2.2.1. Ako P;Ø $\Rightarrow$ Ø;S, tada je σ_S idempotentno.

Jedna od najinteresantnijih osobina sistema *U* je da njegova pravila ne menjaju skup unifikatora sistema.

lema2.2.2. Za bilo koju transformaciju $P;S \Rightarrow \Gamma$, substitucija θ unifikuje $P;S$ akko ona unifikuje Γ.

dokaz: Jedini netrivijalni slučajevi su u vezi provere pripadnosti (Occurs Check) i eliminacije promenljivih (Variable Elimination). Ako se *x* pojavljuje u *t*, ali nije jednako *t*, tada očigledno *x* sadrži manje simbola od *t*; ali tada $x\theta$ takođe sadrži manje simbola od $t\theta$, tako da *x* i *t* nemaju unifikator.

Razmatrajući Variable Elimination, znamo da $x\theta = t\theta$, odakle (strukturnom indukcijom) možemo pokazati da

$$u\theta = (u\{x \to t\})\theta$$

za bilo koji term *u*, ili ustvari za bilo koju jednakost ili skup jednakosti. Ali tada

$$P'\theta = P'\{x \to t\} \text{ i } S\theta = S\{x \to t\}\theta$$

odakle sledi rezultat.

Jedan od najvažnijih rezultata u vezi *U* je da *U* ustvari proizvodi unifikator.

teorema2.2.2. (Soundness): Ako P;Ø $\Rightarrow^+$ Ø;S, tada σ_S unifikuje svaku jednakost u P.

dokaz: Primetimo da σ_S, jer je idempotentna; prosta indukcija i prethodna lema pokazuju da σ_S mora unifikovati jednakosti iz P.

Drugi važan rezultat pokazuje da *U* pronalazi mgu za dva terma koja se mogu unifikovati.

teorema2.2.3. (kompletnost) Ako θ unifikuje svaku jednakost iz P, tada bilo koji maksimalan niz transformacija

$$P;\emptyset \Rightarrow \cdots$$

mora da se završi nekim sistemom Ø;S takvim da je σ_S opštija od θ.

dokaz: Prethodne leme pokazuju da takav niz mora da se završi nekim terminalnim sistemom Ø;S gde θ unifikuje S. Sada za svaku ugradnju $x \to t$ iz σ_S,

$$x\sigma_S\theta = t\theta = x\theta,$$

i za svaki $x \notin Dom(\sigma_S), x\sigma_S\theta = x\theta$, odakle sledi $\theta = \sigma_S\theta$.

Očigledna posledica prethodna dva rezultata je sledeća teorema:

teorema2.2.4. Ako P nema unifikatore, tada bilo koji maksimalni niz transformacija iz sistema P;Ø mora biti oblika

$$P;\emptyset \Rightarrow \cdots \Rightarrow \mathit{false}.$$

Zaključujemo da bilo koji algoritam unifikacije koji nastaje primenom formula iz U generiše idempotentan mgu za dva terma koja se mogu unifikovati. Neki delovi ovih osnovnih operacija mogu biti duži od drugih, ili formirati veće terme, i ne završavaju se svi nizov jednim istim mgu. U sledećem delu razmotrićemo ovo pitanje.

Neke osobine MGU

Iz prethodnih rezultata vidimo da bilo koja substitucija dobijena pomoću U može biti predstavljena kao kompozicija svih mogućih substitucija sa mgu. Ovo znači da se informacije ne gube u simboličkim računskim sistemima (kao npr. dokazivači teorema teorije prvog reda) pri rešavanju potproblema unifikacije i primeni rešenja na ostatak računa.

Sistem izvođenja U, polazeći od jednog para terma s i t, može proizvesti (konačno) mnogo različitih završnih formi, koje odgovaraju različitim najopštijim unifikatorima za s i t. Postavlja se pitanje koja je veza između različitih najopštijih unifikatora, da li postoji još najopštijih unifikatora osim ovih, i da li ih ima beskonačno mnogo? Odgovor na ova pitanja nalazi se u konceptu preimenovanja promenljivih: ako su σ i θ najopštiji unifikatori za s i t, tada $\sigma = \theta\rho$ za neko preimenovanje promenljivih ρ.

Ovo znači da skupovi najopštijih unifikatora za dva terma mogu biti generisani iz jednog mgu, pomoću kompozicije sa preimenovanjem promenljivih. Pomoću takve operacije, moguće je kreirati beskonačno mnogo idempotentnih najopštijih unifikatora i beskonačno mnogo neidempotentnih najopštijih unifikatora; pomoćukonačnog drveta pretraživanja generisanog pomoću *U* nije moguće konstruisati beskonačan broj najopštijih unifikatora.

Opšti zaključak ovog dela je da skup svih unifikatora dva terma ima netrivijalne osobine.

Složenost rekurzivnog spusta

Razmatranje složenosti algoritma rekurzivnog spusta je pitanje koje motiviše dalje razmatranje sofisticiranijih algoritama za unifikaciju. Pristupi unifikaciji do sada razmatrani imaju eksponencijalnu složenost.

primer2.2.1: Dati su sledeći termi:

$$h(x_1, x_2, \ldots x_n, f(y_0, y_0), \ldots, f(y_{n-1}, y_{n-1}), y_n)$$

i

$$h(f(x_0, x_0), f(x_1, x_1), \ldots, f(x_{n-1}, x_{n-1}), y_1, \ldots, y_n, x_n).$$

Unifikacija ova dva terma će proizvesti mgu gde je svaki x_i i svaki y_i vezan za term sa $2^{i+1} - 1$ simbola. Očigledan problem je što substitucija sadrži mnoge duplirane kopije istih podterma. Jedna ideja koja može da pomogne je da predstavimo substitucije na sledeći način:

$$[y_0 \to x_0; y_n \to f(y_{n-1}, y_{n-1}); y_{n-1} \to f(y_{n-2}, y_{n-2}); \cdots]$$

Građenje ovakve substitucije tokom unifikacije sastoji se iz prostog skupljanja liste ugradnji; ne kreiraju se duplicirani termi, i znači forma unifikatora ne može biti veća od originalnog problema.

Nažalost, ova ideja nije dovoljna da spasi algoritam, jer se ispostavlja da substitucija, i znači dupliranje terma, je neophodna u samim termima: u primeru, poziv *Unify* za poslednje argumente, x_n i y_n, koji su do tada vezani za terme sa $2^{n+1} - 1$ simbola, dovešće do eksponencijalnog broja rekurzivnih poziva. Rešenje ovog problema je pronaći pogodniju strukturu podataka za terme, i različite metode primene substitucije.

Unifikacija strukture term drveta

U daljem delu, razmotrićemo dva pristupa koja ubrzavaju proces unifikacije. Prvi pristup, koji su dali Corbin i Bidoit [1983], rešava problem dupliranja podterma kreiranih substitucijom koristeći graf prezentaciju terma koji mogu deliti strukturu; ovo rezultuje kvadratnim algoritmom. Da bismo otkrili asimptotski brži algoritam, neophodno je napustiti pristup rekurzivnim spustom, i svesti problem unifikacije na konstrukciju određenih relacija ekvivalencije na grafovima. Ovaj drugi pristup je otkrio Huet [1976].

Term drvo *s* i substitucija

Razmatrajući prethodni primer, treba naglasiti da se eksplozija veličine terma pojavljuje precizno jer postoje duplicirane pojave istih promenljivih, što uzrokuje dupliranje sve većih i većih terma. U cilju da otklonimo ovaj problem, neophodno je da razmotrimo detaljno kako predstaviti terme eksplicitno kao grafove koji dele podterme.

<u>definicija</u> Term drvo je direktan, acikličan graf čiji čvorovi su označeni funkcijskim simbolima, konstantama, ili promenljivim, i čije izlazni krajevi iz svakog čvora su uređeni, i gde izlazni stepen bilo kog čvora označenog simbolom f je jednak arnosti od f (promenljive imaju izlazni stepen 0).

U takvom grafu, svaki čvor ima prirodnu interpretaciju kao term, i poistovetićemo čvorove i terme kao da su ekvivalentni. Jedina razlika između različitih reprezentacija određenog terma je u strukturi deljenja među podtermima.

Pretpostavljajući da su imena simbola nizovi karaktera, moguće je kreirati drvo sa jedinstvenim, deljenim pojavama promenljivih, veličine O(n), gde je n broj svih karaktera u string reprezentaciji problema unifikacije. U normalnom slučaju, imena imaju konstantnu veličinu, i n predstavlja broj simbola u termu (dalje će ovo biti pretpostavljeno).

Znači, pretpostavimo da je ulaz u našem algoritmu term struktura koja predstavlja dva terma koja treba unifikovati, sa jedinstvenom, deljenom pojavom svih promenljivih. Takođe pretpostavljamo da svaki čvor t ima atribut *parents(t),* koji predstavlja listu svih roditelja čvora t u grafu (ekv. svih čvorova p koji pokazuju na t), ali ovo ne prikazujemo na dijagramu zbog jednostavnosti. Roditeljski pokazivači su neophodni kod deljenja čvorova.

Primena substitucije može biti implicitna ili eksplicitna; eksplicitna uključuje stvarno pomeranje podterm strelica. Na primer, neka su data dva terma $f(x, g(a))$ i

$f(g(y), g(y))$, i njihov mgu $\{x \to g(a), y \to a\}$. Implicitna primena substitucije identifikuje dva čvora povezana substitucionom strelicom, bez stvarnog pomeranja podterm veza. Ovo u primeru rezultuje formom $[x \to g(y); y \to a]$. Eksplicitna primena substitucije rezultuje ugradnjom promenljivih pomoću pomeranja podterm strelica koje pokazuju na promenljivu koja se ugrađuje. U datom primeru rezultat je substitucija $\{x \to g(a), y \to a\}$.

Rekurzivni spust na term strukturi *s*

Ako koristimo novu strukturu terma, substitucija sad ne duplicira terme. Međutim, opet postoji mogućnost da imamo duple pozive istog terma. Npr. u prethodnom primeru termi vezani za x_n i y_n biće unifikovani kad se x_0 veže za y_0; međutim, algoritam rekurzivnog spusta tada će istražiti svaku drugu stazu kroz parove terma, rezultujući eksponencijalnim brojem rekurzivnih poziva.

Očigledno, treba izbeći vraćanje na već rešene probleme u grafu. Najbolje rešenje je uvesti strukturu deljenja pomoću spajanja unifikovanih terma (koji su sada identični, i onda proveriti identičnost čvorova u prvom koraku. Spajanje dva čvora *s* i *t* u grafu Δ može se izvesti pomeranjem strelica. Neka je $parents(s) = \{p_1, \ldots, p_n\}$; tada

1.Za svaki p_i, zamenimo podterm strelicu $p_i \to s$ sa $p_i \to t$;
2.Neka $parents(t) := parents(s) \cup parents(t)$; i
3.Neka $parents(s) := \varnothing$.

Ovo deli strukturu za *t* i izoluje čvor *s*. U datom algoritmu, označićemo sa Replace(Δ, *s*,*t*) novi graf dobijen iz grafa Δ spajanjem *s* i *t* na opisani način.

Algoritam ima na ulazu term strukturu u kojoj su sve pojave promenljivih deljene, tj. svaka promenljiva se pojavljuje tačno jednom. Čak i sa ovim dodacima, algoritam rekurzivnog spusta je uglavnom nepromenjen:

```
global Δ : term struktura;   { Term struktura za s i t sa deljenim promenljivim }
global σ : lista parova čvorova;   { Inicijalizovana na praznu }

UnifyDag( s: node; t: node )
  begin
    if s and t are the same node then
          { Do nothing }
    else if s = f(s_1,...,s_n) and t = g(t_1,...,t_m) then begin
```

```
if f = g then
    for i:= 1 to n do
        UnifyDag(s_i, t_i);
  else Exit with failure   { Symbol clash }
 end
 else if s is not a variable then
   Unify( t,s );
 else if s occurs in t then
   Exit with failure;  { Occurs check }
 else
   Add (s,t) to the end of the list σ;
 Δ := Replace( Δ,s,t );  { Since they are now unified }
end;
```

Provera pripadnosti je implementirana pretragom da li je dati čvor s ispod t sledeći podterm strelice.

Tačnost strukture podataka za ovaj algoritam zavisi od sledećeg rezultata, koji može da se dokaže indukcijom.

lema2.2.3. Neka je Δ term struktura sa čvorovima x i t takva da ne postoji staza iz t u x.

· Replace(Δ,x,t) je acikličan graf koji sadrži iste čvorove (sa istim oznakama) kao Δ.

· Razmotrimo čvor u Δ koji odgovara termu s, i neka je s' term koji odgovara istom čvoru u Replace(Δ,x,t); tada:

- ako $s = x$, tada $s' = x$;

- u suprotnom, $s' = s\{x \to t\}$.

Razmotrimo složenost algoritma UnifyDag. Pošto svaki poziv ove funkcije izoluje čvor, ne može biti više od n poziva ukupno (gde je n broj simbola u originalnim termima). U svakom pozivu najsloženija je provera pripadnosti (proverava se ne više od n čvorova), i ne pomera se više od n čvorova. Označavanje liste roditelja ima takođe vremensku složenost $O(n)$. Originalna konstrukcija strukture ima složenost $O(n)$. Dakle, ukupna složenost algoritma je $O(n^2)$.

Skoro-linearan algoritam

Sada ćemo razmotriti alternativni pristup koji uvodi sledeće bitne promene u pristup do sada razmatran:

· umesto rekurzivnih poziva parova podterma koje treba unifikovati, svešćemo problem na konstruisanje relacije ekvivalencije čije klase su termi koje treba unifikovati;
· substitucija će (u izvesnom smislu) biti zamenjena unijom klasa ekvivalencije;
· ponovljeni pozivi koji proveravaju pripadnost biće zamenjeni jednim prolazom kroz graf da bi se proverila acikličnost.

Term struktura podataka se koristi za ove algoritme takođe, ali nećemo pomerati pokazivače kao u poslednjem delu. Umesto toga, razmotrićemo problem unifikacije kao problem koji uključuje sledeću relaciju na termima:

definicija Term relacija je relacija ekvivalencije na termima, homogena je ako nijedna klasa ekvivalencije ne sadrži $f(\ldots)$ i $g(\ldots)$ za $f \neq g$;ona je aciklična ako nijedan term nije ekvivalentan svom pravom podtermu.

Relacija unifikacije je homogena, aciklična term relacija koja zadovoljava aksiomu unifikacije: Za bilo koji f i terme s_i i t_i,

$$f(s_1,\ldots,s_n) \cong f(t_1,\ldots,t_n) \rightarrow s_1 \cong t_1 \wedge \Lambda \wedge s_n \cong t_n.$$

Unifikacijsko zatvorenje za s i t, kada postoji, je najmanja relacija unifikacije po kojoj su s i t ekvivalentni.

Algoritam predstavljen u ovom delu polazi od sledeće činjenice:

lema2.2.4. Ako se s i t mogu unifikovati, tada postoji unifikacijsko zatvorenje za s i t.

dokaz: Za bilo koji unifikator θ za s i t, definišimo relaciju:

$$u \cong_\theta v \text{ akko } u\theta = v\theta .$$

Očigledno je ovo relacija unifikacije. Pošto je presek dve relacije unifikacije koje se odnose na s i t opet relacija unifikacije koja se odnosi na s i t, kadgod se s i t mogu unifikovati postoji najmanja takva relacija $\cong$ koja spaja klase samo kad se primeni aksioma unifikacije na podterme od s i t.

Pristup pomoću unifikacijskog zatvorenja, prvi put predstavljen u [Huet 1976], pokušava da konstruiše ovu relaciju za dva terma, koja odgovara pronalaženju mgu. Uvedimo sledeće oznake:

def Za bilo koju term relaciju $\cong$, neka je šema funkcija funkcija ς iz klasa ekvivalencije u terme takva da za bilo koju klasu C,

1. $\varsigma(C) \in C$.

2. $\varsigma(C)$ je promenljiva samo ako se C sastoji samo od promenljivih.

Term $\varsigma(C)$ se zove šema term za C.

Treba naglasiti da je šema term u funkcionalnoj formi ako ona postoji, i koristi se da definiše substitucije. Primetimo da šema funkcije nisu jedinstvene, ali uvek postoji barem jedna za svaku term relaciju; pretpostavimo dalje da su te funkcije izabrane za svako dato unifikacijsko zatvorenje.

Primetimo da za svaku acikličnu term relaciju postoji parcijalno uređenje ϕ takvo da za bilo koji term $f(\ldots s \ldots)$, imamo $[f(\ldots s \ldots)]\, \phi\, [s]$.

<u>def</u> Za bilo koje unifikacijsko zatvorenje $\cong$, definišimo $\sigma_{\cong}$ pomoću:

$$x\sigma_{\cong} = \left\{ \begin{array}{c} y,\ ako\ \ \varsigma([x]) = y \\ f(s_1\sigma_{\cong},\ldots,s_n\sigma_{\cong}),\ ako\ \ \varsigma([x]) = f(s_1,\ldots,s_n) \end{array} \right\}.$$

<u>teorema2.2.5.</u> Termi s i t se mogu unifikovati akko postoji unifikacijsko zatvorenje za s i t. U potvrdnom slučaju, $\sigma_{\cong}$ je mgu za s i t.

Ovaj rezultat daje motiv za konstrukciju efikasnog algoritma za unifikaciju koji pokušava da napravi unifikacijsko zatvorenje za dva terma, i onda pronalazi mgu. Da bismo ovo uradili, neophodno je označiti klase ekvivalencije i primeniti aksiomu unifikacije na klase; najefikasnija struktura podataka pretstavlja klase kao drveta klasnih pokazivača, sa reprezentom na mestu korena drveta.

Da bismo odredili da li su dva terma ekvivalentna, treba samo naći korene drveta i proveriti identitet; da bismo spojili dve klase, jedna klasa je poddrvo korena druge. Da bismo smanjili dubinu drveta što je više moguće, uradićemo sledeće:

1.Označimo brojač veličine svake klase u reprezentu, i kad spajamo klase napravimo da manja bude poddrvo veće.
2.Kad sledimo staze do korena da označimo ekvivalenciju, kompresujemo staze tako da svi čvorovi pokazuju direktno na koren.

Term struktura za ovaj pristup ne treba roditelje pokazivače, kao u prethodnom algoritmu, već treba:

· klasne pokazivače

· brojač veličine klase izložen u reprezentu

· pokazivač iz svakog reprezenta na šema term za klasu

· bulove zastave visited i acyclic u svakom čvoru koje se koriste za proveru ciklusa (obe inicijalizovane na false)

· pokazivač vars iz svakog reprezenta na listu svih promenljivih u klasi (koristi se kad generišemo rešenja).

Primetimo da označavanje liste roditelja svakog čvora nije neophodno u ovom algoritmu. Reprezent je prosto čvor čiji klasni pokazivač pokazuje na sebe. U sledećem delu dat je algoritam zasnovan na ovom pristupu. Term struktura Δ za s i t je inicijalizovana na na relaciju identiteta, gde svaka klasa sadrži jedan term; znači za svaki čvor klasni i šema pokazivači su inicijalizovani da pokazuju na isti čvor, i veličina je inicijalizovana na 1. Lista promenljivih je inicijalizovana na praznu za ne-promenljive čvorove, i na listu od jednog elementa za čvorove koji pretstavljaju promenljive.

Ako Unify(s,t) ne propadne, onda σ sadrži rešenje. Find-Solution pokušava da nađe takvo rešenje, i pada akko postoji ciklus u grafu. Polja visited i acyclic su oba neophodna , prvo da pronađe ciklus u trenutno istraživanoj stazi, i drugo da spreči posećivanje čvorova koji su već isključeni iz svih mogućih ciklusa.

Korektnost ovog metoda zavisi od provere da on implementira tačno konstrukciju acikličnog unifikacijskog zatvorenja.

Bitne osobine ovde su sledeće:

· ekvivalencija je očigledno homogena

· klase ekvivalencije se spajaju akko to zahteva aksioma unifikacije

· FindSolution pada akko postoji ciklus u grafu

· Kadgod se ugradnja $[x \rightarrow s]$ doda u σ, sve odgovarajuće ugradnje za promenljive u s već se pojavljuju u σ.

Sada možemo da pretstavimo algoritam:

```
global Δ : termDag;   { Term struktura za s i t sa deljenim promenljivim }
global σ : list of bindings := nil;  { Sadrži rešenje u odgovarajućoj formi }

Unify( s: node; t: node )
  begin
    UnifClosure(s,t);
    FindSolution(s);
  end;
```

```
UnifClosure( s: node; t: node)
  begin
    s := Find(s);
    t := Find(t);
    if s  and  t are the same node  then
         { Ne radi ništa }
    else  begin
      if  ς([s]) = f(s_1,...,s_n)  and  ς([t]) = g(t_1,...,t_m)  for n,m ≥ 0
      then  begin
        if f = g  then begin
           Union(s,t);
           for  i := 1 to n  do
                UnifClosure(s_i, t_i);
        end
        else  Exit with failure     { Nepoklapanje simbola }
      end
      else  Union(s,t);
    end;
  end;

Union( s: node; t: node )   { s i t su reprezenti }
  begin
    if  size(s) ≥ size(t)  then  begin
      size(s) := size(s) + size(t);
      vars(s) := concatenation of lists vars(s) and vars(t);
      if  ς([s])  is a variable  then
          ς([s]) := ς([t]);
      class(t) := s;
    end
    else begin
      size(t) := size(t) + size(s);
      vars(t) := concatenation of lists vars(t) and vars(s);
         if  ς([t])  is a variable  then
             ς([t]) := ς([s]);
         class(s) := t;
    end;
  end;

Find ( s: node )    { Vraća reprezent za [s] i kompresuje staze }
  t : node;
  begin
    if  class(s) = s    { s je reprezent }  then
        Return  s;
```

```
    else begin
      t := Find(class(s));
      class(s) := t;
      return t;
    end;
  end;

FindSolution ( s : node );   { Pada ako postoji ciklus u s }
  begin
    s := ς(Find(s));
    if acyclic(s) then
      Return;  { s nije deo ciklusa }
    if visited(s) then
      Fail;  { Postoji ciklus }
    if s = f(s_1,...,s_n) for some n > 0 then begin
      visited(s) := true;
      for i := 1 to n do
          FindSolution(s_i);
      visited(s) := false;
    end;
    acyclic(s) := true;
    foreach x ∈ vars(Find(s)) do
      if x ≠ s then
        Add [x→s] to front of σ;
  end;
```

Složenost algoritma je $O(n\alpha(n))$. Sve procedure, osim procedure Find, pozivaju se najviše n puta za terme sa n simbola. Primetimo da spajanje lista promenljivih može biti izvršeno tako što se čuvaju pokazivači na poslednji član liste, i spajanje se izvrši pomeranjem pokazivača umesto da vršimo standardno spajanje (složenost opisanog postupka je $O(n)$). Jedino je složenost funkcije Find je $O(n\alpha(n))$.

Analizirajmo sada ovaj algoritam.

Na samom početku svaki čvor pokazuje na samog sebe (odnosno njegov klasni pokazivač je pokazivač na sam taj čvor). Prva dva koraka na početku algoritma unifikacije su:

$s :=$ Find(s);
$t :=$ Find(t);

Ova dva koraka su potpuno suvišna jer svaki čvor pokazuje na sebe, pa se funkcije Find(s) i Find(t) nikad neće izvršiti (tačnije uvek vraćaju unesene vrednosti s i t).

Dalje, ukoliko su s i t funkcijski simboli, procedura UnifClosure se primenjuje rekurzivno na njihove potomke (redom po odgovarajućim parovima). Za potomke važi isto: svi oni pokazuju na samog sebe, pa je funkcija Find opet nepotrebna, itd.

Možemo zaključiti da je funkcija Find nepotrebna u algoritmu i možemo je potpuno izbaciti i dobiti malo jednostavniji algoritam. Osim što dobijamo na uprošćavanju algoritma, važnije je što smo ovim eliminisanjem smanjili složenost algoritma, koja sad iznosi $O(n)$.

Razmotrimo primenu algoritma (opisujući najvažnije korake) na jednom konkretnom primeru.

Neka su zadata dva terma:

$f(x, g(a))$ i $f(g(y), g(y))$, koja treba unifikovati.

Oba terma, ako se predstave u obliku drveta, imaju na vrhu funkcijske simbole. Ako ih uporedimo, vidimo da su jednaki (f); primenom algoritma odmah se izvrši njihova unija (oni sada pripadaju istoj klasi). Označimo ove polazne terme slovima s i t.

Osnovni koraci primene algoritma UnifClosure na polazne terme s i t su sledeći:

1. Union(s, t)
2. UnifClosure(x, $g(y)$)
3. UnifClosure($g(a)$, $g(y)$).

Koraci 2. i 3. predstavljaju rekurzivne pozive procedure UnifClosure iz iste procedure primenjene na polazne terme s i t (funkcija je primenjena na potomke terme s i t.

U koraku 2. se izvrši spajanje (unija) čvorova x i $g(y)$, i pošto je x promenljiva time je u potpunosti završen korak 2. (odnosno u njemu se procedura UnifClosure više ne poziva).

U koraku 3. izvrši se unija čvorova $g(a)$ i $g(y)$, i onda se procedura UnifClosure ponovo rekurzivno pozove i izvrši se unija njihovih potomaka: a i y. Time je korak 3. završen. Dakle, potkoraci u koraku 3. su:

a) Union($g(a)$, $g(y)$)
b) Union(a, y).

Spajanja čvorova koja su bitna za samu unifikaciju su ustvari spajanja parova u kojima je barem jedan član promenljiva. Ovim spajanjem se ustvari promenljiva poveže sa termom u koji će se kasnije ugraditi. Preciznije, konkretno u našem primeru: pri uniji čorova x i $g(y)$, u skup promenljivih terma $g(y)$ se doda promenljiva x.

Dakle, globalan zaključak je sledeći:

Algoritam unifikacije se sastoji od dve procedure: UnifClosure i FindSolution, koje se redom pozivaju. Procedura UnifClosure formira sve odgovarajuće klase sa njihovim elementima, i onda se na taj rezultat primeni procedura FindSolution. Procedura

FindSolution pokupi redom prethodno dobijene rezultate i formira rezultat unifikacije σ, ukoliko nema ciklusa u drvetu, odnosno pada ukoliko u drvetu postoji ciklus.

Algoritam je (uz neke manje izmene), implementiran u sistemu Matematica, i testiran na nekim primerima. Vidimo da je rezultat programa predstavljen u obliku liste, pri čemu je promenljiva predstavljena u vitičastim zagradama, a neposredno iza nje je term u koji se ona ugrađuje (ukoliko se termi mogu unifikovati). Ukoliko se termi ne mogu unifikovati, program izbaci Neuspeh.

```
s=f[x,g[a]];
t=f[g[y],g[y]];
p[a_]:=(If[MemberQ[{x,y,z},a],1,0]);
st=s//TreeForm;
tt=t//TreeForm;
dubina1=Depth[st]-1;
dubina2=Depth[tt]-1;
pom1=2^(dubina1-1);
pom2=2^(dubina2-1);

//Prvo se formiraju trodimenzioni nizovi a i b, koji sadrže podatke za ulazne terme s i t
//prva dimenzija predstavlja nivo u drvetu, druga redni broj na tom nivou, i za svaki
//čvor imamo četiri pod: sam term, pokazivač na represent klase, broj elemenata klase,
// i listu promenljivih

Array[a,{dubina1,pom1,4}];
Array[b,{dubina2,pom2,4}];
a[1,1,1]=s;
Array[broj1,dubina1];
Array[broj,dubina1];
Do[broj[i]=2^(i-1),{i,dubina1}];
Do[broj1[i]=0,{i,dubina1}];

//Funkcije Formnivoa, Formnivob formiraju za polazne terme s i t sve elemente nizo-
//va a i b čija je treća dimenzija jedan, odnosno sve terme i njihove podterme. Ako ne-
//ki element nema levi ili desni potomak, na odgovarajuće prazno mesto se upiše nula

Formnivoa[nivo_]:=
            Module[{s2,s3,l,pom},
                    Do[s2=a[nivo-1,i,1];
                    If[!(!(s2□0)&&!(Depth[s2]□1)),

Do[broj1[nivo]++;pom=broj1[nivo];a[nivo,pom,1]=0,{2}]];
                            If[Depth[s2]≥2,s3=s2//TreeForm;
                                l=Level[s3,{2}];
                                broj1[nivo]++;pom=broj1[nivo];
                                a[nivo,pom,1]=First[l];
                                l=Rest[l];broj1[nivo]++;
                                pom=broj1[nivo];
                                If[l≠{},a[nivo,pom,1]=First[l]];
                                If[l□{},a[nivo,pom,1]=0]],{i,broj[nivo-1]}]]

Do[Formnivoa[j],{j,2,dubina1,1}];
b[1,1,1]=t;
```

```
Array[br1,dubina2];
Array[br,dubina2];
Do[br[i]=2^(i-1),{i,dubina2}];
Do[br1[i]=0,{i,dubina2}];

Formnivob[nivo_]:=
                    Module[{t2,t3,l,pom},
                                   Do[t2=b[nivo-1,i,1];
                                   If[!(!(t2□0)&&!(Depth[t2]□1)),
Do[br1[nivo]++;pom=br1[nivo];b[nivo,pom,1]=0,{2}]];
                                          If[Depth[t2]≥2,t3=t2//TreeForm;
                                                l=Level[t3,{2}]
                                                br1[nivo]++;pom=br1[nivo];
                                                b[nivo,pom,1]=First[l];
                                                l=Rest[l];br1[nivo]++;
                                                pom=br1[nivo];
                                                If[l≠{},b[nivo,pom,1]=First[l]];
                                                If[l□{},b[nivo,pom,1]=0]],{i,br[nivo-1]}]]

Do[Formnivob[j],{j,2,dubina1,1}];

//Sada se popune još tri preostala mesta za nizove a i b: klasni pokazivač, broj eleme-
//nata klase i lista promenljivih

Do[Do[If[a[i,j,1]=!=0,p1=p[a[i,j,1]]];
               a[i,j,2]={a,i,j};
               a[i,j,3]=1;
               If[p1□1,a[i,j,4]={a[i,j,1]}];
               If[p1□0,a[i,j,4]={}],{j,broj[i]}],{i,dubina1}]
Do[Do[If[b[i,j,1]=!=0,p1=p[b[i,j,1]]];
                 b[i,j,2]={b,i,j};
                 b[i,j,3]=1;
                 If[p1□1,b[i,j,4]={b[i,j,1]}];
                 If[p1□0,b[i,j,4]={}],{j,br[i]}],{i,dubina2}];

Spajanje[n1_,b1_]:=(p1=p[a[n1,b1,1]];
                              Which[p1□1,
                                        b[n1,b1,3]=b[n1,b1,3]+1;
                                        b[n1,b1,4]=Union[a[n1,b1,4],b[n1,b1,4]];
                                        a[n1,b1,2]={b,n1,b1},
                                        p1□0,
                                        a[n1,b1,3]=a[n1,b1,3]+1;
                                        a[n1,b1,4]=Union[a[n1,b1,4],b[n1,b1,4]];
                                        b[n1,b1,2]={a,n1,b1}])
```

```
//Ovo je suštinska funkcija u programu, koja vrši upoređivanje terma redom po
//parovima, i ako je moguća unifikacija terma na određenom nivou, izvrši se spajanje
//odgovarajućih parova, redom

UnifClosure[n_,redbr_]:=
        Module[{s1,t1},
                s1=a[n,redbr,1];t1=b[n,redbr,1];
                If[(s1=!=0)&&(t1=!=0)&&(s1=!=t1),
                  If[(p[s1]□0)&&(p[t1]□0),

Which[(Depth[s1]≥2)&&(Depth[t1]≥2)&&(Head[s1]===Head[t1]),
                                Spajanje[n,redbr],

!((Depth[s1]≥2)&&(Depth[t1]≥2)&&(Head[s1]===Head[t1])),
                                Print[Neuspeh]]];
                If[!((p[s1]□0)&&(p[t1]□0)),
                    Spajanje[n,redbr]]]

Do[Do[UnifClosure[i,j],{j,broj[i]}],{i,dubina1}];

//Rezultat unifikacije je pretstavljen listom Sigma, koja je na početku inicijalizovana
//na praznu listu

Sigma={};
DodajEl1[i1_,j1_]:=
      Module[ {l,p1},
              If[a[i1,j1,1]=!=0,l=a[i1,j1,4];
                p1=p[a[i1,j1,1]];

If[(p1□0)&&(l≠{}),Sigma=Join[Sigma,{a[i1,j1,4],a[i1,j1,1]}]]]]
Do[Do[DodajEl1[i,j],{j,broj[i]}],{i,dubina1}];
DodajEl2[i2_,j2_]:=
      Module[{l,p1},
              If[b[i2,j2,1]=!=0,l=b[i2,j2,4];
                p1=p[b[i2,j2,1]];

If[(p1□0)&&(l≠{}),Sigma=Join[Sigma,{b[i2,j2,4],b[i2,j2,1]}]]]]
Do[Do[DodajEl2[i,j],{j,broj[i]}],{i,dubina2}];
Sigma

{{y},a,{x},g[y]}

//Dalje je testiran drugi primer, sa polaznim termima s i t koji se ne mogu unifikovati
```

s=f[b,g[a]];
t=f[c,g[y]];

Neuspeh

komentar: U drugom testiranom primeru smo samo zamenili početne vrednosti terma s i t (koji se u ovom slučaju ne mogu unifikovati) i dobili na izlazu Neuspeh.

Primena teorije unifikacije:

Teorija unifikacije je jako važan pojam u matematičkoj logici. Ima primenu u više oblasti, jedna od njih je eliminacija kvantifikatora. U zavisnosti od same formule, u nekim slučajevima se samom unifikacijom neke kvantifikovane promenljive ugrade u terme, pa je automatski eliminacija kvantifikatora pojednostavljena (smanjen je broj kvantifikatora u formuli, odnosno smanjen je broj promenljivih koje treba eliminisati- u nekim slučajevima može se dobiti formula bez kvantifikatora). Dalje, u nekim slučajevima da bismo uopšte mogli eliminisati kvantifikatore, neophodno je prethodno izvršiti unifikaciju terma.

3. Eliminacija kvantifikatora

3.1.Uvod

def Pretpostavimo da je zadat jezik L i skup A formula u L. Skup A dozvoljava eliminaciju kvantifikatora u formuli F jezika L ako postoji formula bez kvantifikatora F′ u L takva da $F \Leftrightarrow F'$ je posledica od A. Skup A dozvoljava eliminaciju kvantifikatora u L ako A dozvoljava eliminaciju kvantifikatora u svakoj formuli jezika L.

Poznato je u iskaznoj logici, da svaka formula bez kvantifikatora B je ekvivalentna formuli forme $B_1 \vee B_2 \vee \cdots \vee B_k$, gde svaka formula B_i je forme $\alpha_1 \wedge \cdots \wedge \alpha_r$, i svaki α_j je atomična formula jezika L ili negacija atomične formule u L. Takođe, pošto je formula $\exists x(B_1 \vee \cdots \vee B_k)$ ekvivalentna formuli $\exists x B_1 \vee \cdots \vee \exists x B_k$, sledi sledeća teorema:

teorema3.1.1. Skup A formula jezika L dozvoljava eliminaciju kvantifikatora u L akko A dozvoljava eliminaciju kvantifikatora u svim formulama forme: $\exists x(\alpha_1 \wedge \cdots \alpha_k)$, gde svaki α_i je atomična formula ili negacija atomične formule jezika L.

Možemo i na drugi način (bez korišćenja DNF), pokazati da se eliminacija kvantifikatora u nekoj teoriji T svodi na eliminaciju kvantifikatora u formulama oblika: $\exists x \varphi$, gde je φ formula bez kvantifikatora.

Da bismo pojednostavili oznake, umesto $T \vdash \varphi \Leftrightarrow \psi$ pisaćemo $\varphi \approx \psi$.

Dakle, pretpostavimo da možemo efektivno eliminisati egzistencijalni kvantifikator iz svake formule oblika : $\exists x \varphi$, gde je φ formula bez kvantifikatora. Rekurzivna procedura za eliminaciju kvantifikatora je opisana na sledeći način:

Ulaz: formula φ
Izlaz: formula ψ bez kvantifikatora takva da je $\varphi \approx \psi$

• Ako φ ima oblik $\varphi_1 * \varphi_2$, gde je $*$ jedna od operacija $\wedge, \vee, \Rightarrow$ i $\Leftrightarrow$, tada prvo pronalazimo formule bez kvantifikatora ψ_1 i ψ_2 takve da je $\varphi_i \approx \psi_i$, tada je $\psi = \psi_1 * \psi_2$ $(1 \le i \le 2)$.

• Ako φ ima oblik $\neg\varphi_1$, tada prvo pronalazimo formulu bez kvantifikatora ψ_1 takvu da je $\varphi_1 \approx \psi_1$, pa sledi $\psi = \neg\psi_1$.

• Ako φ ima oblik $\exists x \varphi_1$, tada prvo pronalazimo formulu bez kvantifikatora ψ_1 takvu da je $\varphi_1 \approx \psi_1$, pa onda pronalazimo formulu bez kvantifikatora ψ takvu da je: $\exists x \psi_1 \approx \psi$.

• Ako φ ima oblik $\forall x \varphi_1$, tada nastavljamo kao što je opisano u prethodna dva slučaja za formulu $\neg \exists x \neg \varphi_1$.
(Ovim je algoritam, odnosno naš dokaz, završen).

Algoritam je implementiran u sistemu Matematica, i testiran na nekoliko primera.

```
//Funkcija Acf proverava da li data formula fi pripada algebarski zatvorenim poljima
//Funkcija Acf prvo proverava koja je vrhovna operacija, i ukoliko je to neka od logi-
//čkih operacija, funkcija Acf se pozove rekurzivno za podformule, i ako podformule
//pripadaju Acf, vraća kao rezultat True
//Ukoliko je vrhovna operacija jednako, proverava se da li su podformule polinomi
//(funkcijom PolynomialQ u Matematici)

Acf[fi_]:=Module[{p1,p2,r1,r2,rez },
              Which[(Head[fi]===And)||(Head[fi]===Or)
                        ||(Head[fi]===Implies),
                   p1=fi[[1]];
                   p2=fi[[2]];
                   r1=Acf[p1];
                   r2=Acf[p2];
                   rez=r1&&r2;
                   Return[rez],
                   Head[fi]===Not,
                   p1=fi[[1]];
                   rez=Acf[p1];
                   Return[rez],
                   (Head[fi]===Exists)||(Head[fi]===ForAll),
                   p2=fi[[2]];
                   rez=Acf[p2];
                   Return[rez],
                   Head[fi]===Equal,
                   p1=fi[[1]];
                   p2=fi[[2]];
                   r1=PolynomialQ[p1];
                   r2=PolynomialQ[p2];
                   rez=r1&&r2;
                   Return[rez]]]

//Funkcija Rcf proverava da li data formula fi pripada realnim zatvorenim poljima
```

//Funkcija Rcf prvo proverava koja je vrhovna operacija, i ukoliko je to neka od logi-
//čkih operacija, funkcija Rcf se pozove rekurzivno za podformule, i ako podformule
//pripadaju Rcf, vraća kao rezultat True
//Ukoliko je vrhovna operacija jednako,manje, ili veće, proverava se da li su podfor-
//mule polinomi (funkcijom PolynomialQ u Matematici)

```
Rcf[fi_]:=Module[{p1,p2,r1,r2,rez },
        Which[(Head[fi]===And)||(Head[fi]===Or)
                ||(Head[fi]===Implies),
              p1=fi[[1]];
              p2=fi[[2]];
              r1=Rcf[p1];
              r2=Rcf[p2];
              rez=r1&&r2;
              Return[rez],
              Head[fi]===Not,
              p1=fi[[1]];
              rez=Rcf[p1];
              Return[rez],
              (Head[fi]===Exists)||(Head[fi]===ForAll),
              p2=fi[[2]];
              rez=Rcf[p2];
              Return[rez],
              ( Head[fi]===Equal)||(Head[fi]===Greater)
                ||(Head[fi]===Less),
              p1=fi[[1]];
              p2=fi[[2]];
              r1=PolynomialQ[p1];
              r2=PolynomialQ[p2];
              rez=r1&&r2;
              Return[rez]]]
```

//Funkcija Eliminacija vrši eliminaciju kvantifikatora u datoj formuli fi, ukoliko je
//eliminacija moguća, i vraća kao rezultat formulu bez kvantifikatora ekvivalentnu
//formuli fi
//Funkcija eliminacija se izvršava rekurzivno, prvo se proveri koja je operacija u vrhu
//drveta (ako formulu fi predstavimo u obliku drveta), zatim se rekurzivno izvrši eli-
//minacija kvantifikatora u levoj i desnoj podformuli, i na dobijene međurezultate se
//primeni vrhovna operacija, što predstavlja rezultat
//Ukoliko se u formuli pojavi univerzalni kvantifikator, on se ekvivalentnim transfor-
//macijama prevede u egzistencijalni, i onda se dalje poziva funkcija Eliminacija

```
Eliminacija[fi_]:=Module[{p1,p2,r1,r2,rez,p},
                           p1=fi[[1]];
                           p2=fi[[2]];
                        Which[Head[fi]===Or,
                           r1=Eliminacija[p1];
                           r2=Eliminacija[p2];
                            rez=r1|| r2;
                            Return[rez],
                        Head[fi]===And,
                           r1=Eliminacija[p1];
                           r2=Eliminacija[p2];
                            rez=r1&&r2;
                            Return[rez],
                        Head[fi]===Implies,
                          rez=Eliminacija[p1];
                          rez=!rez;
                          p=Eliminacija[p2];
                          rez=rez||p;
                          Return[rez],
                        Head[fi]===Not,
                          rez=Eliminacija[p1];
                          rez=!rez;
                          Return[rez],
                        Head[fi]===ForAll,
                          p2=!p2;
                          rez=Apply[Exists,{p1,p2}];
                          p=Eliminacija[rez];
                          rez=!p;
                          Return[rez],
                        Head[fi]===Exists,
                          rez=Resolve[fi];
                          Return[rez]]];
```

```
//funkcije su testirane na nekoliko primera
//U prvom primeru prvo proverimo da li formula f pripada rcf, i ako pripada to se
//odštampa

f=And[ForAll[x,x+3>2],Exists[x,x^2<0]];
If[Rcf[f]□True,Print[ formula rcf],Print[nije formula rcf]]
formula rcf

//Dalje se za istu formulu poziva funkcija Eliminacija, i štampa se njen rezultat
Eliminacija[f]
False
```

```
//Dalje je testirana funkcija Eliminacija za neke druge formule f

f=Implies[Exists[x,x+3>2],Exists[x,x^2<0]];
Eliminacija[f]
False

f=Or[Exists[x,x+3>2],Exists[x,x^2<0]];
Eliminacija[f]
True

a=Exists[x,x+3>2&&x+y>0];
Eliminacija[a]
y∈Reals

//Na kraju je testirana egzistencija preseka zadatog hiperboličkog paraboloida i zadate
//sfere

a=Exists[x,And[y^2-2yz+x-2y+3z==0,x^2+y^2+z^2-10x+2y+10□0]];
b=Eliminacija[a];
c=Apply[Exists,{y,b}];
Eliminacija[c]
True
```

Još jedan način eliminacije kvantifikatora je način zasnovan na ekvivalentnim logičkim transformacijama.

Koriste se sledeće tautologije, pri čemu promenljiva x nema slobodnih pojava u formuli φ:

1. $\varphi \vee \forall x \psi(x) \Leftrightarrow \forall x(\varphi \vee \psi(x))$
2. $\varphi \vee \exists x \psi(x) \Leftrightarrow \exists x(\varphi \vee \psi(x))$
3. $\varphi \wedge \forall x \psi(x) \Leftrightarrow \forall x(\varphi \wedge \psi(x))$
4. $\varphi \wedge \exists x \psi(x) \Leftrightarrow \exists x(\varphi \wedge \psi(x))$
5. $\varphi \Rightarrow \forall x \psi(x) \Leftrightarrow \forall x(\varphi \Rightarrow \psi(x))$
6. $\varphi \Rightarrow \exists x \psi(x) \Leftrightarrow \exists x(\varphi \Rightarrow \psi(x))$
7. $\forall x \psi(x) \Rightarrow \varphi \Leftrightarrow \exists x(\psi(x) \Rightarrow \varphi)$
8. $\exists x \psi(x) \Rightarrow \varphi \Leftrightarrow \forall x(\psi(x) \Rightarrow \varphi)$
9. $\neg \forall x \psi(x) \Leftrightarrow \exists x \neg \psi(x)$
10. $\neg \exists x \psi(x) \Leftrightarrow \forall x \neg \psi(x)$
11. $\forall x \forall y F(x,y) \Leftrightarrow \forall y \forall x F(x,y)$
12. $\exists x \exists y F(x,y) \Leftrightarrow \exists y \exists x F(x,y)$

Ako primenimo navedene formule sleva udesno da bismo transformisali zadatu formulu vidimo da se kvantifikatori pomeraju ispred podformula (pomeraju se ulevo).

def Teorija A jezika L je kompletna ako za svaku zatvorenu formulu F u L, ili F ili $\neg F$ je posledica od A.

Za svaku teoriju A prirodno se postavlja pitanje njene odlučivosti, odnosno egzistencije algoritma koji za datu $\phi \in Sent_L$ daje odgovor da li je A $\vdash \phi$ tačno. U slučaju rekurzivne kompletne teorije rekurzivnog jezika, odgovor je potvrdan.

Neka je **M** L- struktura (model jezika L). $X \subseteq M^n$ se može definisati akko postoji L-formula $\phi(x_1,...,x_n,y_1,...,y_n)$ i $\bar{b} \in M^n$ tako da je:

$$X = \{ \bar{a} \in M^n : \mathbf{M} \models \phi(\bar{a},\bar{b}) \} .$$

Kažemo da $\phi(\bar{x},\bar{b})$ definiše X. Proučavanje skupova koji se mogu definisati je dosta otežano zbog kvantifikatora koji se mogu pojaviti u definiciji formula.
U teorijama koje dopuštaju eliminaciju kvantifikatora svaki skup koji se može definisati, može se definisati pomoću Bulove kombinacije atomičnih formula, i njihove osobine je lakše proučavati.

Dovoljan uslov za eliminaciju kvantifikatora

Za ovaj rezultat je zaslužan Robinson:

teorema3.1.2. Neka jezik L sadrži konstantni simbol, T je L-teorija i $\phi(x) \in For_L$. Sledeći uslovi su ekvivalentni:

1. Postoji $\psi(\bar{x}) \in For_L$ bez kvantifikatora koja je T-ekvivalentna formuli ϕ (ekv. T $\vdash \phi \Leftrightarrow \psi$).

2. Ako je **M**, **N** $\models$ T i **A** je L- podstruktura modela **M** i **N**, tada za svako $\bar{a} \in A$, **M** $\models \phi(\bar{a})$ akko je **N** $\models \phi(\bar{a})$.

Očigledno je da L- teorija T dopušta eliminaciju kvantifikatora akko za svaku L-formulu oblika $\exists y \phi(\bar{x},y)$, gde je ϕ Bulova kombinacija atomičnih formula, postoji T-ekvivalentna formula bez kvantifikatora $\psi(\bar{x})$.
Očigledna posledica teoreme3.1.2. i gornje konstatacije je sledeća lema:

lema3.1.1. Pretpostavimo da je T L-teorija i za sve formule bez kvantifikatora

$\phi(\bar{x},y)$, ako su **M** i **N** modeli teorije T i **A** je podmodel modela **M** i **N**, $\bar{a}\in A$ i ako je:
M $\models \exists y\phi(\bar{a},y)$, onda važi: **N** $\models \exists y\phi(\bar{a},y)$. Onda T dopušta eliminaciju kvantifikatora.

lema3.1.2. Pretpostavimo da teorija T dopušta eliminaciju kvantifikatora, i da postoji **N** $\models$ T koji se "1-1" preslikava u svaki model teorije T. Tada je T kompletna teorija.

dokaz: Neka je **M** $\models$ T, $\phi\in Sent_L$ i ψ je rečenica bez kvantifikatora koja je T-ekvivalentna ϕ. Pošto je **N** $\subseteq$ **M**, važi sledeće:

$$\mathbf{M}\models\phi \text{ akko } \mathbf{M}\models\psi \text{ akko } \mathbf{N}\models\psi \text{ akko } \mathbf{N}\models\phi.$$

Sada možemo zaključiti da je teorija T kompletna, jer za svaki model teorije T skup rečenica koje važe u tom modelu jednak je skupu rečenica koje važe u modelu **N**.

lema3.1.3. Pretpostavimo da je T odlučiva teorija koja dozvoljava eliminaciju kvantifikatora. Tada postoji algoritam koji za datu formulu ϕ pronalazi T-ekvivalentnu formulu ψ bez kvantifikatora.

dokaz: Neka ϕ ima n slobodnih promenljivih i neka je $(\psi_i)_{i\in N}$ efektivno nabrajanje svih formula jezika L koje imaju n slobodnih promenljivih. Pošto je teorija T odlučiva, postoji algoritam koji pronalazi da li je $\vdash\phi\Leftrightarrow\psi_1$. Ako nije $\vdash\phi\Leftrightarrow\psi_1$, nastavljamo dalje za ψ_2 itd. Opisani postupak se mora završiti jer T dopušta eliminaciju kvantifikatora.

Dalje ćemo razmotriti eliminaciju kvantifikatora za neke konkretne teorije.

3.2.Teorija gustog uređenja sa prvim i zadnjim elementom

Razmotrimo jezik L koji ima dva konstantna simbola 0,1 i dva binarna relacijska simbola $<$, $=$.
Neka je A skup sledećih formula jezika L:

$\forall x\neg(x<x)$
$\forall x\forall y\forall z(x<y\wedge y<z\Rightarrow x<z)$
$\forall x\forall y(x=y\vee x<y\vee y<x)$
$\forall x\forall y\exists z(x<y\Rightarrow x<z\wedge z<y)$
$\forall x(x=0\vee 0<x)$
$\forall x(x=1\vee x<1)$

Pokazaćemo da A dozvoljava eliminaciju kvantifikatora u L.

Pretpostavimo da je data formula oblika $\exists x(\alpha_1 \wedge \cdots \wedge \alpha_k)$, gde svaki α_i je ili atomična formula u L ili negacija atomične formule. Dakle, svaki α_i ima jedan od oblika: $t_1 < t_2, t_1 = t_2, \neg(t_1 < t_2), t_1 \neq t_2$, gde su t_1, t_2 termi jezika L, znači ili 0,1 ili promenljiva.

Iz skupa A sledi da je $\neg(t_1 < t_2)$ ekvivalentno $(t_2 < t_1) \vee (t_1 = t_2)$, i $t_1 \neq t_2$ je ekvivalentno $(t_1 < t_2) \vee (t_2 < t_1)$. Koristeći činjenicu da je $A \wedge (B \vee C)$ ekvivalentno: $(A \wedge B) \vee (A \wedge C)$, $\exists x(A \vee B)$ ekvivalentno $\exists x A \vee \exists x B$, sveli smo problem eliminacije kvantifikatora na eliminaciju kvantifikatora u formulama oblika:

$$\exists x(\alpha_1 \wedge \cdots \wedge \alpha_r),$$

gde svaki α_i ima oblik: $t_1 = t_2$ ili $t_1 < t_2$.

Nastavljamo dokaz rekurzijom po r. Ako r = 1, formula je $\exists x(t_1 < t_2)$ ili $\exists x(t_1 = t_2)$, gde su t_1, t_2 ili 0,1 ili promenljiva. Eliminacija kvantifikatora u ovom slučaju je očigledna.

Sada pretpostavimo da smo eliminisali kvantifikatore u svim formulama gde je r <n, i razmotrimo formulu $\exists x(\alpha_1 \wedge \cdots \wedge \alpha_n)$. Ako jedan od α_i, recimo α_1, ne sadrži *x*, formula je ekvivalentna $\alpha_1 \wedge \exists x(\alpha_2 \wedge \cdots \alpha_n)$, i svedena je na slučaj r = n – 1. Dalje, pretpostavimo da svi α_i sadrže *x*, tako da formulu možemo napisati u sledećem obliku:

$$\exists x(x < t_1 \wedge \cdots x < t_k \wedge u_1 < x \wedge \cdots u_l < x \wedge x = v_1 \wedge \cdots \wedge x = v_m),$$

gde su *t*,*u*,*v* termi različiti od *x* (ako je, na primer, $t_1 = x$, formula je ekvivalentna *false*).

Ako je $k > 1$ formula je ekvivalentna sledećoj formuli:

$$(t_1 < t_2 \wedge \exists x(x < t_1 \wedge x < t_3 ...)) \vee (\neg(t_1 < t_2) \wedge \exists x(x < t_2 \wedge x < t_3 ...)),$$

i ponovo smo sveli formulu na slučaj $r = n - 1$.

Dolazimo do sličnog zaključka ako je $l > 1$.

Ako je $k = l = 1$, formula može biti zapisana na sledeći način:

$$\exists x(x < t_1 \wedge u_1 < x \wedge x = v_1 \wedge \cdots \wedge x = v_m).$$

-za $m \neq 0$ formula je ekvivalentna:

$(v_1 = v_2 = \cdots = v_m) \wedge (u_1 < v_1 < t_1)$

-za $m = 0$ formula je ekvivalentna:

$u_1 < t_1$.

Za $k = 0$ formula može biti zapisana:

$\exists x(u_1 < x \wedge x = v_1 \wedge \cdots \wedge x = v_m)$,

što je za $m \neq 0$ ekvivalentno sa:

$(u_1 < v_1) \wedge (v_1 = v_2 = \cdots = v_m)$,

i za $m = 0$ ekvivalentno $u_1 \neq 1$.

Dobijamo sličan rezultat kad je $l = 0$, što kompletira dokaz.

Na sličan način mogu se istraživati teorije gustog uređenja sa prvim ali bez zadnjeg elementa, sa zadnjim ali bez prvog elementa, i bez prvog i zadnjeg elementa. Navedene teorije takođe dozvoljavaju eliminaciju kvantifikatora.

3.3.Teorija algebarskih zatvorenih polja

Jezik L teorije algebarski zatvorenih polja ima 2 konstantna simbola 0,1, unarni funkcijski simbol , dva binarna funkcijska simbola +, * i jedan relacijski simbol = . (Pisaćemo xy umesto x*y).

Neka je A skup sledećih formula:

1.Aksiome komutativne grupe u odnosu na + :

$\forall x \forall y \forall z(x + (y + z) = (x + y) + z)$

$\forall x \forall y(x + y = y + x)$

$\forall x(x + 0 = x)$

$\forall x(x + (-x) = 0)$

2. $\forall x \forall y \forall z(x(yz) = (xy)z)$

$\forall x \forall y(xy = yx)$

$\forall x(x \cdot 1 = x)$

$\forall x \exists y(x = 0 \vee xy = 1)$

$\forall x \forall y \forall z(x(y + z) = xy + xz)$

$0 \neq 1$

Svaki model skupa A je komutativno polje; za svaki term t jezika L postoji polinom $p(x_1,...,x_n)$ sa koeficijentima u **Z** takav da je $t = p(x_1,...,x_n)$ posledica skupa A.

Ako skupu A dodamo i aksiomu broj 3:

3.Za svako n > 1 važi formula:

$\forall x_0 \forall x_1 \cdots \forall x_{n-1} \exists x(x_0 + x_1 x + \cdots + x_{n-1} x^{n-1} + x^n = 0)$, formirali smo skup aksioma algebarski zatvorenog polja. Dokazaćemo da skup aksioma algebarski zatvorenog polja dozvoljava eliminaciju kvantifikatora. U dokazu nam je potrebna sledeća lema:

lema3.3.1. Neka su $p(x_1,...,x_k,x)$ i $q(x_1,...,x_k)$ dva terma jezika L, odnosno dva polinoma
sa koeficijentima iz **Z**. Tada postoji formula bez kvantifikatora F jezika L takva da u svakom komutativnom polju K, F' je skup onih k-torki $(\xi_1,...,\xi_k)\in K^k$ takvih da $p(\xi_1,...,\xi_k,x)$ deli $q(\xi_1,...,\xi_k,x)$.

dokaz: Neka je $p(x)=a_0+a_1x+\cdots+a_mx^m$ i $q(x)=b_0+b_1x+\cdots+b_nx^n$, gde su a_i i b_j polinomi po $x_1,...,x_k$, sa koeficijentima iz **Z**.

Do tražene formule F dolazimo pomoću rekurzije po m + n. Očigledno za $m+n=0$, formula F ima oblik $a_0\neq 0\vee b_0=0$.

Sada pretpostavimo da smo našli formule F sa zahtevanom osobinom za sve $m+n<h$, i da su polinomi $p(x)$ i $q(x)$ takvi da je $m+n=h$.

Neka je $n<m$; uvedimo oznaku: $p_1=a_0+a_1x+\cdots+a_{m-1}x^{m-1}$. Tražena formula ima sledeći oblik:

$$(a_m\neq 0\wedge b_0=b_1=\cdots=b_n=0)\vee(a_m=0\wedge F).$$

Ako je $m\leq n$, uvodimo oznake:
$p_1=a_0+a_1x+\cdots+a_{m-1}x^{m-1}$ i $q_1=a_mq(x)-b_nx^{n-m}p(x)$. Dakle, q_1 je polinom stepena manjeg od n. Po induktivnoj hipotezi, postoji formula F koja odgovara paru polinoma p_1,q i formula G koja odgovara paru p,q_1. Tražena formula je:

$$(a_m=0\wedge F)\vee(a_m\neq 0\wedge G).$$

Ovim smo dokazali lemu.

Dakle, sada imamo sve neophodne elemente da bismo dokazali da teorija algebarskih zatvorenih polja dozvoljava eliminaciju kvantifikatora.

Već smo pokazali da je dovoljno razmatrati formulu F oblika $\exists x(\alpha_1\wedge\cdots\wedge\alpha_r)$, gde je svaki α_i atomična formula u L ili negacija atomične formule. Dakle, svaki α_i ima oblik $t_1=t_2$ ili $t_1\neq t_2$, i znači ekvivalentno je formuli forme $t=0$ ili $t\neq 0$ (gde je $t=t_1-t_2$). Kako je: $t_1\neq 0\wedge\cdots\wedge t_l\neq 0$ ekvivalentno $t_1\cdots t_l\neq 0$, F možemo napisati na sledeći način:

$$\exists x(t_1=0\wedge\cdots\wedge t_k=0\wedge t\neq 0).$$

Svaki t_i je polinom po x čiji su koeficijenti polinomi po drugim promenljivim sa koeficijentima iz **Z**. Neka je term najvećeg stepena po x u t_i, jednak $a_i x^{n_i}$. Možemo pretpostaviti da ni za jedno i, n_i nije jednako nuli; u suprotnom, ako je naprimer $n_1 = 0$, F je ekvivalentna sa $t_1 = 0 \wedge \exists x(t_2 = 0 \wedge \cdots \wedge t_k = 0 \wedge t \neq 0)$.

Dalje nastavljamo dokaz rekurzijom po sumi svih n_i. Ako je $k \geq 2$, i naprimer $n_1 \geq n_2$, uvedimo sledeće oznake:

$$t_1' = a_2 t_1 - a_1 x^{n_1 - n_2} t_2 \text{ i } t_2' = t_2 - a_2 x^{n_2}.$$

Vidimo da je t_1' stepena manjeg od n_1 i t_2' je stepena manjeg od n_2. Formula F je ekvivalentna formuli:

$$(a_2 = 0 \wedge \exists x(t_1 = 0 \wedge t_2' = 0 \wedge \cdots \wedge t_k = 0 \wedge t \neq 0)) \vee$$
$$\vee (a_2 \neq 0 \wedge \exists x(t_1' = 0 \wedge t_2 = 0 \wedge \cdots \wedge t_k = 0 \wedge t \neq 0)),$$

i dakle, sveli smo je na formulu čiji je zbir najstarijih koeficijenata po x manji od prethodnog.

Ako je $k = 1$, formula F može da se zapiše kao: $\exists x(t_1 = 0 \wedge t \neq 0)$. Poznato je da u svakom algebarski zatvorenom polju K, za data dva polinoma $p(x), q(x)$ sa jednom slobodnom promenljivom x i koeficijentima iz K, postoji neko x_0 iz K takvo da je $p(x_0) = 0$ i $q(x_0) \neq 0$ akko p ne deli q^n, gde je n stepen po x u p. Dakle, ako je G formula bez kvantifikatora, tada, po lemi, ona je povezana sa parom terma t_1, t^n (n je stepen po x u t_1), pa je tražena formula F ekvivalentna $\neg$G (drugim rečima ima istu vrednost kao $\neg$G u svim algebarski zatvorenim poljima).

Ako je $k = 0$, formula F može biti zapisana kao $\exists x(t \neq 0)$. Neka je:
$t = a_0 + a_1 x + \cdots + a_n x^n$.
Pošto su sva algebarski zatvorena polja beskonačna i svaki polinom po jednoj promenljivoj koji nije identički jednak nuli ima samo konačan broj korena, možemo zaključiti da je tražena formula F ekvivalentna $a_0 \neq 0 \vee \cdots \vee a_n \neq 0$.

Ovim smo u potpunosti dokazali da skup A dozvoljava eliminaciju kvantifikatora u L.

Primena: Neki primeri eliminacije kvantifikatora
u algebarski zatvorenim poljima

primer3.3.1: Dokazivanje teorema pomoću eliminacije kvantifikatora
Neka je dat kvadrat ABCD. CE je paralelno sa BD i važi: BE = BD. Tačka F je presečna tačka duži BE i DC. Dokazati da je DF = DE.

Neka je A = (0,0), B = $(u_1,0)$, C = (u_1,u_1), D = $(0,u_1)$, E = (x_1,x_2) i F = (x_3,u_1).

Tada se polazne pretpostavke teoreme mogu izraziti pomoću sledećih jednakosti:

$h_1 = x_2^2 + x_1^2 - 2u_1x_1 - u_1^2 = 0$ BE = BD

$h_2 = u_1x_2 + u_1x_1 - 2u_1^2 = 0$ CE paralelno BD

$h_3 = x_2x_3 - u_1x_2 - u_1x_1 + u_1^2 = 0$ F pripada BE

Činjenica (DF = DE) može se izraziti pomoću formule:

$$c = (x_3 - 0)^2 + (u_1 - u_1)^2 - \left[(x_1 - 0)^2 + (x_2 - u_1)^2\right] = x_3^2 - x_2^2 + 2u_1x_2 - x_1^2 - u_1^2 = 0.$$

Dakle, algebarska forma teoreme je:

$$\forall u_1 \forall x_1 \forall x_2 \forall x_3 \left[(h_1 = 0 \wedge h_2 = 0 \wedge h_3 = 0 \wedge u_1 \neq 0) \Rightarrow c = 0\right].$$

Metodom eliminacije kvantifikatora dokazujemo da je data formula tačna, odnosno da važi teorema.

Eliminacija kvantifikatora se koristi u rešavanju mnogih teških problema u geometriji. Jedan od razloga za uspešnost ovog metoda je što algebarske jednačine za većinu teorema u geometriji uključuju samo kvadratne jednačine.

primer3.3.2: Ovo je primer primene eliminacije kvantifikatora u analitičkoj geometriji.

Ispitati da li se dve krive drugog reda seku.

Neka su zadate dve krive drugog reda F, G u opštem obliku:

$$F: A_1x^2 + 2B_1xy + C_1y^2 + 2D_1x + 2E_1y + F_1 = 0$$
$$G: A_2x^2 + 2B_2xy + C_2y^2 + 2D_2x + 2E_2y + F_2 = 0$$

Ispitivanje postojanja presečnih tačaka krivih F i G ekvivalentno je ispitivanju tačnosti sledeće formule:

$$\exists x \exists y (\ A_1x^2 + 2B_1xy + C_1y^2 + 2D_1x + 2E_1y + F_1 = 0 \wedge$$
$$A_2x^2 + 2B_2xy + C_2y^2 + 2D_2x + 2E_2y + F_2 = 0)$$

Metodom eliminacije kvantifikatora možemo utvrditi da li je formula tačna.
Ispitajmo ovaj metod na jednom konkretnom primeru.

Neka je $F: 2x^2+2xy+3y^2+2x+7=0$, i

$G: x^2+6xy+y^2+2y+4=0$.

Odgovarajuća formula je:

$\exists x \exists y(2x^2+2xy+3y^2+2x+7=0 \wedge x^2+6xy+y^2+2y+4=0)$.

Primenom algoritma eliminacije kvantifikatora na formulu, dobijamo da je ona logički ekvivalentna *true*. Dakle, krive F i G imaju zajedničkih tačaka.

3.4.Teorija realnih zatvorenih polja

Jezik L koji ovde razmatramo ima dva konstantna simbola $0,1$, jedan unarni funkcijski simbol $-$, dva binarna funkcijska simbola +, *, jedan unarni relacijski simbol >0, i binarni relacijski simbol = .

Neka je A skup sledećih formula:

1. Aksiome komutativnog polja:

$\forall x \forall y \forall z(x+(y+z)=(x+y)+z)$
$\forall x \forall y(x+y=y+x)$
$\forall x(x+0=x)$
$\forall x(x+(-x)=0)$
$\forall x \forall y \forall z(x(yz)=(xy)z)$
$\forall x \forall y(xy=yx)$
$\forall x(x \cdot 1=x)$
$\forall x \exists y(x=0 \vee xy=1)$
$\forall x \forall y \forall z(x(y+z)=xy+xz)$
$0 \neq 1$

2. $\forall x \forall y(x>0 \wedge y>0 \Rightarrow x+y>0)$
$\forall x(x=0 \vee x>0 \vee -x>0)$
$\forall x \neg(x>0 \wedge -x>0)$
$\forall x \forall y(x>0 \wedge y>0 \Rightarrow xy>0)$.

Svaki model skupa aksioma (1., 2.) je uređeno polje.

3. $\forall x \exists y(x=y^2 \vee -x=y^2)$
$\forall x_0 \forall x_1 \cdots \forall x_{2n} \exists x(x_0+x_1x+\cdots+x_{2n}x^{2n}+x^{2n+1}=0)$, za svako $n \geq 1$.

Modeli skupa aksioma A = (1. , 2. , 3.) su realna zatvorena polja.

Osnovni primeri modela realnih zatvorenih polja su skup **R** i realno zatvorenje skupa **Q**. Pokazaćemo da skup A dozvoljava eliminaciju kvantifikatora u L.

Za svaki term t, postoji polinom $p(x_1,\dots,x_n)$ sa koeficijentima iz Z takav da je $t = p(x_1,\dots,x_n)$ posledica od A.

Zbog jednostavnosti pisaćemo formulu $t - t' > 0$ kao $t > t'$ ili $t' < t$, i formulu $t < t' \wedge t' < t''$ kao $t < t' < t''$. Svaka atomična formula u L je ekvivalentna formuli forme $p(x,x_1,\dots,x_n) = 0$ ili $p(x,x_1,\dots,x_n) > 0$. Svaka formula bez kvantifikatora F je ekvivalentna (u svim modelima skupa A) disjunkciji formula oblika:

$$p_1 = 0 \wedge \cdots \wedge p_k = 0 \wedge q_1 > 0 \wedge \cdots \wedge q_l > 0.$$

Stepen po x u jednakosti $p_i = 0$ je najveći stepen po x u p_i, i stepen po x u nejednakosti $q_j > 0$ je za jedan veći od najvećeg stepena po x u q_j. Stepen po x u formuli F je maksimum stepena njenih atomičnih delova.

<u>lema3.4.1.</u> Za svaku formulu bez kvantifikatora C forme:

$$p_1 = 0 \wedge \cdots \wedge q_1 > 0 \wedge \cdots \wedge q_l > 0,$$

gde su p_i, q_j polinomi po $x, x_1,\dots,x_n$, postoji formula bez kvantifikatora B koja je ekvivalentna C (u svim modelima skupa A), takva da je stepen po x u formuli B manji ili jednak najmanjem stepenu po x u polinomima p_i (za koji pretpostavljamo da je različit od 0).

<u>dokaz:</u> Dokazaćemo lemu indukcijom po sumi stepena po x u p_i i q_j. Pretpostavimo da smo dokazali lemu za sve formule čiji je zbir stepena po x u p_i i q_j manji od h, i neka je $p_1 = 0 \wedge \cdots \wedge q_1 > 0 \wedge \cdots \wedge q_l > 0$ formula čiji je odgovarajući zbir jednak h.

Ako je $k \geq 2$, neka su $a_1 x^{m_1}$ i $a_2 x^{m_2}$ termi najvećeg stepena po x u p_1 i p_2, i uvedimo oznake:

$$\pi_1 = a_2 p_1 - a_1 p_2 x^{m_1 - m_2} \text{ i } \pi_2 = p_2 - a_2 x^{m_2},$$

uz pretpostavku $m_1 \geq m_2$. Tada je formula koju razmatramo ekvivalentna formuli:

$$(a_2 = 0 \wedge p_1 = 0 \wedge \pi_2 = 0 \wedge \cdots \wedge p_k = 0 \wedge q_1 > 0 \wedge \cdots \wedge q_l > 0) \vee$$
$$\vee (a_2 \neq 0 \wedge \pi_1 = 0 \wedge p_2 = 0 \wedge \cdots \wedge p_k = 0 \wedge q_1 > 0 \wedge \cdots \wedge q_l > 0),$$

i dakle sveli smo je na disjunkciju dve formule čiji je zbir stepena po x u p_i i q_j manji od h.

Ako je $k = 1$, formula ima sledeći oblik:

$$p = 0 \wedge q_1 > 0 \wedge \cdots \wedge q_l > 0 .$$

Ako svi q_i imaju stepen po x manji od stepena po x u p, sama formula zadovoljava lemu. Ako ne, naprimer ako q_1 ima stepen po x veći od stepena po x u p, i ako su ax^m i bx^n termi najvećeg stepena po x u p i q_1, onda je $m \leq n$. Uvedimo oznake:

$$P = p - ax^m \text{ i } Q = a^2 q_1 - abx^{n-m} p .$$

Tada je formula ekvivalentna formuli:

$$(a = 0 \wedge P = 0 \wedge q_1 > 0 \wedge \cdots \wedge q_l > 0) \vee$$
$$(a \neq 0 \wedge p = 0 \wedge Q > 0 \wedge q_2 > 0 \wedge \cdots \wedge q_l > 0),$$

i ponovo smo je sveli na disjunkciju dve formule čiji je zbir stepena po x u p_i i q_j manji od h.

Ako je $k = 0$, praktično nema šta da se dokazuje; ovim je dokaz leme završen.

<u>teorema3.4.1.</u> Neka je $C(x, x_1, ..., x_n)$ formula bez kvantifikatora stepena h po x. Neka su a,
b dve promenljive različite od $x, x_1, ..., x_n$. Tada postoji formula bez kvantifikatora F čije su promenljive među $a, b, x_1, ..., x_n$ takva da nijedan od njenih atomičnih delova ne sadrži obe promenljive a i b, i

$$F \Leftrightarrow \exists x(a < x < b \wedge C(x, x_1, ..., x_n))$$

je posledica skupa formula A $\cup \{a < b\}$.

<u>dokaz:</u> Dokazaćemo teoremu indukcijom po stepenu po x u formuli C. Ako je stepen po x u formuli C jednak nula, to znači da C ne sadrži x. Tada je formula:
$\exists x(a < x < b \wedge C)$ ekvivalentna formuli $C \wedge a < b$, i znači tražena formula F je sama formula C.

Pretpostavimo da smo dokazali teoremu za formule stepena manjeg od h, i da je stepen po x u formuli C jednak h.

Formula C je ekvivalentna disjunkcicji formula oblika $u_1 \wedge \cdots \wedge u_r$, gde je svaki u_i atomična formula ili negacija atomične formule, što znači da ima jedan od sledećih oblika: $p = 0, p \neq 0, p > 0$ ili $\neg(p > 0)$. Pošto je $p \neq 0$ ekvivalentno $p > 0 \vee -p > 0$, možemo pretpostaviti da C ima sledeći oblik:

$$p_1 = 0 \wedge \cdots \wedge p_k = 0 \wedge q_1 > 0 \wedge \cdots \wedge q_l > 0.$$

Lema3.4.1. dokazuje da ako je $k \geq 2$, ili ako je $k = 1$ i neki od q_j ima stepen po x veći ili jednak od stepena po x u p_1, možemo zameniti formulu C formulom B (formulom iz leme). Dakle, sveli smo formulu C na formulu manjeg stepena po x na koju možemo primeniti induktivnu hipotezu.

Dalje, ostalo nam je da razmotrimo formulu C koja može imati jedan od sledeća dva oblika:

1. $p = 0 \wedge q_1 > 0 \wedge \cdots \wedge q_l > 0$, gde je stepen po x u q_j manji od stepena po x u p, tako da je stepen po x u formuli C jednak stepenu po x u p;

2. $q_1 > 0 \wedge \cdots \wedge q_l > 0$.

Prvo ćemo razmatrati formulu C oblika 2. (i stepena h). Uvedimo oznaku:

$G = \exists x(a < x < b \wedge q_1 > 0 \wedge \cdots \wedge q_l > 0)$, gde je stepen po x u svim q_j manji od h.

Važi sledeća činjenica:

U svakom realnom zatvorenom polju, formula G je tačna akko u nekom otvorenom intervalu (α, β) sadržanom u intervalu (a, b), svaki q_j je strogo pozitivan.

Sledeći skup uslova obuhvata sve mogućnosti:

$$G_0(a,b) = \forall x\, [a < x < b \Rightarrow (q_1 > 0 \wedge \cdots \wedge q_l > 0)]$$

$$\begin{aligned} G_i(a,b) = {} & \exists u[a < u < b \wedge q_i(u) = 0 \wedge G_0(a,u)] \vee \\ & \vee \exists v[a < v < b \wedge q_i(v) = 0 \wedge G_0(v,b)] \quad (1 \leq i \leq l) \end{aligned}$$

$$H_{ij}(a,b) = \exists u \exists v\, [a < u < v < b \wedge q_i(u) = 0 \wedge q_j(v) = 0 \wedge G_0(u,v)], \qquad (1 \leq i \leq l, 1 \leq j \leq l).$$

Neka je $q_j^{(m)}$ oznaka za m-ti izvod od q_j, i neka je $Q_j(a)$ oznaka za sledeću formulu:

$$q_j(a) > 0 \vee [q_j(a) = 0 \wedge q_j^{(1)}(a) > 0] \vee \cdots \vee$$

$$[q_j(a)=0\wedge\cdots\wedge q_j^{(h-2)}(a)=0\wedge q_j^{(h-1)}(a)>0].$$

Tada je, u svakom modelu skupa A, $G_0(a,b)$ ekvivalentno:

$$\begin{aligned}&Q_1(a)\wedge\neg\exists x(a<x<b\wedge q_1=0)\wedge\cdots\wedge\\&\quad\wedge Q_l(a)\wedge\neg\exists x(a<x<b\wedge q_l=0).\end{aligned}$$

Pošto je stepen u formuli $q_j=0$ manji od h, induktivna hipoteza se može primeniti na svaku formulu $\exists x(a<x<b\wedge q_j=0)$. Dakle, $G_0(a,b)$ je ekvivalentno formuli bez kvantifikatora, specijalno disjunkciji čije su komponente oblika $K_r(a)\wedge L_r(b)$, $(1\le r\le s)$. $G_i(a,b)$ je ekvivalentno disjunkciji formula:

$$\begin{aligned}&K_r(a)\wedge\exists u[a<u<b\wedge q_i(u)=0\wedge L_r(u)]\vee\\&L_r(b)\wedge\exists v[a<v<b\wedge q_i(v)=0\wedge K_r(v)],\quad(1\le r\le s).\end{aligned}$$

Na svaku komponentu $G_i(a,b)$ možemo primeniti induktivnu hipotezu pošto su stepen po u u formuli $q_i(u)=0\wedge L_r(u)$ i stepen po v u formuli $q_i(v)=0\wedge K_r(v)$ manji od h (moguće primenom leme3.4.1.).

Dalje, $H_{ij}(a,b)$ je ekvivalentno disjunkciji formula :

$$\begin{aligned}&\exists u(a<u<b\wedge q_i(u)=0\wedge K_r(u)\wedge\\&\exists v(u<v<b\wedge q_j(v)=0\wedge L_r(v))).\end{aligned}$$

Pošto je $q_i(v)=0\wedge L_r(v)$ stepena po v manjeg od h, induktivna hipoteza se može primeniti.

Dakle $H_{ij}(a,b)$ je ekvivalentno disjunkciji (po r i t) formula:

$$N_{jrt}(b)\wedge\exists u(a<u<b\wedge q_i(u)=0\wedge K_r(u)\wedge M_{jrt}(u)),$$

na koje se induktivna hipoteza može očigledno primeniti.

Sada nam je ostalo da razmotrimo formule tipa 2. Po lemi3.4.1., treba samo da razmotrimo formule C oblika $p=0\wedge q_1>0\wedge\cdots\wedge q_{l'}>0$, gde je stepen po x u p jednak h, i stepen u svakom $q_j(1\le j\le l')$ je manji od h. Svešćemo ovaj slučaj na formule stepena manjeg od h, i na formule oblika 2. stepena h.

Očigledno je C ekvivalentno $C_1\vee C_2\vee C_3$, gde je:

$C_1 : p = 0 \wedge p' = 0 \wedge q_1 > 0 \wedge \cdots \wedge q_{l'} > 0$

$C_2 : p = 0 \wedge p' > 0 \wedge q_1 > 0 \wedge \cdots \wedge q_{l'} > 0$

$C_3 : p = 0 \wedge -p' > 0 \wedge q_1 > 0 \wedge \cdots \wedge q_{l'} > 0,$

gde p' označava prvi izvod po promenljivoj x.

C_1 pretstavlja slučaj kada p ima višestruku nulu. Pošto je stepen po x u p' manji od h, po lemi3.4.1., C_1 je ekvivalentno formuli stepena manjeg od h i možemo primeniti induktivnu hipotezu.

$\exists x(a < x < b \wedge C_2)$ je tačno u realnom zatvorenom polju akko postoji neki otvoren interval (α, β) sadržan u intervalu (a,b) u kom su svi $q_j (1 \leq j \leq l')$ i p' strogo pozitivni, i $p(\alpha) < 0, p(\beta) > 0$. Označimo $l = l' + 1$ i $q_l = p'$. Koristeći opet oznaku:

$$G_0(a,b) = \forall x\,[a < x < b \Rightarrow (q_1 > 0 \wedge \cdots \wedge q_l > 0)],$$

zaključujemo: $\exists x(a < x < b \wedge C_2)$ je ekvivalentno disjunkciji sledećih formula:

$$p(a) < 0 \wedge p(b) > 0 \wedge G_0(a,b),$$

$$p(a) < 0 \wedge \exists u\,[a < u < b \wedge q_l(u) = 0 \wedge p(u) > 0 \wedge G_0(a,u)] \vee$$
$$\vee\, p(b) > 0 \wedge \exists v\,[a < v < b \wedge q_l(v) = 0 \wedge -p(v) > 0 \wedge G_0(v,b)],$$

$$\exists u \exists v\,[a < u < v < b \wedge q_l(u) = 0 \wedge q_l(v) = 0 \wedge -p(u) > 0 \wedge p(v) > 0 \wedge G_0(u,v)]$$

G_0 smo već razmatrali. Pomoću leme3.4.1. zaključujemo da je svaka formula (bez kvan-tifikatora) u sklopu egzistencijalnog kvantifikatora ekvivalentna formuli stepena ma- njeg od h, pošto q_l ima stepen manji od h.

C_3 treba razmatrati zamenom p' i $-p'$, $p < 0$ i $p > 0$.

Ovim smo završili dokaz teoreme.

<u>teorema3.4.2.</u> A dozvoljava eliminaciju kvantifikatora u L.

<u>dokaz:</u> Dovoljno je dokazati teoremu za formulu forme $\exists x C(x, x_1, \ldots, x_n)$. Dodaćemo u L dve konstante u i $\frac{1}{u}$, i dodaćemo u A aksiomu $u \cdot \frac{1}{u} = 1$. Po teoremi3.4.1.,

formula:

$$\exists x(-1 < x < 1 \wedge C(x \cdot \frac{1}{u}, x_1, \ldots, x_n))$$

je ekvivalentna formuli bez kvantifikatora Q. Svaka atomična formula u Q ima oblik $p(x \cdot \frac{1}{u}) = 0$ ili $p(x \cdot \frac{1}{u}) > 0$, i dakle, po aksiomi $u \cdot \frac{1}{u} = 1$, ima oblik $p(x,u) = 0$ ili $p(x,u) > 0$. Dakle postoji formula bez kvantifikatora $R(z)$, gde je z promenljiva u L, takva da je:

$$u \cdot \frac{1}{u} = 1 \Rightarrow \exists x(-1 < x < 1 \wedge C(x \cdot \frac{1}{u}, x_1, \ldots, x_n))$$

ekvivalentno $R(u)$. Očigledno je da su u svim modelima od A dve formule: $\exists x C(x, x_1, \ldots, x_n)$ i $\exists z(0 < z < 1 \wedge R(z))$ ekvivalentne. Dalje, po teoremi3.4.1, poslednja formula je ekvivalentna formuli bez kvantifikatora.

Ovim je završen dokaz teoreme.

Primena u kontrolnoj teoriji

Opisaćemo dinamički sistem pomoću nelinearne diferencijalne jednačine:

$$\dot{x} = f(x,y) \qquad (1)$$

gde $x \in R^n, u \in R^m$ i f je polinom sa koeficijentima iz R. Sistemske promenljive x i u pretstavljaju redom stanje i kontrolu sistema, i imamo dodatna ograničenja za sistemske promenljive:

$$x \in X \text{ i } u \in U \qquad (2)$$

gde su X i U semi-algebarski skupovi koji definišu dopustivo stanje i dopustivu kontrolu. $x_0 \in X$ je stacionarna tačka ako postoji $u_0 \in U$ tako da je $f(x_0, u_0) = 0$. Dakle, skup stacionarnih tačaka sistema (1), uz uslov (2) je:

$$S = \{x \in R^n : \exists u(f(x,u) = 0 \wedge x \in X \wedge u \in U\}.$$

Primetimo da se "$x \in X$" može izraziti pomoću formule $\psi(x)$ u jeziku realnih zatvorenih polja jer je X semi-algebarski skup.

Primer3.4.1: Pronaći ćemo skup stacionarnih tačaka za dinamički sistem:

$$\dot{x}_1 = -x_1 + x_2 u$$
$$\dot{x}_2 = -x_2 + (1 + x_1^{\,2})u + u^3$$

sa ograničenjem $-\frac{1}{2} \leq u \leq \frac{1}{2}$. Skup stacionarnih tačaka $S \subseteq R^2$ definiše se formulom $\phi(x_1, x_2)$:

$$\exists u(-x_1 + x_2 u = 0 \wedge -x_2 + (1 + x_1^{\,2})u + u^3 = 0 \wedge -\frac{1}{2} \leq u \leq \frac{1}{2}).$$

Pošto teorija realnih zatvorenih polja dopušta eliminaciju kvantifikatora, možemo dobiti formulu bez kvantifikatora $\psi(x_1, x_2)$ ekvivalentnu formuli ϕ:

$$x_2^{\,4} - x_1^{\,3} x_2^{\,2} - x_1 x_2^{\,2} - x_1^{\,3} = 0 \wedge (x_2 + 2x_1 \leq 0 \vee x_2 - 2x_1 \geq 0).$$

Neka je Γ algebarska kriva u R^n data jednačinom:

$x = g(t)$, gde $t \in [a,b]$ i $g: R \to R^n$.

Možemo razmotriti mogućnost vođenja sistema definisanog pomoću (1) i (2) iz inicijalnog stanja $x_{\text{int}} = g(a)$ do finalnog stanja $x_{fin} = g(b)$ duž krive Γ. Pretpostavljajući da $g(t) \in X$ za svako $t \in [a,b]$, takvo vođenje je moguće ako za svaku tačku krive Γ postoji dopustiva kontrola $u \in U$ tako da vektor $f(x,u)$ ima isti pravac i smer kao i tangentni vektor krive:

$$f(g(t),u) = k\dot{g}(t),\ k > 0,\ t \in [a,b].$$

Ovaj uslov može biti zapisan kao sledeća rečenica teorije prvog reda:

$$(\forall t \in [a,b])(\exists u \in U)(\exists k > 0)(f(g(t),u) = k\dot{g}(t)).$$

Eliminacijom kvantifikatora iz ove rečenice dobijamo Bulovu kombinaciju jednakosti i nejednakosti realnih brojeva, čiju je valjanost lako proveriti u skupu R.

primer3.4.2: Razmotrimo sistem:

$$\dot{x}_1 = -x_1 + 2$$
$$\dot{x}_2 = -x_2 - x_1^{\,2} + 4u$$

uz ograničenje $-1 \le u \le 1$. Da li je moguće voditi sistem duž krive:

$$x = g(t) = (t, 3t^2 - 2t^3),\ t \in [0,1],$$

koristeći dopustivu kontrolu?

Problem se može svesti na sledeće pitanje: da li je tačna formula:

$$(\forall t \in [a,b])(\exists u \in [-1,1])(\exists k > 0)(-t + 2 = k \wedge -3t^2 + 2t^3 + t^2 + 4u = k(6t - 6t^2)),$$

i posle eliminisanja kvantifikatora, dobija se potvrdan odgovor.

3.5.Teorija diskretnog uređenja bez prvog ili poslednjeg elementa

Razmotrimo jezik L koji ima jedan unarni funkcijski simbol s (predstavlja oznaku za: sledbenik) i dva binarna relacijska simbola < i = . Dakle, termi u L imaju oblik: $s^p x$ (simbol s ponovljen p puta ispred promenljive x.

Neka je A skup sledećih formula:

$\forall x \neg(x < x)$
$\forall x \forall y \forall z(x < y \wedge y < z \Rightarrow x < z)$
$\forall x \forall y(x = y \vee x < y \vee y < x)$
$\forall x \forall y(x < y \Leftrightarrow (y = sx \vee sx < y))$
$\forall x \exists y(x = sy)$.

Pokazaćemo da skup A dozvoljava eliminaciju kvantifikatora u L.

Kao i u prethodnim primerima, treba samo da razmotrimo formule oblika: $\exists x(\alpha_1 \wedge \cdots \wedge \alpha_r)$, gde svaki α_i ima oblik: $t_1 < t_2$ ili $t_1 = t_2$ odnosno $s^{p_1}x_1 < s^{p_2}x_2$ ili $s^{p_1}x_1 = s^{p_2}x_2$.

Izvešćemo dokaz rekurzijom po r. Slučaj $r = 1$ je očigledan. Pretpostavimo da smo dokazali tvrđenje u slučajevima kada je $r < h$ i da je data formula oblika: $\exists x(\alpha_1 \wedge \cdots \wedge \alpha_h)$. Očigledno je da ukoliko je u nekoj atomičnoj formuli $s^{p_1}x_1 < s^{p_2}x_2$ ili $s^{p_1}x_1 = s^{p_2}x_2$ nijedna od promenljivih x_1, x_2 nije jednaka x, možemo odmah svesti problem na slučaj $r = h - 1$. Ako su obe promenljive x_1, x_2 jednake x u nekom α_i, onda α_i ima oblik $s^{p_1}x < s^{p_2}x$ ili $s^{p_1}x = s^{p_2}x$. Navedene formule su ekvivalentntne redom formulama $p_1 < p_2$ odnosno $p_1 = p_2$, i ponovo smo sveli formulu na slučaj $r = h - 1$.

Da bismo pojednostavili oznake, pisaćemo formule $s^p x < x_1$ i $s^p x = x_1$ kao: $x < s^{-p} x_1$ i $x = s^{-p} x_1$. Znači formule $s^p x < s^{p_1} x_1$ i $s^p x = s^{p_2} x_1$ su ekvivalentne: $x < s^{p_1 - p} x_1$ i $x = s^{p_2 - p} x_1$. Dakle, formula koju razmatramo može biti zapisana na sledeći način:

$$\exists x(x < t_1 \wedge \cdots \wedge x < t_k \wedge u_1 < x \wedge \cdots \wedge u_l < x \wedge x = v_1 \wedge \cdots \wedge x = v_m),$$

gde termi t, u, v imaju oblik $s^p y, p \in Z$.

Ako je $k > 1$ formula je ekvivalentna sledećoj formuli:

$$(t_1 < t_2 \wedge \exists x(x < t_1 \wedge x < t_3 ...)) \vee (\neg(t_1 < t_2) \wedge \exists x(x < t_2 \wedge x < t_3 ...)),$$

i ponovo smo sveli formulu na slučaj $r = h - 1$.

Dolazimo do sličnog zaključka ako je $l > 1$.

Znači, ostalo nam je još da razmotrimo formulu u slučaju $k = 1$ i $l = 1$:

$$\exists x(x < t_1 \wedge u_1 < x \wedge x = v_1 \wedge \cdots \wedge x = v_m),$$

koja je ekvivalentna formuli:

$$(v_1 < t_1 \wedge u_1 < v_1 \wedge v_1 = \cdots = v_m),$$

koja ne sadrži kvantifikatore.

Ovim smo u potpunosti dokazali da teorija diskretnog uređenja dozvoljava eliminaciju kvantifikatora.

<u>primer3.5.1:</u> Pokazaćemo da je u skupu aksioma A simbol s neophodan da bi eliminacija kvantifikatora bila moguća. Preciznije, razmotrimo jezik L koji ima dva relacijska simbola $<$ i $=$, i neka je skup B skup sledećih formula jezika L:

$$\forall x \neg(x < x)$$
$$\forall x \forall y \forall z(x < y \wedge y < z \Rightarrow x < z)$$
$$\forall x \forall y(x = y \vee x < y \vee y < x)$$
$$\forall x \exists y \forall z(x < z \Leftrightarrow y = z \vee y < z)$$
$$\forall x \exists y \forall z(z < x \Leftrightarrow y = z \vee z < y)$$

Modeli skupa formula B su isti kao i modeli skupa A, dakle diskretno uređeni skupovi bez prvog ili poslednjeg elementa, ali skup B ne dozvoljava eliminaciju kvantifikatora u L.

Da bismo dokazali prethodno tvrđenje, uvedimo najpre neke osnovne definicije i teoreme koje su poznate u teoriji modela:

def Neka je **A** model jezika L, i
$L_A = L \cup \{a : a \in A\}$.
Dijagram modela **A** je teorija D(**A**) jezika L_A čije su aksiome atomične rečenice i negacije atomičnih rečenica jezika L_A koje su tačne u $(\mathbf{A}, a)_{a \in A}$.

teorema3.5.1. Ako skup A dozvoljava eliminaciju kvantifikatora u L i ako je D(**A**) dijag-
ram modela **A** skupa A, tada je teorija A $\cup$ D(**A**) kompletna (za jezik L_A teorije A $\cup$ D(**A**)).

dokaz: Razmotrimo sledeći model skupa B: uređen skup **Z** celih brojeva. Ako je D_Z dijagram ovog modela i ako pretpostavimo suprotno, da B dozvoljava eliminaciju kvantifikatora, onda je skup B $\cup D_Z$ kompletan. Međutim, ako dodamo broj $\frac{1}{2}$ u ovaj model, i dalje imamo model skupa B $\cup D_Z$, ali formula $\exists x(0 < x < 1)$ nije zadovoljena u prvom modelu, ali jeste u drugom. Ovo pokazuje da skup B $\cup D_Z$ nije kompletan, pa skup B ne dozvoljava eliminaciju kvantifikatora.

primer3.5.2: Ovo je jedan primer teorije koja ne dozvoljava eliminaciju kvantifikatora.
Jezik L koji ovde razmatramo ima dva konstantna simbola 0, 1, unarni relacijski simbol >0, unarni relacijski simbol $n|$ (čita se n deli, $n > 1$, $n \in$ **N**), binarni relacijski simbol =, unarni funkcijski simbol $-$, i binarni funkcijski simbol +.
Neka je skup A skup sledećih formula:

1.Aksiome komutativne grupe:

$$\forall x \forall y \forall z (x + (y + z) = (x + y) + z)$$
$$\forall x \forall y (x + y = y + x)$$
$$\forall x (x + 0 = x)$$
$$\forall x (x + (-x) = 0)$$

2.Aksiome totalnog uređenja koje su kompatibilne sa strukturom grupe:

$$\forall x \forall y (x > 0 \wedge y > 0 \Rightarrow x + y > 0)$$
$$\forall x (x = 0 \vee x > 0 \vee -x > 0)$$
$$\forall x \neg(x > 0 \wedge -x > 0)$$

3.Aksiome diskretnog uređenja:

$$\forall x(x > 0 \Leftrightarrow (x = 1 \vee x - 1 > 0)$$

4. $\forall x(n \mid x \Leftrightarrow \exists y(x = ny))$ za svako $n > 1$.

Pokazaćemo da skup A = (1.,2.,3.,4.) ne dozvoljava eliminaciju kvantifikatora. Razmotrimo grupu $G = \mathbf{Z} \times \mathbf{Z}$, u kojoj je uvedena relacija > 0 na sledeći način: $(a,b) > 0$ akko $(a > 0 \vee (a = 0) \wedge b > 0)$.

G je model skupa A koji sadrži skup **Z** kao podmodel (izvrši se identifikacija elementa (0, n) i elementa n). Neka je D_Z dijagram modela **Z**. Tada skup (1.,2.,3.,4., D_Z) nije kompletan pošto je formula $\forall x \exists y(x = 2y \vee x + 1 = 2y)$ tačna u **Z** ali nije tačna u G.

primer3.5.3: Ovo je primer teorije strukture (**Q**, +, −, <, 0).

Dakle, neka je T teorija strukture (**Q**, +, −, <, 0), odnosno T je skup svih rečenica koje su tačne u pomenutom modelu. Pokazaćemo da T dopušta eliminaciju kvantifikatora. Da bismo izveli dokaz, prvo treba da analiziramo formule bez kvantifikatora po njihovoj složenosti. Atomične formule imaju jedan od sledećih oblika:

$$n_1 x_1 + \cdots + n_k x_k = 0 \tag{1}$$

$$n_1 x_1 + \cdots + n_k x_k < 0, \tag{2}$$

gde su n_i celi brojevi $\neq 0$, i naprimer, $2x$ i $-2x$ su redom oznake za izraze $x + x$ i $(-x) + (-x)$. Dalje, negacija formule (1) je formula:

$$(\sum_{i=1}^{k} n_i x_i) < 0 \vee (\sum_{i=1}^{k} - n_i x_i) < 0, \tag{3}$$

i negacija formule (2) je formula:

$$(\sum_{i=1}^{k} n_i x_i) = 0 \vee (\sum_{i=1}^{k} - n_i x_i) < 0 \tag{4}$$

Ako kombinujemo formule (3) i (4) sa činjenicom da su negacije formula $\varphi \wedge \psi$, $\varphi \vee \psi$, $\varphi \Rightarrow \psi$ i $\varphi \Leftrightarrow \psi$ redom formule $\neg\varphi \vee \neg\psi, \neg\varphi \wedge \neg\psi, \varphi \wedge \neg\psi$ i $(\neg\varphi \wedge \psi) \vee (\varphi \wedge \neg\psi)$, možemo zaključiti da svaka formula bez kvantifikatora je, do na ekvivalenciju, Bulova kombinacija atomičnih formula (1) i (2). Da bismo dalje pojednostavili oznake, neka je:

$$\frac{m_1}{n_1}x_1 + \cdots + \frac{m_k}{n_k}x_k = 0$$

označeno sa:

$$m_1 n_2 \cdots n_k x_1 + \cdots + m_k n_1 \cdots n_{k-1} x_k = 0,$$

i neka je:

$$\frac{m_1}{n_1}x_1 + \cdots + \frac{m_k}{n_k}x_k < 0$$

označeno sa:

$$m_1 n_2 \cdots n_k x_1 + \cdots + m_k n_1 \cdots n_{k-1} x_k < 0$$

(gde su n_i pozitivni brojevi). Konačno možemo da skiciramo algoritam:

Ulaz: formula $\exists x\varphi$, gde je φ formula bez kvantifikatora.
Izlaz: formula bez kvantifikatora $A(\varphi)$ takva da je $\exists x\varphi \approx A(\varphi)$.

• Proveriti da li je x lažna promenljiva za φ ili ne. Ako je x lažna promenljiva, onda je $A(\varphi) = \varphi$. U suprotnom, nastavljamo algoritam.

• Ako je φ formula oblika $\varphi_1 \vee \cdots \vee \varphi_n$, tada je $A(\varphi)$ formula $A(\varphi_1) \vee \cdots \vee A(\varphi_n)$.

• Ako je φ formula oblika:

$$\bigwedge_{i=1}^{m} \bigvee_{j=1}^{n_i} \varphi_{ij},$$

sa barem jednim i za koje je $n_i > 1$, tada je $A(\varphi)$ formula:

$$\bigvee_{f \in F} A(\bigwedge_{i=1}^{m} \varphi_{if(i)}),$$

gde je F skup svih funkcija sa domenom $\{1,\ldots,m\}$ takvih da $f(i) \in \{1,\ldots,n_i\}$,

za sve i iz domena.

• Ako je φ formula oblika $n_0 x + A = 0 \wedge \psi(x)$, gde je A term bez pojave x, tada je $A(\varphi)$ formula $A(\psi(-A/n_0))$.

• Ako je φ formula oblika $p_1 x + A_1 < 0 \wedge \cdots \wedge p_k x + A_k < 0$, gde su A_i termi bez pojave x i p_i su pozitivni brojevi, tada je $A(\varphi)$ formula koja pretstavlja logičku istinu.

• Ako je φ formula oblika $n_1 x + B_1 < 0 \wedge \cdots \wedge n_l x + B_l < 0$, gde su B_i termi bez pojave x i n_i su negativni brojevi, tada je $A(\varphi)$ formula koja pretstavlja logičku istinu.

• Ako je φ formula oblika:

$$(p_1 x + A_1 < 0 \wedge \cdots \wedge p_k x + A_k < 0) \wedge (n_1 x + B_1 < 0 \wedge \cdots \wedge n_l x + B_l < 0),$$

gde su A_i i B_i termi bez pojave x, p_i su pozitivni i n_i su negativni brojevi, tada je $A(\varphi)$ formula:

$$p_1 B_1 + n_1 A_1 < 0 \wedge p_1 B_2 + n_2 A_1 < 0 \wedge \cdots \wedge p_k B_l + n_l A_k < 0.$$

Obratimo pažnju na složenost algoritma: poznato je da je ovaj problem NP-kompletan. Ova procedura se lako može transformisati u proceduru za nedeterminističke mašine polinomijalne složenosti. Na determinističkim mašinama ona ima eksponencijalnu složenost.

Literatura

1. An introduction to model theory, Ž.Mijajlović Novi Sad 1987

2. Handbook of automated reasoning , Alan Robinson, Andrei Voronkov

3. Elements of Mathematical logic , G. Kreisel (Stanford University), J. L. Krivine (Université de Paris)

4. Matematički programski problemi veštačke inteligencije u oblasti automatskog dokazivanja teorema , P. Hotomski, I. Pevac Beograd 91

5. Uvod u matematiku , M. Pepić UM BiH Sarajevo , 2000

6. Quantifier elimination in mathematical theories , M. Milošević, M. Udovičić, D. Doder, D. Ilić ETRAN Conference , Čačak 2004

7. One implementation of the quantifier elimination for the theory of structure $(\mathbf{Q}, +, -, <, 0)$, D. Doder , M. Udovičić , M. Milošević , D. Ilić ETRAN Conference , Budva