

Implementacija Furije-Mockinove procedure za ispitivanje zadovoljivosti konjukcije linearnih ograničenja

Seminarski rad u okviru kursa
Automatsko rezonovanje
Matematički fakultet

Aleksandar Stefanović, 1021/2023

27. septembar 2025.

Sažetak

Furijsko-Mockinova procedura se, kao metod za proveru zadovoljivosti sistema nejednakosti, prvobitno pojavila još u prvoj polovini 19. veka. Sa razvojem oblasti automatskog rezonovanja, vremenom je pronášla i upotrebu u automatskom dokazivanju teorema. U ovom radu je predstavljena implementacija jednostavnog dokazivača teorema zasnovanog na Furije-Mockinovoj proceduri.

Sadržaj

1	Uvod	2
2	Osnove	2
2.1	Problemi ograničenja	2
2.2	Logika prvog reda	3
2.2.1	Sintaksa logike prvog reda	3
2.2.2	Semantika logike prvog reda	4
2.2.3	Deduktivni sistemi	5
2.3	Jezik logike prvog reda nad linearnim fragmentom teorije uređenih polja	5
3	Opis metode	5
3.1	Eliminacija promenljive po jednakosti	6
3.2	Eliminacija promenljive po nejednakostima	6
3.3	Proširenje do procedure odlučivanja u linearnom fragmentu teorije uređenih polja	7
4	Implementacija	8
4.1	Organizacija projekta	8
4.2	Prevođenje projekta i upotreba	9
5	Zaključak	10

1 Uvod

Furije-Mockinova procedura, u svojoj osnovi, predstavlja metod za proveru zadovoljivosti skupa linearnih ograničenja [1]. Međutim, kako je ova procedura zasnovana na eliminaciji promenljivih, ona se može razmatrati i u kontekstu procedure odlučivanja, zasnovane na eliminaciji kvantifikatora, u linearnom fragmentu teorije uređenih polja [2].

Ostatak rada je organizovan na sledeći način. Poglavlje 2 predstavlja osnovne definicije i pojmove koji će se koristiti u ostatku rada. Poglavlje 3 daje detaljan opis Furije-Mockinove procedure, kako u kontekstu metoda za proveru zadovoljivosti linearnih ograničenja, tako i u kontekstu njegovog proširenja u proceduru odlučivanja u linearnom fragmentu teorije uređenih polja. Konačno, poglavlje 4 daje pregled implementacije Furije-Mockinove procedure.

2 Osnove

U ovom poglavlju će biti predstavljeni osnovni pojmovi i definicije koje će se koristiti u nastavku rada. U poglavlju 2.1 su definisani problemi ograničenja i jedne njihove varijante — problema zadovoljivosti linearnih ograničenja. Poglavlje 2.2 opisuje osnovne aspekte logike prvog reda, a poglavlje 2.3 opisuje jezik logike prvog reda koji će biti korišćen u ostatku rada.

2.1 Problemi ograničenja

Problem zadovoljivosti ograničenja je uređena trojka $P = (X, D, C)$, pri čemu je:

- $X = (x_1, \dots, x_n)$ — niz promenljivih;
- $D = (D_1, \dots, D_n)$ — niz domena odgovarajućih promenljivih (D_i je domen promenljive x_i);
- $C = (C_1, \dots, C_m)$ — niz ograničenja nad vrednostima promenljivih ($C_k \subseteq D_1 \times \dots \times D_n$).

Rešenje problema zadovoljivosti ograničenja je onda bilo koja uređena n-torka $d = (d_1, \dots, d_n) \in D_1 \times \dots \times D_n$ za koju dodatno važi $d \in C_k$ za svako $C_k \in C$. Analogno, **problem ispitivanja zadovoljivosti ograničenja** je problem odlučivanja koji se bavi samo ispitivanjem da li postoji takva uređena n-torka d , ali ne i pronalaženjem konkretnih vrednosti odgovarajućih promenljivih. Ukoliko su ograničenja C zadata kao niz linearnih (u odnosu na promenljive x_i, \dots, x_n) jednakosti i nejednakosti oblika $a_1x_1 + a_2x_2 + \dots + a_nx_n \otimes b$, pri čemu je $\otimes \in \{=, <, >, \neq, \leq, \geq\}$, onda se može govoriti o **problemu (ispitivanja) zadovoljivosti linearnih ograničenja**.

U nastavku sledi jedan jednostavan primer problema zadovoljivosti linearnih ograničenja. Neka su date dve promenljive, x i y — potrebno je odrediti da li naredni skup linearnih nejednakosti ima bar jedno rešenje nad domenom racionalnih brojeva \mathbb{Q} :

$$x + y < 4$$

$$x - y > 1$$

Jedno očigledno rešenje datog skupa nejednakosti je $x = 2$ i $y = \frac{1}{2}$, te je ovaj skup nejednakosti zadovoljiv. Međutim, kako ručno rešavanje problema zadovoljivosti linearnih ograničenja postaje nepraktično sa rastom broj promenljivih i broja ograničenja, neophodni su algoritmi za automatizaciju ovog postupka. Jedna od metoda koja to omogućava je upravo Furije-Mockinova procedura koja će biti detaljnije opisana u poglavlju 3.

2.2 Logika prvog reda

Logika prvog reda, ili predikatska logika, formalni je sistem za zadavanje odnosa između određenih logičkih objekata. Uvodeći, između ostalog, egzistencijalne i univerzalne kvantifikatore, kao i funkcijske i predikatske simbole, logika prvog reda je izražajnija od iskazne logike koja je omogućavala samo predstavljanje iskaza i zadavanje jednostavnih odnosa između njih. Glavni aspekti logike prvog reda su njena sintaksa, njena semantika i deduktivni sistemi koji se bave konstrukcijom (tj. izvođenjem) dokaza na osnovu aksioma (tvrđenja koja uvek važe) u logičkom sistemu.

2.2.1 Sintaksa logike prvog reda

Sintaksa logike prvog reda opisuje strukturu njenih validnih formula, bez razmatranja značenja tih formula ili njihovih elemenata. Pre definisanja strukture formule logike prvog reda, potrebno je definisati pojam signature.

Signatura (ili jezik) \mathcal{L} je uređena trojka $\mathcal{L} = (\Sigma, \Pi, ar)$, pri čemu Σ označava skup funkcijskih simbola, Π skup predikatskih simbola, a ar funkciju $ar : \Sigma \cup \Pi \rightarrow \mathbb{N}$ koja dodeljuje arnost, tj. broj argumenata, svim funkcijskim i predikatskim simbolima. Struktura formule logike prvog reda, nad nekom signaturom \mathcal{L} , formalno se može definisati preko narednih pojmova:

Term — skup termova nad signaturom \mathcal{L} je najmanji skup koji zadovoljava:

- svaka promenljiva je term;
- svaki simbol konstante signature \mathcal{L} je term;
- ako su t_1, t_2, \dots, t_k termovi i f funkcijski simbol signature \mathcal{L} arnosti k , onda je i $f(t_1, \dots, t_k)$ term.

Atomička formula — skup atomičkih formula nad signaturom \mathcal{L} je najmanji skup koji zadovoljava:

- logičke konstante \top i \perp su atomičke formule;
- svako iskazno slovo (predikatski simbol arnosti 0) signature \mathcal{L} je atomička formula;
- ako su t_1, t_2, \dots, t_k termovi i p predikatski simbol signature \mathcal{L} arnosti k , onda je $p(t_1, \dots, t_k)$ atomička formula.

Formula — skup formula (prvog reda) nad signaturom \mathcal{L} je najmanji skup koji zadovoljava:

- atomičke formule su formule;
- ako je A formula, onda je i $\neg A$ formula;
- ako su A i B formule, onda su i $A \wedge B$, $A \vee B$, $A \implies B$ i $A \iff B$ formule;
- ako je A formula i x promenljiva, onda su i $\exists x.A$ i $\forall x.A$ formule — promenljiva x u formuli A je u ovom slučaju *vezana* (promenljive koje nisu vezane ni jednim kvantifikatorom su *slobodne*, a formule bez slobodnih promenljivih se nazivaju *rečenice*);
- ako je A formula, onda je i (A) formula.

Prioriteti logičkih veznika su, u opadajućem redosledu, $\neg, \wedge, \vee, \implies, \iff$. U implementaciji i ostatku rada, smatra se da je prioritet kvantifikatora manji od bilo kog logičkog veznika.

2.2.2 Semantika logike prvog reda

Semantika logike prvog reda dodeljuje značenje validnim formulama prvog reda. U svrhe definisanja semantike, a time i tačnosti formula, uvode se naredni pojmovi:

\mathcal{L} -struktura \mathcal{D} — \mathcal{L} -struktura \mathcal{D} , definisana nad signaturom \mathcal{L} , definiše domen, u oznaci D , kao i interpretacije funkcijskih i predikatskih simbola. Za funkcijski simbol f arnosti k ovo podrazumeva funkciju $f_{\mathcal{D}} : D^k \rightarrow D$, a za predikatski simbol p arnosti l ovo podrazumeva funkciju $p_{\mathcal{D}} : D^l \rightarrow \{0, 1\}$.

Valuacija v — Funkcija koja, za skup promenljivih V i domen D iz \mathcal{L} -strukture \mathcal{D} , svakoj promenljivoj $v \in V$ dodeljuje neku vrednost iz domena D .

Interpretacija I_v — Funkcija koja, za definisanu \mathcal{L} -strukturu \mathcal{D} i valuaciju v , određuje tačnost formule logike prvog reda, tj. vrši preslikavanje formule u skup $\{0, 1\}$.

U kontekstu određivanja tačnosti formule, definišu se i naredni pojmovi:

Zadovoljivost — Ukoliko za neku \mathcal{L} -strukturu \mathcal{D} i valuaciju v za formulu A važi $I_v(A) = 1$, onda se za formulu A kaže da je zadovoljliva, a za par (\mathcal{D}, v) da je model formule A , u oznaci $(\mathcal{D}, v) \models A$.

Tautoložičnost — Ukoliko je za neku \mathcal{L} -strukturu \mathcal{D} interpretacije formule A tačna za svaku valuaciju v , onda se za formulu A kaže da je tautologija u \mathcal{L} -strukтури \mathcal{D} , u oznaci $\mathcal{D} \models A$. Ukoliko ovo važi za svaku \mathcal{L} -strukturu \mathcal{D} , onda se za formulu A kaže da je tautologija, u oznaci $\models A$.

Ekvizadovoljivost — Formule A i B su ekvizadovoljlive, u oznaci $A \equiv_s B$, ako i samo ako su obe zadovoljlive ili su obe nezadovoljlive.

Ekvivalentnost — Formule A i B su ekvivalentne, u oznaci $A \equiv B$, ako i samo ako za svaku \mathcal{L} -strukturu \mathcal{D} i svaku valuaciju v važi $(\mathcal{D}, v) \models A$ akko $(\mathcal{D}, v) \models B$.

2.2.3 Deduktivni sistemi

Uloga deduktivnih sistema je ispitivanje dokazivosti i dokazivanje tvrdjenja u datom logičkom sistemu. Za razliku od semantičkog pristupa, u kojem se zadovoljivost (ili zadovoljivost) dokazuju ili opovrgavaju na osnovu semantike, tj. interpretacije simbola, dokazivanje u deduktivnim sistemima podrazumeva izvođenje dokaza na osnovu sintaksičkih pravila i skupa polaznih pretpostavki. Osnovni pojmovi koji će se koristiti dalje u radu su:

Aksioma — Tvrdjenje za koje se pretpostavlja da uvek važi. Aksiome predstavljaju početne tvrdnje u dokazima u deduktivnim sistemima.

Teorema — Formula A je teorema, u oznaci $\vdash A$, ako i samo ako se za A može konstruisati dokaz samo iz skupa aksioma.

Teorija — Teorija \mathcal{T} nad signaturom \mathcal{L} zadata skupom aksioma $Ax(\mathcal{T})$ je skup svih teorema u odnosu na aksiome $Ax(\mathcal{T})$.

Model teorije — Model teorije \mathcal{T} je signatura \mathcal{L} u kojoj su sve formule iz $Ax(\mathcal{T})$ tačne.

2.3 Jezik logike prvog reda nad linearnim fragmentom teorije uređenih polja

Polje racionalnih brojeva je jedan od modela teorije uređenih polja. Upravo ovaj model će se razmatrati u ostatku rada. Njemu odgovarajuća signatura je sledeća — $\mathcal{L} = (\mathbb{Q}, +, \cdot, \leq)$, pri čemu svi simboli imaju naredne arnosti i interpretacije:

- \mathbb{Q} — Simboli konstanti koji odgovaraju racionalnim brojevima.
- $+$ — Funkcijski simbol arnosti 2 koji se interpretira kao standardno sabiranje nad racionalnim brojevima.
- \cdot — Funkcijski simbol arnosti 2 koji se interpretira kao standardno množenje nad racionalnim brojevima.
- \leq — Predikatski simbol arnosti 2 koji se interpretira kao poređenje *manje ili jednako*¹ nad racionalnim brojevima.

Furijs-Mockinova procedura, koja je opisana u poglavlju 3, ograničava se na linearni fragment teorije uređenih polja, te je primena množenja dozvoljena samo nad simbolima konstanti ili jednom konstantom i jednom promenljivom. Prema tome, $\forall x. \forall y. x > 0 \wedge y > 0 \implies x + y > 0$ je primer validne formule, dok $\forall x. x \cdot x \geq 0$ to nije.

3 Opis metode

Glavna ideja iza Furijs-Mockinove procedure je da je, pod pretpostavkom određenih aksioma teorije u kojoj se rezonuje, skup linearnih ograničenja moguće

¹Bez narušavanja korektnosti, predstavljena implementacija proširuje signaturu sa ostalim standardnim relacijama jednakosti i nejednakosti.

transformisati u skup ograničenja sa manjim brojem promenljivih od polaznog, pri čemu su transformisani i polazni skup ograničenja ekvizardovoljivi. Nakon eliminacije svih promenljivih iz skupa ograničenja, po jedne u svakoj iteraciji procedure, dobija se skup ograničenja za koji se trivijalno može proveriti da li je zadovoljiv ili ne. Eliminacija neke fiksirane promenljive x_i iz skupa ograničenja se može izvršiti na dva načina — eliminacijom po jednakosti i eliminacijom po nejednakostima.

U nastavku poglavlja će biti opisani osnovni koraci Furije-Mockinove procedure, nakon čega će biti opisano i proširenje procedure za rezonovanje u kontekstu procedure odlučivanja u određenim teorijama prvog reda. Radi veće čitljivosti, u nastavku važi pretpostavka da se procedura primenjuje za rezonovanje u polju racionalnih brojeva \mathbb{Q} .

3.1 Eliminacija promenljive po jednakosti

Eliminacija promenljive x_i po jednakosti podrazumeva da se u skupu ograničenja nalazi ograničenje oblika $\dots + a_i x_i + \dots = b$, $a_i \neq 0$. Tada se promenljiva x_i može izraziti preko ostalih promenljivih koje učestvuju u datoj jednakosti:

$$x_i = \sum_{j \neq i} c_j x_j + c \quad (1)$$

Zbog aksioma jednakosti, opravdano je zameniti sva pojavljivanja promenljive x_i u skupu ograničenja sa desnom stranom jednakosti (1), a da pritom skupovi ograničenja ostanu ekvizardovoljivi. Ukoliko je transformisani skup (sa manje promenljivih) zadovoljiv, onda se vrednost promenljive x_i lako može izračunati na osnovu jednakosti (1). Polazno ograničenje jednakosti, iz koje je dobijena jednakost (1), se onda može eliminisati iz skupa ograničenja.

U nastavku sledi primer izvršavanja ovog koraka. Neka je dat naredni skup linearnih ograničenja:

$$\begin{aligned} x + y + z &= 42 \\ x + 2y &> 12 \\ y + z &< 0 \\ z &> 14 \end{aligned} \quad (2)$$

Pokušajem eliminacije promenljive x iz skupa ograničenja, može se primetiti da je moguće primeniti eliminaciju po jednakosti. Promenljiva x se, na osnovu jednakosti $x + y + z = 42$, može izraziti kao $x = 42 - y - z$. Zamenom ovog izraza u ostalim ograničenjima dobijamo sledeći ekvizardovoljivi skup ograničenja:

$$\begin{aligned} y - z &> -30 \\ y + z &< 0 \\ z &> 14 \end{aligned} \quad (3)$$

3.2 Eliminacija promenljive po nejednakostima

Ukoliko skup ograničenja ne sadrži ni jednu jednakost u kojoj je koeficijent uz promenljivu x_i različit od nule, ona se mora eliminisati na osnovu ograničenja nejednakosti (pretpostavka je da su ove nejednakosti stroge). U tom slučaju,

podskup nejednakosti koje sadrže promenljivu x_i se deli na dva podskupa, C_L i C_G . Skup nejednakosti C_L sadrži sve nejednakosti koje se mogu transformisati u oblik $x_i < \sum_{j \neq i} c_j x_j + c$, a C_G sve nejednakosti koje se mogu transformisati u oblik $x_i > \sum_{j \neq i} d_j x_j + d$. Tada se iz polaznog skupa izbacuju sve nejednakosti iz C_L i C_G , a za svaki par nejednakosti $(C_L, C_G) \in C_L \times C_G$, u novi skup ograničenja se ubacuje nejednakost narednog oblika:

$$\sum_{j \neq i} d_j x_j + d < \sum_{j \neq i} c_j x_j + c \quad (4)$$

Ovako transformisani skup ograničenja ne sadrži promenljivu x_i , ali je i dalje ekvizadovoljiv polaznom, što opravdava gustina racionalnih brojeva, tj. osobina da za svaka dva broja l i g za koje važi $l < g$ postoji i broj s takav da važi $l < s < g$. U specijalnom slučaju da je jedan od skupova C_L i C_G prazan, u kom se ne dodaju nove nejednakosti, već se samo uklanjaju polazne, transformacija je opravdana neograničenošću racionalnih brojeva, tj. osobinom da za svaki racionalni broj s postoje racionalni brojevi l i g takvi da važi $l < s < g$.

Usled dodavanja $|C_L||C_G|$ novih nejednakosti u ovom koraku, vremenska složenost Furiije-Mockinove procedure je eksponencijalna po broju koraka, tj. po broju promenljivih u skupu ograničenja. Iako postoje različite tehnike za poboljšanje performansi procedure, složenost je, i dalje, eksponencijalna [3].

Pošto u skupu ograničenja (3) nema jednakosti, promenljivu y je potrebno eliminisati primenom koraka eliminacije po nejednakostima. Kako se prva nejednakost može prepisati kao $y > z - 30$, a druga kao $y < -z$, eliminacijom polaznih nejednakosti i transformacijom zadatom nejednakošću (4) dobija se novi skup ograničenja:

$$\begin{aligned} 2z &< 30 \\ z &> 14 \end{aligned} \quad (5)$$

U ograničenjima je preostala samo promenljiva z . Ponovom primenom eliminacije promenljive po nejednakostima dobija se trivijalno zadovoljiv (jednočlan) skup ograničenja:

$$0 < 1 \quad (6)$$

3.3 Proširenje do procedure odlučivanja u linearnom fragmentu teorije uređenih polja

U kontekstu logike prvog reda, skup linearnih ograničenja se može posmatrati i kao njihova konjukcija, pri čemu je svako ograničenje neka atomička formula. Kako se Furiije-Mockinovom procedurom eliminišu promenljive iz konjukcije linearnih ograničenja, pri čemu ona ostaje ekvizadovoljiva polaznoj konjukciji, a kvantifikator koji ne vezuje ni jednu promenljivu u formuli logike prvog reda ne utiče na njenu semantiku, ispostavlja se da je Furiije-Mockinova procedura primenljiva i kao procedura odlučivanja za određene teorije prvog reda. U nastavku se pretpostavlja da je reč o linearnom fragmentu teorije uređenih polja.

Neka je data proizvoljna rečenica u preneks normalnoj formi (ukoliko nije u preneks normalnoj formi, lako se može dovesti do nje), pri čemu sve njene atomičke potformule predstavljaju linearna ograničenja. Tada se za eliminaciju kvantifikatora u datoj formuli može primeniti algoritam 1.

Algorithm 1 Eliminacija kvantifikatora Furije-Mockinovom procedurom

```
1: function ELIMINATE_QUANTIFIERS(Formula  $f$ )
2:   if  $f$  is quantifier free then
3:     return  $f$ 
4:   end if
5:   // Let  $f$  be  $Q_1x_1.Q_2x_2....Q_nx_n.F$ 
6:   if  $Q_n$  is  $\forall$  then
7:      $f \leftarrow Q_1x_1.Q_2x_2....\neg\exists x_n.\neg F$ 
8:   end if
9:   // Transform the base formula  $F$  to DNF
10:   $f \leftarrow Q_1x_1.Q_2x_2....\exists x_n.DNF(F)$ 
11:  // Let  $f$  be  $Q_1x_1.Q_2x_2....\exists x_n.C_1 \vee C_2 \vee ... \vee C_k$ 
12:  // where each  $C_i$  is a conjunction of constraints
13:  for  $i$  from 1 to  $k$  do
14:    // Eliminate variable  $x_n$  in  $C_i$  with Fourier-Motzkin
15:     $C'_i \leftarrow \text{var\_elim}(C_i, x_n)$ 
16:     $f \leftarrow Q_1x_1.Q_2x_2....\exists x_n.C_1 \vee ... \vee C'_i \vee ... \vee ... \vee C_k$ 
17:  end for
18:  // Remove the existential quantifier binding  $x_n$ 
19:   $f \leftarrow Q_1x_1.Q_2x_2....Q_{n-1}x_{n-1}.F'$ 
20:  return eliminate_quantifiers( $f$ )
21: end function
```

Po eliminaciji svih promenljivih (a time i kvantifikatora), transformisana formula ostaje ekvizadovoljiva polaznoj, ali se sada trivijalno može interpretirati kao tačna ili netačna. Polazna formula je tada teorema u teoriji uređenih polja ako i samo ako je ovako transformisana formula tačna.

4 Implementacija

Implementacija prethodno opisane Furije-Mockinove procedure javno je dostupna na sledećoj lokaciji: <https://github.com/Aca-S/fourier-motzkin>. Procedura je u celosti implementirana u programskom jeziku C++.

4.1 Organizacija projekta

Celokupna implementacija se nalazi u direktorijumu `source/`. Projekat je podeljen na nekoliko glavnih klasa i skup pomoćnih klasa i funkcija. Glavne klase projekta su:

Constraint<T> — Generička klasa koja predstavlja jedno linearno ograničenje. U zavisnosti od parametra T , ograničenje može biti definisano u različitim uređenim poljima.

ConstraintConjunction<T> — Predstavlja konjukciju linearnih ograničenja. Dva glavna javna metoda ove klase su:

bool is_satisfiable() — Proverava da li je data konjukcija linearnih ograničenja zadovoljiva.

`void eliminate_variable(std::size_t var_index)` — Furije-Mockinovem procedurom eliminiše promenljivu sa zadatim indeksom iz konjukcije linearnih ograničenja.

TheoremProver — Dokazivač teorema u teoriji uređenih polja. Konkretno, svi dokazi se sprovode pod pretpostavkom rada u polju racionalnih brojeva. Glavni metod ove klase je:

`bool is_theorem(const std::string &fol_formula)` — Proverava da li je zadata formula logike prvog reda teorema u teoriji uređenih polja. Ukoliko je tokom konstrukcije **TheoremProver** dokazivača prosleđen izlazni tok (npr. tok standardnog izlaza), na njemu će se ispisivati postupak dokazivanja.

Pomoćne strukture i funkcije pružaju podršku za predstavljanje apstraktnog sintaksnog stabla formula u logici prvog reda (strukture definisane u okviru `fol_ast.hpp` datoteke), parsiranje formula (Klasa **FOLDriver** definisana u `fol_driver.hpp/cpp` i prateći parser i lekser), kao i transformisanje i normalizaciju formula (funkcije definisane u `fol_normalization.hpp/cpp`). Dodatno, klasa **Fraction** pruža podršku za rad sa racionalnim brojevima.

4.2 Prevođenje projekta i upotreba

Projekat se može izgraditi pozicioniranjem u `source` direktorijum i izvršavanjem narednih komandi: `mkdir build && cd build && cmake .. && make`. Po završetku prevođenja, direktorijum `build` će sadržati program `fourier-motzkin` koji se dalje može pokrenuti. Za izgradnju su neophodni C++ kompilator sa podrškom za verziju standarda jezika C++20, kao i instalirani paketi za GNU Bison (minimalna verzija 3.2) i GNU Flex.

Izgrađeni program se pokreće bez argumenata komandne linije. Po pokretanju, na standardnom ulazu se očekuje validna formula za koju je ispituje da li je teorema. Prosleđena formula ne mora biti u preneks normalnoj formi — dokazivač će je sam konvertovati u nju. U slučaju da formula nije rečenica, sve slobodne promenljive u formuli će biti vezane egzistencijalnim kvantifikatorom. Primer upotrebe prikazan je listingom 1.

```
$ ./fourier-motzkin
!x.!y.x > 0 & y > 0 => x + y > 0
===== [PROOF START] =====
[FORMULA] !x.!y.x>0 & y>0 => x+y>0
[CLOSED PRENEX] !x.!y.~x>0 | ~y>0 | x+y>0
[VARIABLE ELIMINATION]
    Eliminating universally bound variable "y"
    Base formula (negated due to universal quantification):
        ~(~x>0 | ~y>0 | x+y>0)
    Base formula DNF:
        x>0 & y>0 & x+y<0 | x>0 & y>0 & x+y=0
    New base formula (negated due to universal quantification):
        ~(0+x>0 & 0+x<0 | 0+x>0 & 0-x>0)
[VARIABLE ELIMINATION]
    Eliminating universally bound variable "x"
    Base formula (negated due to universal quantification):
        ~~(0+x>0 & 0+x<0 | 0+x>0 & 0-x>0)
```

```

Base formula DNF:
    0+x>0 & 0+x<0 | 0+x>0 & 0-x>0
New base formula (negated due to universal quantification):
    ~(0<0 | 0<0)
[QUANTIFIER FREE FORM] ~(0<0 | 0<0)
[RESULT] Formula is a theorem
===== [PROOF END] =====

```

Listing 1: Primer pokretanja fourier-motzkin programa

5 Zaključak

U radu su predstavljeni osnovni pojmovi i definicije koje su bitne u kontekstu rešavanja problema ograničenja i dokazivanja teorema u teoriji uređenih polja. Nakon toga, dat je detaljan opis Furiје-Mockinove procedure za rešavanje konjukcije linearnih ograničenja. Konačno, kao glavni doprinos rada, implementiran je dokazivač teorema u linearnom fragmentu teorije uređenih polja zasnovan na Furiје-Mockinovoj proceduri.

Jedan moguć pravac daljeg razvoja je implementacija različitih heuristika za izbor redosleda eliminacije promenljivih iz konjukcije linearnih ograničenja, čime bi se smanjio efekat kombinatorne eksplozije. Kao dodatna mogućnost, mogu se razmotriti i tehnike za izbegavanje generisanja i čuvanja redundantnih ograničenja (ograničenja koja su posledica ostalih ograničenja u konjukciji).

Literatura

- [1] Schrijver, Alexander, Theory of linear and integer programming. John Wiley & Sons, Inc., 1986.
- [2] Eaves, B. Curtis and Rothblum, Uriel G., Dines–Fourier–Motzkin quantifier elimination and an application of corresponding transfer principles over ordered fields. Mathematical Programming: Series A and B, Volume 53, Issue 1-3. Springer-Verlag, January 1992.
- [3] Rui-Juan Jing and Marc Moreno-Maza and Delaram Talaashrafi, Complexity Estimates for Fourier-Motzkin Elimination, Computer Algebra in Scientific Computing (pp.282-306), 2019.