Univerzitet u Beogradu
Matematički fakultet

Mladen Nikolić

Metodologija izbora pogodnih vrednosti parametara SAT rešavača

— Magistarska teza —

mentor: dr Predrag Janičić

Beograd,
2008.

Sadržaj

Predgovor

U poslednjih deset godina primetan je nagli razvoj sistema za ispitivanje zadovoljivosti iskaznih formula — SAT rešavača. Kako je njihov razvoj od velikog praktičnog značaja, mnogo pažnje se posvećuje povećanju njihove efikasnosti kako bi mogli da se primene na kompleksne iskazne formule, obično one koje dolaze iz primena u industriji. Jedna od mogućnosti za ubrzavanje je prilagođavanje raznih aspekata njihovog funkcionisanja formuli koja se rešava.

Jedna od oblasti koja se brzo razvija sa razvojem sistema za skladištenje i obradu informacija je istraživanje podataka — oblast koja se bavi otkrivanjem znanja u velikim količinama podataka.

Pomenute dve oblasti za sada skoro da nemaju dodirnih tačaka. Međutim, različite familije iskaznih formula pokazuju zakonitosti koje se mogu otkriti metodama istraživanja podataka. Kako napredak u oblasti konstrukcije SAT rešavača odavno zavisi od otkrivanja sve efikasnijih heuristika za razne aspekte njihovog funkcionisanja, postoji širok prostor za uključivanje tehnika iz drugih disciplina uključujući i istraživanje podataka.

Moja primarna naučna interesovanja obuhvataju mašinsko učenje i istraživanje podataka. Posebno su mi zanimljive primene ovih tehnika u drugim značajnim oblastima. Zbog toga mi je ovo istraživanje i bilo privlačno. Kombinovanje tehnika konstrukcije SAT rešavača i istraživanja podataka ne samo što omogućava praktična poboljšanja SAT rešavača, već i spajanje dve zanimljive discipline kojim bi se u daljem radu možda došlo i do novih rezultata u vezi sa problemom zadovoljivosti iskaznih formula.

Mladen Nikolić

Beograd, 10.12.2008.

Glava 1

Uvod

Problem ispitivanja iskazne zadovoljivosti (SAT) je jedan od fundamentalnih matematičkih problema. On ima centralno mesto u teoriji složenosti izračunavanja. Dokazano je da je problem SAT NP-kompletan [4]. Pošto ima široke primene, posvećuje se velika pažnja algoritmima koji bi što efikasnije rešavali instance ovog problema u praksi (pod rešavanjem iskazne formule, podrazumeva se ispitivanje njene zadovoljivosti). Do sada je razvijen veliki broj *SAT rešavača*, sistema koji implementiraju ovakve algoritme.

SAT rešavači se mogu svrstati u dve grupe — kompletne i stohastičke. Za datu iskaznu formulu kompletni SAT rešavači sigurno nalaze zadovoljavajuću valuaciju ili utvrđuju da ona ne postoji, dok stohastički rešavači ne mogu dokazati nezadovoljivost formule iako mogu dokazati njenu zadovoljivost. Njihova prednost je u tome što za neke klase formula, mogu utvrditi zadovoljivost brže od kompletnih rešavača. U ovom radu biće razmatrani samo kompletni SAT rešavači.

Moderni kompletni SAT rešavači su najčešće zasnovani na proceduri DPLL [6, 5], objavljenoj ranih šezdesetih godina. SAT rešavači postižu veliki napredak na prelazu prošlog u ovaj vek, kada sa pojavljuju poznati rešavači kao SATO, CHAFF, MiniSAT i drugi. Ovaj napredak je posledica kako konceptualnih, tako i implementacionih unapređenja procedure DPLL.

SAT rešavači najčešće imaju nekoliko parametara koji određuju njihovo funkcionisanje. Efikasnost SAT rešavača u velikoj meri zavisi od konkretne instance problema (tj. iskazne formule čija se zadovoljivost ispituje) i od konkretnih vrednosti parametara. Zbog toga je veoma važno pitanje izbora vrednosti parametara rešavača, posebno u zavisnosti od instance problema koja se rešava.

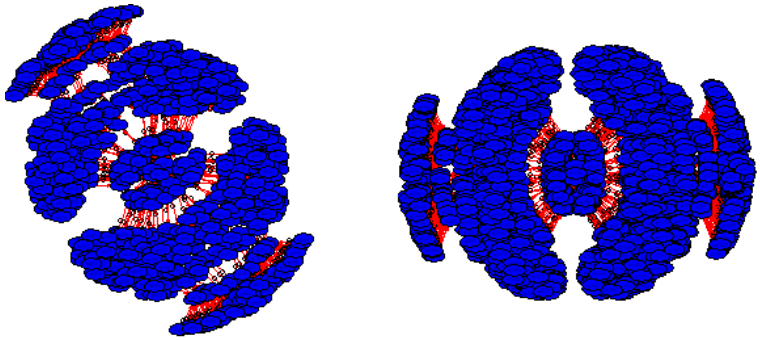
Skupovi parametara kojima SAT rešavači raspolažu variraju od implementacije do implementacije. Ti parametri se odnose na različite aspekte funkcionisanja SAT rešavača. Uobičajeno je da svaki od aspekata funkcionisanja bude određen nekom politikom. Na primer, s vremena na vreme rešavanje formule otpočinje iznova za slučaj da se pretraga vrši u delu pros-

tora u kome se ne nalazi zadovoljavajuća valuacija. Politike otpočinjanja iznova mogu biti različite. Najjednostavnija je da otpočinjanja iznova uopšte nema i ona nema parametara. Jedna česta politika podrazumeva eksponencijalno povećavanje vremenskog intervala posle kojeg se vrši otpočinjanje rešavanja iznova. U tom slučaju parametri su dužina polaznog intervala i umnožak kojim se on povećava. Kada se u nastavku teksta bude govorilo o parametrima SAT rešavača neće se nužno podrazumevati parametri nekih politika. Naime, i izbor politike se može smatrati parametrom i to često važnijim od parametara koji finije definišu ponašanje politike.

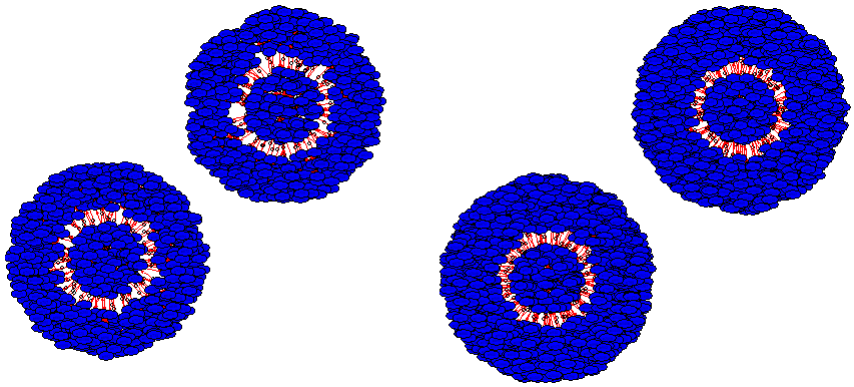
Iskazne formule čiju je zadovoljivost potrebno proveriti često dolaze iz primena u kojima se iskazna logika koristi za modelovanje stvarnih fenomena, uređaja i slično. Zbog toga je opravdano pretpostaviti da sličnosti u njihovom poreklu impliciraju i sličnosti u vrednostima parametara koje treba primeniti prilikom njihovog rešavanja kako bi se dobilo na brzini. Stoga bi za SAT rešavač bilo korisno da za različite familije formula istog porekla raspolaže vrednostima parametara koje se ponašaju u proseku najbolje na ovim, manjim skupovima, a da se familija kojoj formula pripada automatski prepoznaje kako bi se primenile pogodne vrednosti parametara. Posebno zanimljiv bi mogao da bude granični slučaj planirane klasifikacije u kojem bi svaka formula mogla da razmatra kao jednočlana klasu.

Formule koje dolaze iz primena, a pripadaju različitim familijama imaju različite i, u pogodnoj grafovskoj reprezentaciji, vizuelno prepoznatljive strukture. Postoji veći broj načina reprezentovanja formula grafovima. Takođe, postoji i veći broj načina vizualizovanja grafova. Stoga ovo zapažanje predstavlja samo neformalnu motivaciju za dalje razmatranje, a ne dokaziv argument. Ipak, vizualizacija grafovske strukture formule je tema kojoj je u literaturi već poklonjena pažnja [34]. Pretpostavka ovog rada je da se ta struktura može na neki način automatski prepoznati i da se na osnovu nje može postići visok nivo preciznosti klasifikacije. Kao ilustraciju prepoznatljivosti prikaza grafovskih struktura koje odgovaraju formulama, navodimo nekoliko primera. Razmatrane formule pripadaju korpusu sastavljenom za takmičenje SAT rešavača, SAT Competition 2002, a odgovarajući grafovi su vizuelizovani pomoću paketa GRAPHVIZ. Na slikama 1.1, 1.2 i 1.3 su prikazane grafovske reprezentacije po dve formule iz tri familije. Neformalno govoreći, očigledno je da prikazi grafova koji predstavljaju formule koje pripadaju istoj familiji imaju sličan oblik, iako sa različitim brojem čvorova.

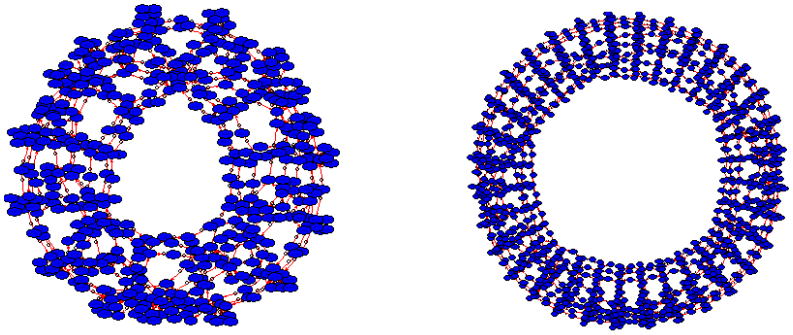
Formula je najčešće zapisana u vidu teksta. Jedna od zanimljivih mogućnosti je predstavljanje iskaznih formula pomoću grafova, tako da se problem klasifikovanja formula svodi na problem klasifikovanja grafova. Motivacija za ovakav pristup u poređenju sa tekstualnom reprezentacijom je komutativnost veznika konjunkcije i disjunkcije, kao i invarijantnost zadovoljivosti formule u odnosu na preoznačavanje promenljivih. Ove promene značajno menjaju izgled formule kao teksta, dok se u reprezentaciji pomoću grafa



Slika 1.1: Prikaz formula bart10.shuffled.cnf sa 144 promenljive i 560 klauza i bart20.shuffled.cnf sa 270 promenljivih i 1476 klauza. Obe formule potiču iz problema verifikacije hardvera.



Slika 1.2: Prikaz formula homer06.shuffled.cnf sa 180 promenljivih i 830 klauza i homer09.shuffled.cnf sa 270 promenljivih i 1920 klauza. Obe formule potiču iz problema verifikacije hardvera.



Slika 1.3: Prikaz formula rope_0003.shuffled.cnf sa 108 promenljivih i 252 klauze i rope_0008.shuffled.cnf sa 288 promenljivih i 672 klauze. Obe formule potiču iz problema bojenja grafova.

menja samo njegovo označavanje, ali ne i struktura. Naravno, moguće su i drugačiji pristupi klasifikovanju formula.

Vrednosti parametara SAT rešavača se mogu birati na različite načine i često se biraju neke koje su se često u praksi pokazale kao dobre. Ovo očigledno ne mora biti najbolje rešenje: zato što na taj način izabrane vrednosti parametara ne moraju stvarno biti najbolje ni u kom smislu i, još važnije, zato što različitim iskaznim formulama mogu odgovarati različite vrednosti parametara za koje te formule bivaju efikasno rešene. Stoga se bolja metodologija izbora vrednosti parametara SAT rešavača može sastojati grubo iz dva koraka:

- Sistematično rešavanje nekog reprezentativnog korpusa iskaznih formula za različite vrednosti relevantnih parametara u fazi treniranja rešavača u cilju određivanja dobrih vrednosti parametara za pojedinačne formule.
- Inteligentno biranje pogodnih vrednosti parametara u fazi eksploatacije rešavača, u skladu sa određenim karakteristikama formule koja se rešava.

Cilj ovog rada je formulisanje jedne ovakve metodologije i testiranje njene efikasnosti na osnovu više mera kvaliteta. Najvažniji aspekti ovakve metodologije su:

Izbor parametara čiji bi se uticaj na rešavanje formule razmatrao. Broj parametara koji bi se mogli uzeti u obzir je veliki. Idealan pristup bi podrazumevao bavljenje svim parametrima svih rešavača koji su na raspolaganju. Međutim, ovakav pristup je računski vrlo zahtevan. Naime, broj kombinacija različitih vrednosti parametara, za koje, prema prvom koraku metodologije, treba rešavati formule, eksponencijalno raste sa brojem parametara koji se uzimaju u obzir. Stoga se, iz praktičnih razloga, pažnja mora obratiti samo na one parametre za koje se proceni da imaju najveći uticaj na efikasnost rešavanja.

Izbor dopustivih vrednosti parametara koji se smatraju relevantnim. Povećanje broja dopustivih vrednosti parametra povećava šanse da se nađe bolja vrednost parametra za rešavanje neke formule. S druge strane, kako ukupan broj kombinacija vrednosti parametara za koje se formula rešava predstavlja proizvod brojeva dopuštenih vrednosti za svaki od parametara, iz praktičnih razloga, potrebno je težiti manjem broju dopuštenih vrednosti po parametru. Potrebno je naći balans između ova dva suprotstavljena zahteva. Kada se u nastavku teksta bude govorilo o optimalnim vrednostima parametara za jednu formulu, misliće se na najbolju kombinaciju vrednosti parametara u izabranom skupu vrednosti. Smatra se da

je jedna kombinacija vrednosti parametara bolja od druge za datu formulu, ako ona za tu kombinaciju biva rešena za kraće vreme nego za drugu.

Izbor korpusa formula na kojem bi se vršili treniranje i evaluacija SAT rešavača koji bi implementirao predloženu metodologiju. Kako bi rešavač koji implementira grubo formulisanu metodologiju bio što primenljiviji u praksi i kako bi sama metodologija bila što pouzdanije testirana, potrebno je da skup formula na kojem se vrši treniranje rešavača bude što reprezentativniji. Iskazne formule se često mogu grupisati u familije. Na primer, formule za verifikaciju različitih hardverskih komponenti obično čine različite familije. Reprezentativnost korpusa se može popravljati zastupljenošću što većeg broja familija formula, kao i zastupljenošću celog spektra težine ovih formula — od trivijalnih koje se rešavaju u deliću sekunde do onih koje su na granici praktične rešivosti. Ovakvi korpusi su već sastavljeni, na primer, za takmičenja SAT rešavača (koja obično prate značajne konferencije iz ove oblasti).

Identifikacija karakteristika iskazne formule na osnovu kojih bi se vršio izbor pogodnih vrednosti parametara. Ovaj aspekt rešavanja iskaznih formula do sada nije detaljnije analiziran u literaturi. Potrebno je naći neke karakteristike koje bi u zadovoljavajućem stepenu bile zajedničke za formule koje bivaju efikasno rešene za iste kombinacije vrednosti parametara. Pri tome je pogodno da te karakteristike budu invarijantne u odnosu na preimenovanje promenljivih, zamenu mesta literala i zamenu mesta klauza u formuli, jer ove transformacije menjaju formulu, ali ne utiču na njenu zadovoljivost. Očekivano je da formule iz iste familije imaju slične karakteristike, kao i da dele vrednosti parametara za koje bivaju efikasno rešene.

Izbor mera kvaliteta na osnovu kojih bi se vršila evaluacija metodologije. Mere kvaliteta moraju biti na neki način relevantne za ocenu upotrebljivosti predložene metodologije. U tom cilju, biće usvojene neke koje se već koriste u literaturi i izabrane nove u skladu sa potrebama istraživanja.

Navedni izbori mogu biti učinjeni za proizvoljan rešavač. Predložena metodologija je opšteg tipa i prilikom njenog konkretnijeg formulisanja potrebno je očuvati tu opštost.

Važne hipoteze ovog rada su:

- Među formulama iz iste familije postoji sintaksna sličnost koja se može automatski prepoznati. Ovo se može iskoristiti za automatsko prepoznavanje familije kojoj formula pripada.

- Za skupove formula koje su sintaksno slične postoje dominantno najbolje vrednosti parametara.
- Za sintaksno slične formule, najbolje kombinacije vrednosti parametara iz dopustivog skupa su takođe slične.

Osnovna teza ovog rada je *da se inteligentnim biranjem vrednosti parametara SAT rešavača, zasnovanim na analizi sintakse formule koja se rešava, može povećati njegova efikasnost.*

U ovoj oblasti postoji nekoliko značajnih, uglavnom skorašnjih, rezultata, ali postoji dosta prostora za dalja istraživanja. Kako se ovaj rad oslanja ne samo na rezultate u okvirima logike i konstrukcije SAT rešavača, relevantni su i radovi iz drugih oblasti. U radu [25] je opisan način rangiranja strategija koje se koriste u dokazivanju teorema. Ono se vrši na osnovu procene mogućeg doprinosa strategije koja se računa na osnovu nekoliko unapred određenih kriterijuma. Rangiranje zavisi od polaznog, fiksiranog rangiranja, složenosti onoga što treba dokazati i ocene mogućeg doprinosa strategije. Primena metoda mašinskog učenja za ubrzavanje DPLL procedure može se naći u radu [21]. Određeno ubrzanje postignuto je učenjem izbora literala na koji treba primeniti pravilo split DPLL algoritma. To se postiže tako što se problem rešavanja razmatra kao Markovljev proces odlučivanja, što omogućava primenu metoda *učenja uslovljavanjem* (eng. reinforcement learning). Više o ovim pristupima biće rečeno u glavi 3. Verovatno najrelevantniji radovi u ovoj oblasti su [14] i [38]. U ovim radovima korišćenjem linearne regresije, na osnovu velikog broja pokretanja SAT rešavača aproksimira se funkcija koja omogućava predviđanje vremena rešavanja instance sa određenim parametrima ili pomoću određenog rešavača. Pretpostavlja se da je poznato kojoj familiji formula pripada. Više o ovim radovima biće rečeno u glavi 3. Još jedan rad koji se tiče parametara SAT rešavača [13] bavi se optimizacijom parametara pomoću metode optimizacije koja se može smatrati pristrasnim slučajnim hodom po lokalnim optimumima. Ovaj pristup se ipak ne bavi prilagođavanjem parametara konkretnoj instanci, već nalaženjem najboljih parametara za neku grupu formula. U radu [18] je opisan metod utvrđivanja frekvencija podgrafova, koji se koristi za pravljanje profila grafova. Pošto je utvrđivanje tačnih frekvencija podgrafova računski neisplativo, problem se rešava računanjem na slučajnom uzorku. Ispostavlja se da ocene frekvencija brzo konvergiraju pravim frekvencijama. Ovaj pristup biće detaljnije opisan u glavi 4.

Nastavak ovog teksta je organizovan na sledeći način:

U glavi 2 izloženi su osnovni pojmovi koji se koriste u definisanju predložene metodologije. Ukratko su opisani iskazna logika sa fokusom na prob-

lemu SAT, SAT rešavači i sistem ArgoSAT, osnovni pojmovi koji se odnose na grafove, istraživanje podataka i problem klasifikacije, kao i neke osnovne uzoračke statistike korišćene u ovom radu.

Glava 3 opisuje relevantne radove i tehnike koji prethode ovom istraživanju. Opisano je nekoliko radova vrlo bliskih po tematici, ali je dat i širi osvrt na prilagođavanje SAT rešavača.

Glava 4 sadrži opis novih metoda za klasifikovanje iskaznih formula. Kao najjednostavniji pristup klasifikovanju koristi se klasifikovanje formula na osnovu njihovih n-gramskih profila kad se one razmatraju kao običan tekst. Jedan pristup klasifikovanju se zasniva i na skupu sintaksnih svojstava preuzetom iz literature [31]. Klasifikovanje grafova bazira se na frekvencijama podgrafova koji se javljaju u grafovima.

U glavi 5 detaljno je opisana metodologija izbora vrednosti parametara SAT rešavača. Definiše se skup razmatranih parametara i njihove dopustive vrednosti. U nastavku, u glavi 6, opisuje se implementacija sistema koji se bazira na predloženoj metodologiji, zatim se opisuje korpus koji je korišćen, kao i dizajn eksperimenata i principi evaluacije metodologije. Na kraju su dati eksperimentalni rezultati.

Glava 7 opisuje dalje pravce istraživanja koji iz ovog rada proističu ili se na njega oslanjaju. U glavi 8 su sumirani zaključci koji se mogu izvesti iz dobijenih eksperimentalnih rezultata.

Glava 2

Osnovni pojmovi, notacija i tehnike

U ovoj glavi biće dat opis teorijskih i praktičnih osnova ovog rada. Biće objašnjeni osnovni pojmovi, utvrđena notacija i opisano nekoliko postojećih tehnika istraživanja podataka.

Problem zadovoljivosti iskaznih formula (SAT) pripada oblasti matematičke logike. Konstrukcija SAT rešavača već predstavlja zasebnu, dobro razvijenu matematičko-informatičku disciplinu. Tehnike koje su iskorišćene u ovom radu pripadaju i drugim oblastima. Metode istraživanja podataka se koriste za analizu grafovske strukture formule, ocenjivanje njihove sličnosti i klasifikovanje.

2.1 Iskazna logika i problem SAT

Iskazna logika se bavi istinitošću iskaza, izgrađenih nad iskaznim promenljivim pomoću određenih logičkih veznika. Iskazne promenljive predstavljaju elementarne iskaze. Način izgradnje složenijih iskaza iz njih je definisan *sintaksom* iskazne logike. Značenje konstruisanih iskaza predstavlja *semantiku* iskazne logike. Osnovni problem iskazne logike — problem SAT je problem ispitivanja da li je iskazna formula zadovoljiva. Postoji veći broj načina na koje se ovaj problem može rešavati.

Definicije osnovnih pojmova i DPLL algoritma su izložene u skladu sa [16].

Definicija 1 Skup iskaznih formula *nad prebrojivim skupom iskaznih slova* P *je skup za koji važi:*

- *iskazna slova (iz skupa* P*) i logičke konstante (*$\top$ *i* $\bot$*) su iskazne formule;*
- *ako su* A *i* B *iskazne formule, onda su i* $(\neg A)$, $(A \wedge B)$ *i* $(A \vee B)$ *iskazne formule.*

- *iskazne formule mogu se dobiti samo konačnom primenom prethodna dva pravila.*

Kako je rešavanje problema SAT obično vezano za konjunktivnu normalnu formu formule, veznici $\Rightarrow$ i $\Leftrightarrow$ se ne koriste jer je skup veznika $\{\neg, \wedge, \vee\}$ koji se koristi u konjunktivnoj normalnoj formi potpun.

Zagrade često izostavljamo i oslanjamo se na prioritet operatora u sledećem poretku od najvišeg ka najnižem: $\neg$, $\wedge$, $\vee$.

Iskazna slova nazivamo i *iskaznim promenljivim*, a iskazna slova i logičke konstante nazivamo *atomičkim iskaznim formulama*. Atomičke iskazne formule i njihove negacije nazivamo *literalima*. Disjunkcije literala nazivamo *klauzama*.

Funkcije koje preslikavaju skup iskaznih slova u skup $\{0, 1\}$ se nazivaju *valuacijama*. Polazeći od valuacije v konstruiše se funkcija I_v koja preslikava skup iskaznih formula u skup $\{0, 1\}$. Nju nazivamo *interpretacijom* za valuaciju v i definišemo na sledeći način:

- $I_v(p) = v(p)$, za svaki element p skupa P;
- $I_v(\top) = 1$ i $I_v(\bot) = 0$;
- $I_v(\neg A) = 1$ ako je $I_v(A) = 0$ i $I_v(\neg A) = 0$ ako je $I_v(A) = 1$;
- $I_v(A \wedge B) = 1$ ako je $I_v(A) = 1$ i $I_v(B) = 1$; $I_v(A \wedge B) = 0$ inače;
- $I_v(A \vee B) = 0$ ako je $I_v(A) = 0$ i $I_v(B) = 0$; $I_v(A \vee B) = 1$ inače;

Definicija 2 *Iskazna formula F je* zadovoljiva *ako postoji valuacija v takva da je $I_v(F) = 1$. U suprotnom, F je* nezadovoljiva. *Iskazna formula F je* valjana *(*tautologija*) ako za svaku valuaciju v važi $I_v(F) = 1$. U suprotnom, F je* poreciva.

Definicija 3 *Iskazna formula je u* konjunktivnoj normalnoj formi *(KNF) ako je oblika*

$$A_1 \wedge A_2 \wedge \ldots \wedge A_n$$

pri čemu je svaka od formula A_i $(1 \leq i \leq n)$ klauza.

Osnovni algoritam za proveru zadovoljivosti iskaznih formula je Davis-Patnam-Logman-Lovelandova procedura (DPLL procedura). Formula čija se zadovoljivost ispituje mora biti u konjunktivnoj normalnoj formi. Svaka klauza se razmatra kao multiskup[1] literala, dok se formula razmatra kao multiskup klauza. DPLL procedura je data na slici 2.1.

[1] *Multiskup* je uređeni par $\mathcal{M} = (A, m)$, gde je A skup, a $m : A \to \mathbf{N}^+$ funkcija iz skupa A u skup pozitivnih prirodnih brojeva. Skup A se naziva *skupom nosačem*. Za svako $a \in A$, vrednost $m(a)$ predstavlja broj pojavljivanja elementa a u multiskupu M. Kažemo da element a pripada multiskupu M i pišemo $a \in \mathcal{M}$ ako $a \in A$.

Algoritam: DPLL

Ulaz: Multiskup klauza D ($D = \{C_1, C_2, \ldots, C_n\}$)
Izlaz: *DA*, ako je multiskup D zadovoljiv;
NE, ako multiskup D nije zadovoljiv

1. Ako je D prazan, vrati *DA*.
2. Zameni sve literale $\neg\bot$ sa $\top$ i zameni sve literale $\neg\top$ sa $\bot$.
3. Obriši sve literale jednake $\bot$.
4. Ako D sadrži praznu klauzu, vrati *NE*.
5. Ako neka klauza C_i sadrži literal $\top$ ili sadrži i neki literal i njegovu negaciju, vrati vrednost koju vraća DPLL($D \setminus C_i$) (eng. *tautology*).
6. Ako je neka klauza jedinična i jednaka nekom iskaznom slovu p, onda vrati vrednost koju vraća DPLL($D[p \mapsto \top]$); ako je neka klauza jedinična i jednaka $\neg p$, gde je p neko iskazno slovo, onda vrati vrednost koju vraća DPLL($D[p \mapsto \bot]$) (eng. *unit propagation*).
7. Ako D sadrži literal p (gde je p neko iskazno slovo), a ne i literal $\neg p$, onda vrati vrednost koju vraća DPLL($D[p \mapsto \top]$); ako D sadrži literal $\neg p$ (gde je p neko iskazno slovo), a ne i literal p, onda vrati vrednost koju vraća DPLL($D[\neg p \mapsto \top]$) (eng. *pure literal*).
8. Ako DPLL($D[p \mapsto \top]$) vraća *DA*, onda vrati *DA*; inače vrati vrednost koju vraća DPLL($D[p \mapsto \bot]$) (gde je p jedno od iskaznih slova koja se javljaju u D) (eng. *split*).

Slika 2.1: DPLL procedura.

Ovaj algoritam se može koristiti i za proveru da li je formula valjana, tako što se krene od negacije polazne formule i ako je ona nezadovoljiva, zaključuje se da je polazna formula valjana.

2.2 SAT rešavači i sistem ArgoSAT

Ovaj pregled osnovnih koncepata SAT rešavača, zasnovan je na radovima [8], [30] i [7].

Iako je problem SAT NP-kompletan problem, zahvaljujući napretku modernih SAT rešavača, moguće je u prihvatljivom vremenu rešavati mnoge instance ovog problema koje sadrže i milione promenljivih. Stoga su njihove primene u industriji vrlo česte — u verifikaciji hardvera, automatskom raspoređivanju, planiranju i drugim problemima.

Postoje dve osnovne vrste SAT rešavača — kompletni i stohastički. Većina kompletnih SAT rešavača je zasnovana na proceduri DPLL. Oni mogu da ustanove zadovoljivost, odnosno nezadovoljivost bilo koje iskazne formule, dok stohastički rešavači mogu samo da ustanove zadovoljivost formule, ali ne i njenu nezadovoljivost. S druge strane, stohastički rešavači često mogu rešiti zadovoljive formule za dosta manje vremena od kompletnih rešavača.

Neki od najznačajnijih stohastičkih SAT rešavača su WalkSAT [32, 28], GSAT [33] i SAPS [15]. Princip na kome funkcioniše prvi ovakav rešavač — GSAT je gramziva lokalna pretraga. Polazeći od slučajno generisane valuacije razmatraju se sve valuacije koje se od nje razlikuju u verdnosti najviše jedne promenljive i kao sledeći čvor pretrage bira se ona koja zadovoljava najveći broj klauza formule koja se rešava. Inovacija koju uvodi WalkSAT se sastoji u tome da se sa verovatnoćom p slučajno izabere jedna promenljiva u formuli i da se promeni vrednost koja joj je dodeljena, a da se sa verovatnoćom $1-p$ kao kod rešavača GSAT izabere najbolja valuacija koja se od trenutne razlikuje u dodeli vrednosti jedne promenljive. Navedene ideje su vrlo jednostavne, ali se u praksi koriste i kompleksnije ideje kao u slučaju rešavača SAPS. Mogući su i pristupi pomoću metaheuristika optimizacije kao što su genetski algoritmi [11, 9], metoda promenljivih okolina [29] i druge. Pomoću njih se zadovoljavajuća valuacija nalazi tako što se traži valuacija sa maksimalnim brojem zadovoljenih klauza.

Neki od najznačajnijih kompletnih SAT rešavača su MiniSAT, CHAFF, PICOSAT, SATO i drugi. SAT rešavač ArgoSAT [27] je deo paketa ARGOLIB — biblioteke, pisane u jeziku C++, koja pruža podršku korišćenju procedura odlučivanja. Zasnovan je na DPLL proceduri i predstavlja ponovnu implementaciju MiniSAT rešavača kojom je postignuta proširivost i fleksibilnost rešavača, kao i čitljivost njegovog kôda. Korektnost osnovnih delova ovog rešavača je i formalno dokazana [26].

U daljem tekstu će se pod SAT rešavačima podrazumevati kompletni SAT rešavači.

2.2.1 DPLL procedura i njene moderne varijante

Većina modernih SAT rešavača se zasniva na DPLL proceduri, ali često sa kompleksnim izmenama radi povećanja efikasnosti. Jedna od bitnih izmena je da se ne vrše transformacije nad formulom, već se u koracima konstruiše valuacija u kojoj bi formula trebalo da bude tačna. Zamena promenljive konstantom $\top$ ili $\bot$ u *split* pravilu indukuje vrednost te promenljive u valuaciji koja se generiše u toku procesa rešavanja i ta vrednost može biti 1 ili 0. Izbor konkretne vrednosti se naziva *odlukom* (eng. decision). Prilikom odluke, potrebno je izabrati promenljivu čija će vrednost biti zadata, kao i inicijalni *polaritet* (eng. polarity) literala, tj. odrediti da li pri odluci promenljiva prvo dobija vrednost 0 ili 1. Za negiranu promenljivu kažemo da ima negativan polaritet u literalu, dok za onu koja to nije, kažemo da ima pozitivan polaritet. Kada se ispostavi (na primer, u slučaju pravila *unit propagation*) da neka promenljiva mora imati određenu vrednost, radi se o *implikaciji* (eng. implication). Situacija u kojoj neka promenljiva u toku rešavanja ima implicirane obe vrednosti (i 0 i 1), naziva se *konfilkt* (eng. conflict). Kada se konflikt desi, potrebno je izvršiti promenu neke odluke kako bi se pokušala naći neka valuacija koja zadovoljava formulu. Takođe je moguće tražiti razloge zbog kojih je do konflikta došlo. Oni se formulišu u obliku *naučenih klauza* (eng. learnt clauses) koje se dodaju u početni skup klauza, a konstruišu se nekim mehanizmom rezonovanja (na primer iskaznom rezolucijom), na osnovu informacija koje su raspoložive o konfliktu. Ako je klauza $\{x, y, z\}$ postala nezadovoljiva u toku rešavanja, onda $\neg x \wedge \neg y \wedge \neg z$ nazivamo *razlogom konflikta* (eng. reason set of the conflict) [7]. Vrednost 0 je implicirana za promenljivu x, na primer, tako što je prisutna klauza $\{\neg u, \neg v, \neg x\}$, a promenljive u i v imaju vrednost 1. Odatle sledi da i $u \wedge v \wedge \neg y \wedge \neg z$ takođe mora voditi ka konfliktu. Taj konflikt se može sprečiti dodavanjem klauze $\{\neg u, \neg v, y, z\}$ za koju kažemo da je *naučena* (eng. learnt). Kako broj naučenih klauza obično raste, njihovo analiziranje opterećuje rešavač, pa stoga ove klauze u nekom trenutku treba brisati. Ovaj postupak se naziva *zaboravljanje*. Kako je drvo pretrage koju DPLL procedura sprovodi veliko, rešavač može veliku količinu vremena potrošiti pretražujući grane koje ne sadrže rešenje. Da bi se to sprečilo, proces rešavanja s vremena na vreme *otpočinje iznova* (eng. restart). Da otpočinjanje iznova ne bi ugrozilo potpunost pretrage ono se obično vrši sve ređe i ređe u toku rešavanja.

Razni aspekti ponašanja SAT rešavača se mogu preciznije definisati i to se postiže uvođenjem različitih heuristika. Za neke od tih aspekata će biti detaljnije opisane korišćene heuristike.

SAT rešavači obično tvrdo kodiraju politike koje koriste za upravljanje glavnim aspektima svog funkcionisanja radi dobitaka u brzini. Osnovna teza

praćena prilikom konstrukcije ArgoSAT rešavača je da povećanje efikasnosti koje nose inteligentne heuristike daleko prevazilazi dobitke od sitnih trikova u implementaciji[2]. Stoga je usvojen modularni pristup koji omogućava lako dodavanje novih politika kao parametara. Ovo je osnovni kvalitet zbog koga je ArgoSAT izabran kao rešavač za ovo istraživanje. U nastavku su opisane neke politike koje ArgoSAT implementira, a koje su relevantne za ovo istraživanje.

2.2.2 Politike izbora promenljive

Politike izbora promenljive se odnose na izbor promenljive u vezi sa čijom vrednošću će biti doneta odluka. U nastavku će biti opisane neke od njih.

Jednostavna politika je politika pseudoslučajnog izbora jedne od promenljivih čija vrednost u datom trenutku nije definisana nekom prethodnom odlukom niti implicirana. Ovu strategiju ćemo u daljem tekstu označavati sa VS_RANDOM. Ova politika se može uopštiti tako što bi se dozvolilo da se u određenom broju biranja izbor promenljive vrši u skladu sa VS_RANDOM politikom, a u ostalim, prema alternativnoj ponuđenoj politici. Za zadatu verovatnoću izbora u skladu sa slučajnom politikom p ($0 \leq p \leq 1$), ovakvu politiku ćemo dalje označavati sa VS_RANDOM p ALTERNATIVNA_POLITIKA.

Nešto složeniju politiku koristi rešavač MiniSAT. Ona koristi dinamičko rangiranje promenljivih prema njihovom učešću u skorašnjim konfliktima. Za svaku promenljivu se čuva *faktor aktivnosti*. Prilikom svakog konflikta, svim promenljivim koje učestvuju u njemu povećava se faktor aktivnosti za neku vrednost `bumpAmount`. Takođe prilikom konflikta, `bumpAmount` se množi koeficijentom `decayFactor` koji je veći od 1. Na taj način promenljive koje su učestvovale u skorijim konfliktima imaju veće faktore aktivnosti. S vremena na vreme, svi faktori aktivnosti se skaliraju. Politika je opisana u pseudokodu koji sledi, a u nastavku teksta ćemo je označavati sa VS_MINISAT.

```
int selectVariable() {
    int i;
    int maxActivityVariable = 0;

    for(i=1; i < variableCount; i++)
        if(activity[maxActivityVariable] < activity[i])
            maxActivityVariable = i;

    return maxActivityVariable;
}

void onConflict(Clause c) {
    varDecayActivity();
```

[2]Lična komunikacija sa Filipom Marićem, glavnim autorom sistema ArgoSAT.

```
    foreach v in c
        bumpVariableActivity(v);
}

void varDecayActivity() {
    bumpAmount *= decayFactor;
}

void bumpVariableActivity(int n) {
    if((activity[x] += bumpAmount) > MAX_ACTIVITY)
        rescaleVariableActivities();
}

void rescaleVariableActivities() {
    int i;

    for(i=0; i < variableCount; i++)
        activity[i] *= (1.0 / MAX_ACTIVITY);

    bumpAmount *= (1.0 / MAX_ACTIVITY);
}
```

Problem sa politikom VS_MINISAT je što na početku rešavanja ni jedna promenljiva nema prednost nad nekom drugom. Jednostavna modifikacija koja se u praksi pokazala kao korisna je da se na početku faktor aktivnosti svake promenljive inicijalizuje na broj njenih pojavljivanja u formuli. Ovako modifikovanu politiku zvaćemo VS_MINISAT_INIT.

2.2.3 Politike izbora polariteta promenljive

Kada je izabrana promenljiva u vezi sa čijom vrednošću će biti doneta odluka, potrebno je izabrati i njen polaritet, tj. odlučiti da li će joj prilikom odluke biti prvo dodeljivana vrednost 0 ili 1. Najjednostavnije su konstantne politike. Pod PS_TRUE ćemo podrazumevati politiku koja uvek bira pozitivan polaritet za promenljivu, dok će PS_FALSE označavati politiku koja uvek bira negativan polaritet. Politiku koja se bazira na slučajnom izboru, tako da se sa verovatnoćom p odlučuje za pozitivan polaritet, a inače za negativan, označavaćemo sa PS_RANDOM p.

Nešto složenija politika, PS_POLARITY_SAVING, koja se u praksi pokazala vrlo uspešnom, bira polaritet promenljive koji je za nju bio poslednji put izabran u odluci ili impliciran. Za polazni polaritet se u izvornoj verziji politike bira negativan. Kako na ovaj način postoji mogućnost da se negativan polaritet predugo zadrži za veliki broj promenljivih i da tako ova politika

bude previše slična politici PS_FALSE, ona se često modifikuje tako da se početni polaritet promenljive inicijalizuje na pozitivan ako ona u formuli češće učestvuje bez negacije, a na negativan u suprotnom. Ovakvu politiku označavamo sa PS_POLARITY_SAVING_INIT.

2.2.4 Politike otpočinjanja iznova

Najjednostavnija politika otpočinjanja iznova je da se ono ne koristi. Nju ćemo nazivati RS_NO_RESTART. Druge politike koje će biti opisane se zasnivaju na brojanju konflikata do sledećeg otpočinjanja iznova.

Politika koju koristi MiniSAT zahteva da se kao parametar zada polazni broj konflikata do otpočinjanja iznova `numConflictsForFirstRestart`. Posle svakog otpočinjanja iznova, broj potrebnih konflikata se povećava množenjem sa drugim parametrom `restartInc` koji treba da bude veći od 1. Ovu politiku nazivamo RS_MINISAT. Opisana je pseudokodom u nastavku.

```
void init() {
    numConflictsForNextRestart = numConflictsForFirstRestart;
}

bool shouldRestart() {
    return numConflicts >= numConflictsForNextRestart;
}

void onRestart() {
    numConflicts = 0;
    numConflictsForNextRestart *= restartInc;
}

void onConflict() {
    numConflicts++;
}
```

Zbog eksponencijalnog rasta broja potrebnih konflikata posle kojeg dolazi do otpočinjanja iznova pri ovoj politici, ono posle određenog vremena praktično nestaje. Zbog toga su formulisane dve politike brzog otpočinjanja iznova. Lubijeva politika [24], RS_LUBY, vrši i-to otpočinjanje iznova posle $t_i * \texttt{sizeFactor}$ konflikata od prethodnog. Pri tome je `sizeFactor` parametar politike, dok je niz t_i definisan na sledeći način:

$$t_i = \begin{cases} 2^{k-1} & ako\ je\ i = 2^k - 1 \\ t_{i-2^{k-1}+1} & ako\ je\ 2^{k-1} \leq i \leq 2^k - 1 \end{cases}$$

Prvih nekoliko elemenata niza t_i su:

$$1, 1, 2, 1, 1, 2, 4, 1, 1, 2, 1, 1, 2, 4, 8, 1, \ldots$$

Arminova politika [17], RS_ARMIN, kao parametre traži početni broj konflikata do otpočinjanja iznova `numConflictsForFirstRestart` i faktor kojim se ovaj broj množi — `restartInc`. Množenje se vrši posle svakog otpočinjanja iznova, sve dok broj potrebnih konflikata ne pređe određeno ograničenje. U tom trenutku se to ograničenje množi sa `restartInc`, a broj potrebnih konflikata se ponovo postavlja na `numConflictsForFirstRestart`. U nastavku je dat odgovarajući pseudokod.

```
void init() {
    inner = numConflictsForFirstRestart;
    outer = numConflictsForFirstRestart;
}

bool shouldRestart() {
    return numConflicts >= numConflictsForNextRestart;
}

void onRestart() {
    numConflicts = 0;

    if(inner >= outer)
    {
        outer *= restartInc;
        inner = numConflictsForFirstRestart;
    }
    else
    {
        inner = inner * restartInc;
    }

    numConflictsForNextRestart = inner;
}

void onConflict() {
    numConflicts++;
}
```

2.2.5 Politika izbora klauza za zaboravljanje

U trenutku kada treba vršiti zaboravljanje, potrebno je imati način izbora klauza koje će biti zaboravljene. Kod MiniSAT rešavača, klauze se sorti-

raju prema učestalosti njihove upotrebe u analizama konflikta. Zaboravlja se određeni procenat, `percentToForget`, ukupnog broja klauza koje smeju biti zaboravljene. Klauze koje ne smeju biti zaboravljene su one koje su uzrokovale dodelu vrednosti trenutno impliciranih promenljivih. Njih nazivamo *zaključanim* (eng. locked).

Aktivnost klauza se računa na isti način kao aktivnost promenljivih kod politike VS_MINISAT. Stoga u pseudokodu koji sledi računanje aktivnosti klauza nije prikazano.

```
void onForget(clauseArray learntClauses) {
    unlockedClauses=removeLockedClauses(learntClauses)
    sortAscendingOnActivity(unlockedClauses);

    firstToForget=ulockedClauses.length()*(1-percentToForget);

    for(i=firstToForget; i<unlockedClauses.length(); i++)
        delete(unlockedClauses[i]);
}
```

2.2.6 Politika izbora trenutka zaboravljanja

MiniSAT koristi politiku kod koje se zaboravljanje vrši pošto se nauči određeni broj klauza. Ovaj broj se povećava za faktor `forgetInc` posle svakog otpočinjanja iznova. U nastavku je dat odgovarajući pseudokod.

```
void init() {
    numClausesForNextForget = numInitialClauses / 3.0;
}

bool shouldForget() {
    numLearntClauses >= numClausesForNextForget;
}

void onConflict() {
        numClausesForNextForget *= forgetInc;
}

void onForget() {
        numLearntClauses = 0;
}
```

2.3 Grafovi i sličnost grafova

Grafovi su matematičke strukture koje odražavaju binarne relacije između nekih entiteta. Njihove primene su brojne i pojavljuju se u raznim naučnim disciplinama. Tipični primeri su hemija, biologija i računarstvo. Jedan od problema vezanih za grafove koji još uvek nisu zadovoljavajuće rešeni je ispitivanje sličnosti grafova. Samo definisanje sličnosti dva grafa je netrivijalan matematički zadatak. Jedan način merenja njihove sličnosti biće opisan u glavi 4.

Definicija 4 Graf $G = (V_G, E_G)$ *je uređeni par skupa čvorova* V_G *i skupa grana* $E_G \subseteq V_G \times V_G$. *Ovakav graf nazivamo* usmerenim *grafom. Kod* neusmerenog *grafa se podrazumeva da ako* $(u, v) \in V_G$, *onda i* $(v, u) \in V_G$.

Ukoliko je potrebno naglasiti da čvor i pripada grafu G, on se može označiti i kao i_G. Za granu (u, v) kažemo da je u njen *početni*, a v njen *krajnji čvor*. Kažemo da je grana (u, v) *ulazna grana* čvora v, a *izlazna grana* čvora u. Kod usmerenog grafa, za čvor v kažemo da je *naslednik* čvora u ako postoji grana (u, v), a *prethodnik* čvora w ako postoji grana (v, w). Za dva čvora $i, j \in V_G$ kažemo da su *susedni* ako važi $(i, j) \in E_G$ ili $(j, i) \in E_G$.

U daljem tekstu se koristi sledeći tekstualni zapis grafova: svi čvorovi se numerišu prirodnim brojevima, a za svaku granu se u jednom redu teksta zapisuju redni brojevi njenog početnog i krajnjeg čvora.

Definicija 5 *Matricu* $\mathcal{A}$, *takvu da je* $\mathcal{A}_{ij} = 1$ *ako je* $(j, i) \in E_A$, *a* $\mathcal{A}_{ij} = 0$ *u suprotnom, zvaćemo* matricom susednosti grafa A. *Matricom susednosti smatraće se i matrica za koju važi* $\mathcal{A}_{ij} \neq 0$ *ako važi* $(j, i) \in E_A$.

Definicija 6 *Za graf* $G' = (V', E')$ *kažemo da je* indukovani podgraf *grafa* $G = (V, E)$ *ako važi:*

- $V' \subseteq V$
- *ako važi* $u, v \in V'$ *i* $(u, v) \in E$, *onda važi i* $(u, v) \in E'$.

Definicija 7 *U neusmerenom grafu* $G = (V, E)$ klika *je takav podgraf* C *grafa* G *u kome su svi čvorovi međusobno povezani granama iz* G.

Definicija 8 *Za funkciju* $f : V_A \to V_B$ *kažemo da je* izomorfizam grafova A *i* B *ukoliko* $(i, j) \in E_A$ *implicira* $(f(i), f(j)) \in E_B$. *Izomorfizam grafa sa samim sobom nazivamo* automorfizmom grafa.

2.3.1 Sličnost grafova

Što se tiče sličnosti grafova, jedna od intuitivnijih definicija je Ulamova i neformalno je možemo opisati na sledeći način: dva grafa su utoliko sličnija ukoliko se moraju razbiti na manji broj podgrafova tako da su podgrafovi jednog grafa bijektivno upareni sa izomorfnim podgrafovima drugog grafa [10]. Očito, ova definicija je prihvatljiva za grafove koji imaju jednak broj čvorova, ali su moguće i modifikacije. Računanje broja podgrafova koji se mogu upariti zahteva rešavanje problema izomorfnosti podgrafa koji je NP-kompletan, tako da je njegov značaj više teorijski.

Drugi sistematičan pristup problemu sličnosti grafova je razmatranje sličnosti čvorova grafova umesto sličnosti celih grafova. Osnovni princip na kojem ove metode počivaju je da se čvorovi smatraju sličnim ako su im susedni čvorovi slični. Pošto je ova definicija rekurzivna, definiše se neka polazna sličnost koja se profinjuje iteracijama. Neke od primena ovog pristupa su heuristika za nalaženje podgrafa nekog grafa izomorfnog drugom grafu, prepoznavanje oblika i slično. Ovi pristupi su opisani u radovima [2] i [39].

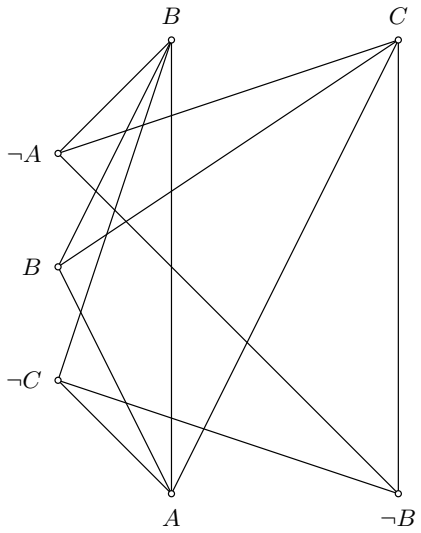
Postoje pristupi utvrđivanju sličnosti grafova koji se baziraju na nalaženju minimalnog zajedničkog nadgrafa, odnosno maksimalnog zajedničkog podgrafa [3], ali su ovi problemi, ponovo, NP-kompletni.

Osim nabrojanih, postoji i veći broj *ad hoc* pristupa koji se zasnivaju na određivanju minimalnih rastojanja između svih čvorova grafa i njihovom poređenju sa istom merom drugog grafa, raspodelama stepena čvorova i drugim sumarnim statistikama koje se mogu definisati nad grafovima.

2.3.2 Reprezentovanje formula u vidu grafova

U literaturi postoji više načina reprezentovanja iskaznih formula grafovima. U jednom od njih [37] polazi se od konjunktivne normalne forme formule. Svakoj pojavi literala u nekoj klauzi odgovara po jedan čvor grafa. Granama se povezuju čvorovi koji odgovaraju literalima iz različitih klauza, osim ako se radi o čvorovima koji odgovaraju negiranom i istom nenegiranom iskaznom slovu. Primer je dat na slici 2.2. Ova reprezentacija omogućava svođenje problema zadovoljivosti iskaznih formula na problem klika — problem provere da li za dati graf G i prirodan broj k, G sadrži kliku veličine k [37].

Drugi pristup koji je korišćen u ovom radu je da svaka disjunkcija iz konjunktivne normalne forme formule predstavlja čvor grafa [31]. Svako iskazno slovo koje se pojavljuje u formuli i njegova negacija takođe čine po jedan čvor grafa. Grane postoje od nenegiranog ka negiranom iskaznom slovu i od čvora koji predstavlja disjunkciju ka literalu (negiranom ili nenegiranom iskaznom slovu) koji se u njoj javlja. Ova reprezentacija se naziva *graf promenljivih i klauza*. Na slici 2.3 je dat primer. Ovaj pristup će biti nadalje korišćen u radu.

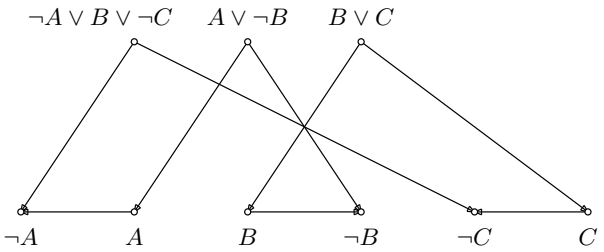


Slika 2.2: Graf koji reprezentuje formulu $(\neg A \vee B \vee \neg C) \wedge (A \vee \neg B) \wedge (B \vee C)$ u skladu sa prvim pristupom.

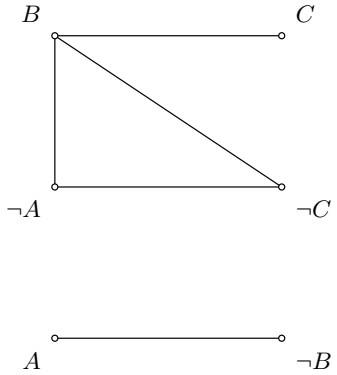
Kao i prethodni, treći način reprezentovanja formula grafovima pretpostavlja postojanje čvora za svako iskazno slovo ili njegovu negaciju [31]. Za klauze se ne dodaju čvorovi. Grana postoji između dva čvora koji odgovaraju literalima koji se zajedno javljaju bar u jednoj klauzi. Ova reprezentacija se naziva *graf promenljivih*. Primer je dat na slici 2.4.

2.4 Istraživanje podataka

Sa porastom količine podataka koji se čuvaju u okviru raznih informacionih sistema, javljaju se i novi problemi vezani za njihovo skladištenje i obradu. Skladištenje se obično vrši u velikim bazama podataka. Disciplina koja se bavi pronalaženjem skrivenog znanja u velikim količinama podataka, naziva se *istraživanje podataka* (eng. data mining). Iako su mogućnosti i efikasnost modernih baza podataka zadovoljavajući za mnoge probleme za koje su dizajnirane, analiza i istraživanje podataka zahteva drugačije pristupe od tradicionalnih. Razlozi za to su raznorodni. Pre svega, upiti ne moraju uvek biti jasno definisani. Nekada korisnik ne mora biti siguran ni šta želi da nađe. Podaci sa kojima se barata u analizi često mogu zahtevati različite vrste preprocesiranja zbog svoje nekompletnostni, nekonzistentnosti i slično. Takođe, izlaz ne mora biti podskup podatka iz baze ili sumarna statistika koju mogu da pruže ugrađene funkcije upitnih jezika, već model podataka, skup nekih zakonitosti koje važe među podacima, otkrivanje trendova koje



Slika 2.3: Graf promenljivih i klauza formule $(\neg A \vee B \vee \neg C) \wedge (A \vee \neg B) \wedge (B \vee C)$.



Slika 2.4: Graf promenljivih formule $(\neg A \vee B \vee \neg C) \wedge (A \vee \neg B) \wedge (B \vee C)$.

podaci sadrže i slično.

Osnovni zadaci istraživanja podataka su klasifikacija, regresija, analiza vremenskih serija, predviđanje, klasterovanje, pronalaženje pravila pridruživanja itd. Tehnike koje se koriste u ovim zadacima često pripadaju poljima mašinskog učenja i statistike. Primena ovih tehnika često zavisi od vrste i kvaliteta podataka. Stoga se podacima i njihovim karakteristikama posvećuje velika pažnja. Priroda podataka može biti različita — od jednostavnih numeričkih podataka, preko kategoričkih podataka sa izraženim relacijama između njih, do složenih prostornih i grafovskih podataka koji zahtevaju specifične tehnike obrade. Osim problema koji mogu biti posledica njihove prirode, podaci mogu biti nepotpuni, neprecizno mereni, mogu sadržati greške, njihovi određeni atributi mogu biti nerelevantni za problem koji se razmatra, njihova dimenzija može biti visoka i, naravno, količina podataka može biti ogromna. Zbog svega toga, algoritmi koji se primenjuju u ovoj oblasti moraju biti robusni i ne smeju imati visoku složenost u zavisnosti od količine ulaznih podataka i njihove dimenzije. Količina ulaznih podataka se često smanjuje i izborom uzorka, umesto da se razmatra ceo skup podataka koji su na raspolaganju.

Od posebnog je značaja problem klasifikacije koji je verovatno najčesće rešavani problem istraživanja podataka.

2.4.1 Problem klasifikacije

Klasifikacija je jedan od najčešćih problema u istraživanju podataka. Neka postoji konačan broj unapred određenih disjunktnih klasa $\mathcal{C}_i$ čija je unija ceo skup instanci. Problem se sastoji u tome da se za novu, nepoznatu instancu pronađe jedna klasa kojoj ova instanca pripada. Formalno, ako je X skup svih instanci i ako je dat skup $\{(x_1, c_1), (x_1, c_1), \ldots, (x_n, c_n)\}$ gde je $x_i \in X$, a $c_i \in \mathbf{N}$ predstavlja redni broj klase kojoj pripada i-ta instanca, onda je potrebno naći funkciju $h : X \to \mathbf{N}$ koja preslikava skup instanci X u skup rednih brojeva klasa kojima te instance pripadaju.

Problem klasifikacije se može javiti u različitim praktičnim situacijama: prepoznavanje relevantnih priloga na Internet grupama [22], prepoznavanje rukom pisanog teksta [23], prepoznavanje autorstva dokumenata [20] i slično.

Problem klasifikacije se može uopštiti tako što bi se dopustilo da svaka instanca pripada većem broju klasa. Ovo značajno otežava problem, uvodeći pitanja koja se u osnovnoj postavci problema ne javljaju.

Problem klasifikacije se često rešava tehnikama mašinskog učenja. Učenje se vrši na osnovu podataka koje nazivamo *podacima za trening*, a njihov skup *trening skupom*. Rezultat procesa učenja nazivamo *modelom*. Naučeni model se kasnije na neki način primenjuje na nepoznate podatke kako bi se došlo do zaključaka o njima. Pre upotrebe uobičajeno je da se model testira i evaluira. U tu svrhu se koristi odvojen skup *podataka za testiranje* — *test skup* na kojima se ocenjuje unapred izabrana mera kvaliteta kao što je procenat pravilno klasifikovanih instanci.

Pošto skupovi trening i test podataka obično nisu dati kao odvojeni, potrebno je iz ukupnog korpusa podataka izdvojiti test podatke. Pri tome se obično razmatra da li test podaci predstavljaju reprezentativan uzorak ukupnog korpusa, odnosno da li je raspodela karakteristika koje treba naučiti ista na ukupnom korpusu i na izdvojenim test podacima. U slučaju problema klasifikacije, bitna je raspodela pripadnosti instanci različitim klasama. Dodatno, razmatra se koja je dobra veličina test uzorka, koliko rezultati mogu varirati u zavisnosti od razlika u izboru ovog skupa i slično. Ovi problemi se prevazilaze korišćenjem tehnike k-slojnog unakrsnog ocenjivanja (eng. *cross validation estimation*). Radi se o pouzdanoj tehnici za ocenu mere kvaliteta. Najpre se ceo korpus podeli na k približno jednakih podskupova. Svaki *sloj* ocene se sastoji od treniranja modela na $k - 1$ podskupova i njegovog testiranja na preostalom podskupu. Ova procedura se vrši k puta, pri čemu se svaki put bira različit skup za testiranje. U svakom sloju se računa mera kvaliteta. Na kraju, ove vrednosti se sumiraju i dele brojem k. Dobijena vrednost se uzima za finalnu ocenu mere kvaliteta. Na ovaj način, svaka instanca iz korpusa je uključena u test skup tačno jed-

nom, tako da su svi raspoloživi podaci uključeni u proces ocenjivanja mere kvaliteta. K-slojno unakrsno ocenjivanje se naziva *stratifikovanim* ako u svakom sloju skupovi za trening i testiranje imaju raspodele karakteristika koje je potrebno naučiti približne onim iz celog korpusa podataka.

Ako se instance predstave preko vrednosti nekih svojih atributa kao elementi $\mathbf{R}^n$ i ako se odredi neka funkcija rastojanja, za rešavanje problema klasifikacije može se koristiti algoritam k *najbližih suseda* koji novu instancu jednostavno dodeljuje klasi koja se najčešće javlja među k najbližih instanci za trening u smislu usvojene distance. Jedno prirodno profinjenje metoda je uzimanje u obzir rastojanja suseda. Ako je V skup rednih brojeva klasa, za novu instancu x klasa može biti određena brojem $v \in V$ za koji se dobija maksimalna vrednost izraza:

$$\sum_{s \in NS(x,k)} w(s,x)\delta(v,h(s))$$

gde je $NS(x,k)$ skup k najbližih suseda instance x, $w(s,x)$ je težina uticaja koja se pridružuje susedu s, h je funkcija klasifikovanja, i $\delta(x,y)$ je 1 ako je $x = y$, a 0 inače. Najjednostavniji izbor za funkciju w je

$$w(s,x) = 1$$

Funkcija w se moze definisati i na sledeci nacin:

$$w(s,x) = \frac{1}{d(s,x)^2}$$

gde je d funkcija rastojanja. U ovom slučaju može se dopustiti i da svi podaci za trening daju neki doprinos, jer je on zanemarljiv zbog njihovog rastojanja. U slučaju da nisu svi atributi podjednako bitni i atributima se mogu dodeliti težine prilikom računanja rastojanja. Moguće su i druge modifikacije ovog metoda.

2.4.2 N-grami

Metode mašinskog učenja ili istraživanja podataka su često formulisane tako da se jednostavno primenjuju na numeričke podatke, ali teško na podatke u nekom drugom obliku. Stoga se traže načini da se i drugi podaci predstave u numeričkom obliku. To često podrazumeva i određeni gubitak informacije. U slučaju problema klasifikovanja tekstova, proteinskih sekvenci i sličnih podataka često se u svrhu predstavljanja podataka u numeričkom obliku koriste n-gramski profili [35].

Ako je data niska $S = s_1 s_2 \dots s_N$ nad azbukom Σ, gde je N pozitivan ceo broj, n-gram niske S, za $n \geq N$, je bilo koja podniska susednih simbolja dužine n. Na primer, za nisku `sad_ili_nikad`, 1-grami su: `s`, `a`, `d`, `_`, `i`, `l`, `i`, `_`, `n`, `i`, `k`, `a`, `d`. 2-grami su: `sa`, `ad`, `d_`, `_i`, `il`, `li`, `i_`, `_n`, `ni`, `ik`, `ka`, `ad`. 3-grami bi bili: `sad`, `ad_`, `d_i`, `_il`, `ili`, `li_`, `i_n`, `_ni`, `nik`, `ika`, `kad`, itd.

N-gramski profil niske je lista uređenh parova *(n-gram, frekvencija)* gde je frekvencija izračunata u odnosu na sve n-grame niske.

Osnovne prednosti korišćenja n-grama su robusnost (na primer, nisu mnogo osetljivi na greške u kucanju), nezavisnost od domena koji se analizira, efikasnost (dovoljan je jedan prolaz kroz tekst) i jednostavnost. Problem je eksponencijalna zavisnost broja mogućih n-grama u odnosu na dužinu n-grama.

2.5 Neke uzoračke statistike

U ovom radu biće korišćene i neke uzoračke statistike. Definicije date ovde se ne bave fundamentalnim statističkim pojmovima već samo korišćenim statistikama uzoraka. U daljem tekstu pod *uzorkom* će se podrazumevati niz realnih brojeva koji će se označavati sa $(X_1, X_2, \ldots, X_n)$.

Definicija 9 Prosek *ili* uzoračka sredina *uzorka* $(X_1, X_2, \ldots, X_n)$ *se definiše na sledeći način:*

$$\overline{X}_n = \frac{1}{n}\sum_{i=1}^{n} X_i$$

Definicija 10 *Neka je* $(X_1, X_2, \ldots, X_n)$ *dati uzorak. Njegov* n-ti percentil *je najmanja vrednost* X_i *koja je veća od* $n\%$ *elemenata uzorka. Pedeseti percentil se naziva* medijana *ili* središnja vrednost.

Definicija 11 Uzoračka disperzija *uzorka* $(X_1, X_2, \ldots, X_n)$ *se definiše sledećom formulom:*

$$\overline{S}_n^2 = \frac{1}{n}\sum_{i=1}^{n}(X_i - \overline{X})^2$$

Definicija 12 *Standardna devijacija uzorka* $(X_1, X_2, \ldots, X_n)$ *je kvadratni koren odgovarajuće uzoračke disperzije.*

Definicija 13 *Neka elementi uzorka* $(X_1, X_2, \ldots, X_n)$ *pripadaju konačnom skupu vrednosti* $\{c_1, c_2, \ldots, c_k\}$. *Neka je* p_i *udeo elemenata uzorka koji imaju vrednost* c_i. Entropija *ovog uzorka se definiše formulom:*

$$H_n = -\sum_{i=1}^{k} p_i \log_2 p_i$$

Entropija predstavlja meru neuređenosti vrednosti nekog skupa, odnosno meru nesigurnosti prilikom predviđanja vrednosti slučajno izabranog elementa nekog skupa. Vrednost entropije 0 znači zastupljenost samo jedne vrednosti u skupu. Maksimalna vrednost entropije znači ravnomernu zastupljenost svih mogućih vrednosti. U slučaju da skup vrednosti kome elementi uzorka pripadaju nije konačan, može se izvršiti njegova diskretizacija

— podela na konačan broj podskupova, a zatim se entropija računa tako što vrednosti p_i predstavljaju udeo pojavljivanja vrednosti iz i-tog podskupa.

Definicija 14 Kumulativna funkcija raspodele *uzorka* $(X_1, X_2, \ldots, X_n)$ *je:*

$$F(x) = \frac{1}{n} \sum_{i=1} I\{X_i \leq x\}$$

pri čemu je $I\{X \leq x\}$ indikatorska funkcija *koja ima vrednost 1 ukoliko je njen uslov zadovoljen, a 0 u suprotnom.*

Definicija 15 *Pirsonov koeficijent korelacije za dva uzorka* $(X_1, X_2, \ldots, X_n)$ *i* $(Y_1, Y_2, \ldots, Y_n)$ *se definiše na sledeći način:*

$$r_{XY} = \frac{n \sum_{i=1}^{n} X_i Y_i - \sum_{i=1}^{n} X_i \sum_{i=1}^{n} Y_i}{\sqrt{n \sum_{i=1}^{n} X_i^2 - (\sum_{i=1}^{n} X_i)^2} \sqrt{n \sum_{i=1}^{n} Y_i^2 - (\sum_{i=1}^{n} Y_i)^2}}$$

Koeficijent korelacije predstavlja meru linearne zavisnosti između dve promenljive čije vrednosti dati uzorci predstavljaju. Vrednost 1 označava postojanje pozitivne linearne zavisnosti, a vrednost -1 postojanje negativne linearne zavisnosti. Vrednost 0 označava nepostojanje linearne zavisnosti, a ostale vrednosti stepen u kome se može smatrati da ona postoji.

Glava 3

Pregled relevantnih radova i tehnika

U ovoj glavi će biti opisani neki rezultati koji su sa ovim radom bliski po tematici, odnosno koji se bave prilagođavanjem nekih sistema za dokazivanje teorema konkrentim instancama. Takvih radova još uvek nema mnogo. Radovi koji se bave širim tehnikama primenjenim u ovom radu neće biti detaljnije prikazivani, pošto su relevantne tehnike izložene u glavi 2.

3.1 Prilagođavanje dokazivača teorema instanci koja se dokazuje

Rad [25] se bavi rangiranjem strategija dokazivača teorema radi njegovog prilagođavanja konkretnoj instanci problema. Kod heurističkih dokazivača teorema do dokaza se dolazi primenom različitih strategija dokazivanja, pri čemu se dokaz obično konstruiše dostizanjem nekih podciljeva. Često su u jednom trenutku primenljive različite strategije i postavlja se pitanje koju primeniti. Dokazivači obično koriste neki unapred fiksiran redosled primene strategija, pri čemu se uvek koristi prva primenljiva strategija. Ovakav redosled je rezultat iskustva dizajnera dokazivača na određenom domenu. Druge strategije se mogu isprobati tek posle vraćanja unazad usled neuspele primene izabranih strategija. Neke strategije mogu voditi ciklusima, tako da to predstavlja dodatnu opasnost. Prilikom formiranja nekog fiksiranog redosleda strategija, kako bi dokazivač bio efikasniji, treba uzeti u obzir posledice u slučaju da je strategija neadekvatno primenjena, složenost provere primenljivosti strategije, kao i broj podciljeva koje strategija uvodi u proces dokazivanja. Strategije čija se primenljivost teže proverava, kao i one koje uvode veći broj novih podciljeva su obično niže rangirane jer zahtevaju veću količinu posla. Obično za svaki fiksiran redosled postoje instance za koje on nije pogodan.

U radu [25] dinamička promena redosleda strategija se vrši sortiranjem

strategija na osnovu numeričkog skora koji se računa za svaku strategiju. Faktori koji se pri tom uzimaju u obzir su polazni rang strategije u fiksiranom redosledu, veličina i složenost cilja na koji je potrebno primeniti strategiju i mera dobitka od primene strategije bazirana na stepenu zadovoljenosti preduslova strategije. Za svaku strategiju postoje preduslovi koji moraju biti zadovoljeni, ali i dodatni koji povećavaju verovatnoću da će strategija rezultirati dokazom. Finalni skor se računa tako što se saberu mere vezane za tri pomenuta faktora pomnožene nekim unapred određenim težinama. U cilju testiranja vršeno je poređenje između ovako poboljšanog i polaznog dokazivača CLAM-OYSTER na 8 teorema koje je trebalo dokazati. U 5 slučajeva je dinamičko rangiranje strategija dalo iste ili kraće dokaze, ali pored toga u 3 slučaja standardni dokazivač ne zaustavlja izvršavanje, što se u slučaju dinamičkog rangiranja strategija ne dešava.

Razlika u odnosu na sistem planiran u ovom radu je velika u domenu primene, a i u metodologiji. Sličnost je u tome što se ovaj sistem prilagođava instanci koju je potrebno dokazati, međutim dok ovaj pristup nema trening, već raspolaže ugrađenom funkcijom za rangiranje strategija, metodologija koja će biti predložena u ovom radu zavisi od trening korpusa koji se koristi, pa je i fleksibilnija. Takođe, pristup iz rada [25] je testiran na svega nekoliko teorema jedne teorije, što je neprihvatljivo za ovde planirani sistem. S druge strane, ovaj rad je jedan od prvih na temu prilagođavanja dokazivača problemu, tako da se ne može kritikovati u skladu sa današnjim merilima.

3.2 Prilagođavanje SAT rešavača korišćenjem mašinskog učenja

Jedan pristup ubrzavanju procedure DPLL pomoću mašinskog učenja opisano je u radu [21]. Iako je složenost procedure DPLL u najgorem slučaju eksponencijalna, vreme izvršavanja dosta zavisi od izbora promenljive na koju se primenjuje pravilo *split*. Rad [21] se bavi upravo učenjem izbora *split* promenljive. Primenjena tehnika učenja je učenje uslovljavanjem. Za njenu primenu potrebno je proces rešavanja modelirati kao Markovljev proces odlučivanja, tako što bi se identifikovala stanja procesa, kao i akcije koje je moguće preduzeti. Ovim pristupom se aproksimira funkcija vrednosti para stanje-akcija, odnosno vrednosti preduzimanja određene akcije u određenom stanju. Za to je potreban veći broj pokretanja procesa rešavanja. Osnovni problem kod učenja uslovljavanjem je identifikacija skupa stanja. U radu se naglašava da je isproban veći broj reprezentacija stanja koje su uključivale različite karakteristike instance koja se rešava, ali da je na kraju jedina karakteristika koja je zadržana kao relevantna broj promenljivih u instanci. Za moguće akcije uzeto je 7 već postojećih pravila izbora *split* promenljive. Opisuje se i niz tehničkih modifikacija standardnog metoda za učenje uslovljavanjem. Eksperimeni su rađeni odvojeno za 4 familije sa po 100 formula

za trening i 100 za testiranje. Kod 3 familije novi pristup je u istom rangu sa najboljim fiksiranim pravilima, dok je kod jedne postignuto značajno poboljšanje (mereno u broju posećenih čvorova pretrage, poboljšanje je trostruko).

Osnovni kvalitet učenja uslovljavanjem kao pristupa učenju je u prilagodljivosti u toku rada. To u pomenutom pristupu nije iskorišćeno. Pošto se rešavanje pokrene sa jednim pravilom izbora, ono se dalje ne menja. Prilagođavanje procedure za ispitivanje zadovoljivosti u toku rada je obećavajući pravac za buduća istraživanja.

Rad [21] je relevantan po tome što se radi o korišćenju mašinskog učenja u cilju ubrzavanja procedure DPLL koja je u osnovi svih SAT rešavača, ali se dosta razlikuje od ovde planiranog pristupa. Pre svega, učenje je vršeno za svaku familiju formula pojedinačno pretpostavljajući da je poznato kojoj familiji formula pripada. Ovakav problem je značajno lakši za rešavanje. Od metodologije koja će ovde biti predložena se očekuje da je u stanju da prepozna familiju kojoj instanca pripada. Osim toga, naučeno je koje pravilo izbora *split* promenljive treba koristiti za formule određene veličine iz određene familije. Za bolje rezultate potrebno je koristiti bogatiji skup svojstava formula i podešavati veći broj aspekata rada procedure za ispitivanje zadovoljivosti.

Rad [14] se bavi predviđanjem vremena rešavanja za neku instancu. Korišćena su dva stohastička rešavača. Instanca se predstavlja vektorom nekih svojih karakteristika [31]. Pomoću statističke regresije uči se funkcija koja preslikava takve vektore u medijane vremena izvršavanja pri određenom broju pokretanja rešavača na datoj instanci. Korišćeno je 6 familija formula, ali su testiranja vršena pre svega na instancama iz iste familije. Predviđanje vremena izvršavanja je prvo vršeno sa nekim fiksiranim parametrima. Koeficijent korelacije između stvarnih i predviđenih vremena je dostizao vrednost od 0.995. U daljim eksperimentima su pored karaketeristika formule i parametri smatrani za argumente funkcije koju treba naučiti. U slučaju jednog rešavača korišćena su 2 parametra, a u slučaju drugog 1. Koeficijent korelacije između stvarnih i predviđenih vremena rešavanja je dostizao 0.98. Poznavanje funkcije koja predviđa vremena rešavanja omogućava i biranje dobrih parametara za neku formulu. To se radi tako što se nalazi minimum ove funkcije po argumentima koji predstavljaju parametre rešavača. Pri tom su argumenti koji predstavljaju karakteristike formule fiksirani na vrednosti koje odgovaraju instanci koja se rešava. Koristeći ovakav pristup na izabranim test skipovima dobijena su ubrzanja od 2 puta do preko jednog reda veličine. Precizni rezultati vezani za ubrzanje rešavanja formula nisu dati.

Dok je predviđanje za formule iz iste ili vrlo slične familije odlično funkcionisala, pokušaj evaluacije naučene funkcije na raznovrsnom korpusu doveo je do, kako autori kažu, grešaka od nekoliko desetina redova veličine. Ovaj pristup je vrlo relevantan pošto se bavi podešavanjem vrednosti para-

metara prema instanci koja se rešava. Ipak, potrebno je konstruisati sistem koji može da funkcioniše sa velikim brojem familija, dok to nije slučaj sa ovim pristupom.

Za uspešno previđanje vremena rešavanja formule potrebno je raspolagati kvalitetnim skupom njenih svojstava koja bi je reprezentovala u toku učenja. Jedan ovakav skup je naveden u radu [31]. Jedan od osnovnih pokazatelja težine instanci sa fiksiranom dužinom klauze (k-SAT) je odnos između broja klauza i broja promenljivih. Za 3-SAT poznato je da je odnos za najteže instance oko 4.26. Pored ovog bitnog svojstva, u radu [31] se navodi još oko 90 svojstava koja mogu biti korelirana sa vremenom rešavanja formule. Ova svojstva čine osnovu za učenje predviđanja vremena rešavanja formule [14, 38] i koriste se u SATzilla rešavaču. Najvažnija svojstva su navedena u poglavlju 4.4.

Posle primene statističkih tehnika za identifikaciju značaja promenljivih izabrano je 30 od polaznih 91 svojstava. Finalni skup korišćen u predviđanju vremena rešavanja formula u radu [31] sastojao se od 368 svojstava koja uključuju polaznih 30 i neka od njihovih proizvoda. U kasnijem radu, isti autori su ipak koristili karakteristike nabrojane u poglavlju 4.4.

Najnovija dostignuća na polju prilagođavanja SAT rešavača instanci koja se rešava prikazana su u radu [38]. Opisan je sitem SATzilla koji uključuje nekoliko SAT rešavača i omogućava izbor jednog od njih koji najbolje odgovara instanci koju je potrebno rešiti. Mehanizam izbora rešavača se bazira na pristupu iz opisanog rada [14]. U toku treninga instance se rešavaju po nekoliko puta svakim od rešavača i na osnovu dobijenih podataka statističkom regresijom se uči funkcija koja predviđa vremena izvršavanja na sličnim problemima. Za svaki od rešavača uči se posebna funkcija, a za rešavanje nove instance bira se onaj koji ima najmanje predviđeno vreme. Jedno od predloženih poboljšanja je učenje odvojenih funkcija za zadovoljive i nezadovoljive instance uz korišćenje klasifikatora koji za svaku instancu predviđa njenu zadovoljivost pri tom računajući verovatnoću tačnosti tog predviđanja. Predviđanja vremena izvršavanja data funkcijama za zadovoljive i za nezadovoljive formule se kombinuju u odnosu koji odgovara izračunatim verovatnoćama.

Sistem je dizajniran tako da se u toku računanja karakteristika nepoznate instance paralelno pokreće i određen broj takozvanih predrešavača koji se prekidaju posle određenog vremena ukoliko ne reše instancu. Ukoliko je reše, prekida se dalje računanje karakteristika formule i proces predviđanja. Tako se omogućava da se lake instance brže reše. Postoji i rezervni rešavač koji se pokreće u slučaju da neke karakteristike instance budu predugo računate ili ako izabrani rešavač prijavi grešku. Za rezervni rešavač se obično bira onaj koji je u proseku najbolji među raspoloživim.

Za SAT takmičenje 2007. trenirana su 3 rešavača. Jedan za kategoriju `RANDOM`, drugi za `CRAFTED` i treći za sve instance. Trening korpusi ovih rešavača su sastavljeni od formula iz odgovarajućih kategorija sa svih

prethodnih takmičenja. Prva dva rešavača su se na takmičenju 2007. pokazali kao najbolji u svojim kategorijama.

Evaluacija na `RANDOM` kategoriji je pokazala značajno poboljšanje u brzini rešavanja formula. Prosečno vreme rešavanja za sistem SATzilla iznosi oko 90s, dok je u slučaju njegovog najboljeg komponentnog rešavača prosečno vreme oko 290s. Za `CRAFTED` kategoriju prosečno vreme sistema SATzilla je oko 150s, dok je za najbolji komponentni rešavač to vreme oko 280s. Za poslednju kategoriju — `INDUSTRIAL` ovo poboljšanje je manje, ali i dalje značajno — 260s u odnosu na 330s.

Iako se bave prilagođavanjem procesa rešavanja formule konkretnoj formuli koja se rešava, radovi [14, 31, 38] se ne bave analizom parametara SAT rešavača.

3.3 Optimizacija parametara SAT rešavača

Pristup opisan u radu [13] se odnosi na optimizaciju parametara SAT rešavača za određene grupe formula. Osnovni princip u optimizaciji parametara je slučajna lokalna pretraga pristrasna prema boljim lokalnim optimumima. Polazeći od neke kombinacije parametara koja se evaluira na nekom skupu, traži se lokalni optimum. Kada je nađen, parametri se slučajno mutiraju određeni broj puta i od novodobijene tačke se optimizacija nastavlja do novog lokalnog optimuma. Bolji od dva lokalna optimuma se prihvata kao polazna tačka za nove mutacije parametara i pretragu. Ovaj pristup predstavlja specijalan slučaj opšteg postupka za optimizaciju parametara opisanog u radu [12]. Na ovaj način se mogu efikasnije obraditi veći korpusi formula nego u slučaju sistematičnog rešavanja. S druge strane, postoji i opasnost nalaženja lokalnog optimuma daleko od globalnog (koja je u ovom pristupu donekle prevaziđena), kao i nedostatak kompletne raspodele vremena izvršavanja koja se dobija sistematičnim rešavanjem. Težina ovih zamerki zavisi od nameravane primene. Odsustvo sistematične pretrage je omogućilo eksperimentisanje sa 26 parametara. Prvi trening korpus sadrži 406 instanci vezanih za verifikaciju hardvera i softvera, uglavnom sa takmičenja. Od njih je 300 slučajno izabranih korišćeno za trening. Postignuto je smanjenje geometrijske sredine vremena rešavanja pojedinačnih formula od 21%. Ograničenje na vreme izvršavanja prilikom treninga je bilo 10 sekundi. Ograničenje u vreme testiranja nije navedeno. Drugi korpus se sastojao od dva problema iz verifikacije hardvera i softvera. Broji instanci u ovom korpusu nije precizno naveden. Vreme rešavanja bilo je ograničeno na 300 sekundi. Odvojeno su nađeni najbolji parametri za oba problema i konstatovano ubrazanje na test skupu od u proseku 4.5 puta za jedan problem i 500 puta za drugi. Parametri učeni na jednom domenu mogu davati bolje rezultate i na drugom, ali ne u meri u kojoj to važi na polaznom domenu. Sva pomenuta ubrzanja su računata u odnosu na podrazumevanu

konfiguraciju parametara za korišćeni rešavač, ali u slučaju drugog korpusa konstatovano je značajno ubrazanje i u odnosu na rešavač MiniSAT.

Osnovna pretpostavka ovog rada je da je familija formula za koju se radi poznata, dok se od planiranog sistema očekuje da je automatski prepoznaje. U slučaju testiranja na različitoj familiji takođe postoji ubrzanje, međutim postoji potreba za primenom na korpusu sa velikim brojem različitih familija. Glavni značaj ovog rada za planirani sistem je mogućnost upotrebe metodlogije koja je u njemu data u trening fazi tog sistema.

3.4 Automatsko unapređivanje SAT rešavača

Kvalitativno drugačiji pristup povećavanju efikasnosti SAT rešavača je prikazan u radu [1]. Ovaj rad se bavi evolucijom algoritama za *probleme zadovoljenja uslova* (eng. constraint satisfaction problems), sa posebnim akcentom na problemima kod kojih je broj uslova preveliki, tako da rešenje ne postoji. U takvim slučajevima je potrebno naći rešenje koje krši najmanji broj uslova. Kako se i SAT problem može razmatrati kao problem zadovoljenja uslova, pristup je demonstriran i na ovom domenu. Posebno su evoluirane politike izbora literala za različite skupove formula. Evoluirane politike su po performansama u rangu postojećih. Iako nema povećanja efikasnosti, ovo je značajan rezultat jer pokazuje da je automatska konstrukcija upotrebljivih politika moguća i da postoji mogućnost za automatsko unapređivanje SAT rešavača.

Glava 4

Klasifikovanje iskaznih formula

Opšte metode klasifikacije se ne mogu direktno primeniti na iskazne formule. One se obično primenjuju na podatke koji su na neki način numerički opisani. Stoga je potrebno da se formule predstave na ovakav način ili da se formuliše neka mera sličnosti nad samim formulama. Kako je prvi pristup računski manje zahtevan, što je bitno za korišćenje u SAT rešavačima, metode korišćene u ovom radu će se bazirati na njemu.

4.1 Opšti pristup klasifikovanju iskaznih formula

Klasifikovanje formula biće vršeno primenom funkcija rastojanja nad numeričkim opisima formula. Postoji više načina da se formuli pridruži neki numerički opis. Ovo je poznati problem *izbora atributa* (eng. feature selection) pomoću kojih se predstavljaju instance u mašinskom učenju i istraživanju podataka. Ovi numerički opisi biće nazivani *profilima instanci*. Kako bi se lakše primenile postojeće tehnike, bilo bi poželjno da skup izabranih atributa formule poseduje sledeća svojstva:

- Broj atributa treba da bude konačan i jednak za sve instance. Neke metode klasifikacije zahtevaju da se instance predstave kao elementi nekog vektorskog prostora. Metoda k najbližih suseda, zahteva pre svega strukturu metričkog prostora, ali se funkcije rastojanja značajno lakše zadaju u slučaju vektorskog prostora. Fiksiran broj atributa je stoga bitan uslov.

- Atributi treba da budu dobro definisani u smislu da jednoj instanci odgovara tačno jedan skup vrednosti atributa. Primer loše definisanih atributa nekog entiteta su elementi matrice susednosti grafa. Naime, isti graf se može zapisati pomoću više matrica susednosti menjanjem

numeracije čvorova, pa stoga jednom grafu može odgovarati više različitih instanci ako se koristi taj skup atributa.

Profili instanci mogu biti definisani na različite načine. U ovom radu će biti razmatrane tri vrste profila pomoću kojih će biti vršeno klasifikovanje:

- n-gramski profili za formule reprezentovane tekstom,
- profili frekvencija indukovanih podgrafova grafa pridruženog formuli i
- profili formula zasnovani na izabranom skupu sintaksnih svojstava.

Metoda kojom će biti vršeno klasifikovanje za sve tri vrste profila biće metoda k najbližih suseda. Za nju je potrebno formulisati konkretno rastojanje koje će biti korišćeno. U ovom radu biće korišćeno nekoliko funkcija rastojanja [35] za sve vrste profila. U zapisima svih funkcija rastojanja koje slede $f_1(x)$ i $f_2(x)$ su vrednosti atributa x u profilima $\mathcal{P}_1$ i $\mathcal{P}_2$.

$$d_1(\mathcal{P}_1, \mathcal{P}_2) = \sqrt{\sum_{x \in atributi} (f_1(x) - f_2(x))^2} \tag{4.1}$$

$$d_2(\mathcal{P}_1, \mathcal{P}_2) = \sum_{x \in atributi} \left(\frac{f_1(x) - f_2(x)}{\sqrt{|f_1(x) f_2(x)|} + 1} \right)^2 \tag{4.2}$$

$$d_3(\mathcal{P}_1, \mathcal{P}_2) \sum_{x \in atributi} \frac{|f_1(x) - f_2(x)|}{\sqrt{|f_1(x) f_2(x)|} + 1} \tag{4.3}$$

$$d_4(\mathcal{P}_1, \mathcal{P}_2) = \sum_{x \in atributi} \left(\frac{f_1(x) - f_2(x)}{\sqrt{|f_1(x) f_2(x)|} + 10} \right)^2 \tag{4.4}$$

U imeniocu kod distanci 4.2, 4.3 i 4.4 je geometrijska sredina vrednosti atributa, a aditivna konstanta se koristi u imeniocu da ne bi došlo do deljenja nulom.

4.2 N-gramski profili formula

Iskazne formule u konjunktivnoj normalnoj formi se jednostavno zapisuju u tekstualnom obliku. Formula se predstavlja nizom linija. Svaka linija predstavlja po jednu klauzu. Promenljive se označavaju celim brojevima osim nulom, pri čemu negativni brojevi označavaju negirane promenljive. Nula označava kraj linije. Na primer, formula

$$(A \vee \neg B \vee \neg C) \wedge (\neg A \vee B \vee \neg C) \wedge (\neg A \vee B \vee C)$$

se može zapisati na sledeći način:

```
1 -2 -3 0
-1 2 -3 0
-1 2 3 0
```

Ovaj način zapisivanja formula u konjunktivnoj normalnoj formi se zove DIMACS format.

N-gramiski profili su definisani u potpoglavlju 2.4.2. Broj različitih n-grama je konačan, ali veliki. Koristi se 13 karaktera — 10 za cifre i po jedan za negaciju, razmak i nov red. Stoga je broj n-grama dužine n jednak 13^n. Kako su puni profili previše veliki, biće uvedeno ograničenje na broj n-grama u profilu. Za zadato ograničenje l, biće izabrano l najfrekventnijih n-grama.

Frekvencije n-grama su jednoznačno definisane, ali očigledan problem je što se one mogu menjati pri preoznačavanju promenljivih ili zameni mesta promenljivih unutar klauza. Zbog ovoga se ne očekuju dobri rezultati prilikom korišćenja n-gramskih profila.

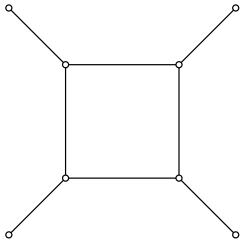
4.3 Profili formula zasnovani na frekvencijama podgrafova

Reprezentovanje formula u vidu grafova opisano je u poglavlju 2.3.2. Zahvaljujući toj reprezentaciji problem klasifikovanja iskaznih formula može se svesti na klasifikovanje grafova.

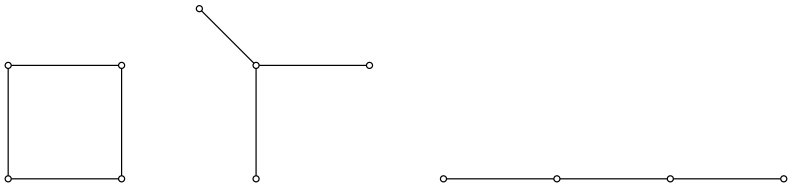
Jedan od skupova atributa grafa koji ispunjavaju zahteve date u poglavlju 4.1 je skup frekvencija indukovanih podgrafova polaznog grafa, sa fiksiranim brojem čvorova. Vrednosti ovih atributa biće zapisivani u obliku vektora. Za ove vektore biće korišćen i naziv *profili grafova*. Broj čvorova grafa će biti nazivan njegovom veličinom, a o indukovanim podgrafovima, će se, jednostavnosti radi, u nastavku govoriti samo kao o podgrafovima. Pod frekvencijom F_i jednog podgrafa se podrazumeva količnik broja njegovih pojavljivanja N_i i ukupnog broja pojavljivanja podgrafova iste veličine u polaznom grafu:

$$F_i = \frac{N_i}{\sum_j N_j}$$

Najjednostavniji način formiranja profila nekog grafa je sistematsko prebrojavanje svih indukovanih podgrafova zadate veličine. Da bi se izbrojali svi podgrafovi veličine n polazi se od neke grane grafa e i pronalaze se svi podgrafovi veličine n u kojima se ona javlja. Ovaj proces se ponavlja za svaku granu grafa. Posle toga se broj pojavljivanja svakog podgrafa deli brojem njegovih grana. Na kraju se izračunavaju količnici broja pojavljivanja svakog podgrafa i ukupnog broja pojavljivanja svih njegovih podgrafova. U slučaju grafa prikazanog na slici 4.1 postoje četiri neizomorfna indukovana



Slika 4.1: Primer grafa.



Slika 4.2: Podgrafovi veličine 4 grafa sa slike 4.1.

podgrafa veličine 4. Oni su prikazani na slici 4.2. Broj njihovih pojavljivanja je redom 1, 4 i 12, a frekvencije $\frac{1}{17}$, $\frac{4}{17}$ i $\frac{12}{17}$.

Ovaj postupak je vrlo neefikasan za grafove sa velikim brojem grana pošto zahteva obrađivanje svih podgrafova određene veličine.

Drugi pristup formiranju profila grafa zasniva se na semplovanju, odnosno na obradi slučajnog uzorka podgrafova [18]. Semplovanje podgrafa zadate veličine može biti opisano na sledeći način: izabrati slučajno granu grafa, a zatim iterativno proširivati listu čvorova koje izabrane grane spajaju, slučajno birajući susedne grane dok broj čvorova ne dostigne unapred zadatu vrednost (veličinu podgrafa). Pre svakog slučajnog biranja grane, priprema se lista susednih grana koje mogu povećati broj čvorova za jedan. Potom se grana bira iz te liste. Na kraju, nađeni podgraf se kompletira svim granama koje se u polaznom grafu javljaju između odabranih čvorova. Postupak je precizno opisan algoritmom datim na slici 4.3.

Algortam semplovanja se ponavlja unapred zadati broj puta N_S i pri svakom semplovanja se dobija jedan podgraf.

Kad se govori o frekvencijama pografova prirodno je izomorfne podgrafove smatrati jednakim. Zato kad se u nastavku bude govorilo o semplovanju nekog podgrafa, podrazumevaće se i slučajevi semplovanja njemu izomorfnih podgrafova.

Verovatnoća semplovanja različitih podgrafova iste veličine nije ista, jer u

Algoritam semplovanja podgrafa

Ulaz: V - skup čvorova ulaznog grafa.
E - skup grana ulaznog grafa.
n - tražena veličina podgrafa.
Izlaz: E_S je skup izabranih grana.
V_S je skup svih krajnjih čvorova grana iz E_S.

1. Slučajno izabrati granu $e_1 = (v_i, v_j)$ i inicijalizovati $E_S = \{e_1\}$ i $V_S = \{v_i, v_j\}$.
2. Napraviti listu L svih grana susednih granama iz E_S, a zatim iz L izostaviti sve grane između čvorova iz V_S.
3. Slučajno iz L izabrati granu $e = (v_k, v_l)$ i ažurirati $E_S = E_S \cup \{e\}$ i $V_S = V_S \cup \{v_k, v_l\}$.
4. Ponavljati korake 2 i 3 sve dok je $|V_S| < n$.
5. Dodati u E_S sve grane (v_k, v_l) za $v_k, v_l \in V_S$, koje već nisu u E_S.

Slika 4.3: Algoritam semplovanja podgrafa.

zavisnosti od njihove strukture, broj načina na koje se dva podgrafa jednog grafa mogu semplovati opisanim postupkom može biti različit. Kako bi se prilikom procene frekvencija podgrafova ovo uzelo u obzir, potrebno je izračunati verovatnoću P_i semplovanja podgrafa i (ili nekog njemu izomorfnog grafa). Svakom podgrafu i se pridružuje vrednost S_i čija je vrednost na početku 0 i koja se prilikom pronalaženja podgrafa i ažurira na osnovu pravila

$$S_i \leftarrow S_i + \frac{1}{P_i}$$

Frekvencija podgrafa i se tada aproksimira na osnovu formule

$$F_i = \frac{S_i}{\sum_j S_j}$$

Verovatnoća P_i se određuje na sledeći način. Semplovanje podgrafa veličine n se vrši slučajnim izborima $n-1$ grana. Nizove ovih grana nazivaćemo $(n-1)$-permutacijama. Verovatnoća semplovanja podgrafa je suma verovatnoća semplovanja ovih nizova od $n-1$ grana. Neka je $\{e_1, e_2, \ldots, e_m\}$ skup grana podgrafa, gde je $m \geq n-1$, S_m skup svih $n-1$ permutacija ovog skupa koje mogu dovesti do semplovanja razmatranog podgrafa i neka je e'_j slučajna promenljiva koja označava j-tu granu u $(n-1)$-permutaciji. Tada je

$$P_i = \sum_{\sigma \in S_m} \prod_{e'_j \in \sigma} P(e'_j = e_j | (e'_1, \ldots, e'_{j-1}) = (e_1, \ldots, e_{j-1}))$$

Računska složenost ovog pristupa se može oceniti kao $N_S \times O(n^{n+1})$ gde je n veličina podgrafa. Ona je očito nezavisna od veličine grafa u kojem se analiziraju podgrafovi.

Kad god se koristi semplovanje, postavlja se pitanje koliko uzoraka je dovoljno uzeti kako bi se dobili pouzdani rezultati. Preporučene vrednosti za različite veličine podgrafova, date su u tabeli 4.1 [19].

Veličina podgrafa	Broj uzoraka
3	5000
4	10000
5	50000
6	100000

Tabela 4.1: Broj preporučenih uzoraka za različite veličine podgrafa.

Broj podgrafova koji se javljaju u nekom grafu zavisi od samog grafa. Uprkos postojanju velikog broja mogućih podgrafova, može se desiti da se samo mali broj njih stvarno javlja u konkretnom slučaju, kao u grafovima koji predstavljaju iskazne formule. Primera radi, postoji 9364 neizomorfnih grafova veličine 5, ali se u profilima često pojavljuje svega dvadesetak ovakvih podgrafova. Ovo sugeriše da je pogodno koristiti retke zapise vektora što dovodi do uštede prostora. Još važnije, ovo omogućava efikasno ugnježdenje instanci u visokodimenzionalne vektorske prostore ukoliko upotrebljena metoda klasifikacije može adekvatno da iskoristi ovo svojstvo. Primera radi, kod metode k najbližih suseda, prilikom računanja euklidske distance

$$d(x, y) = \sqrt{\sum_i (x_i - y_i)^2}$$

sumiranje se može vršiti samo po frekvencijama prisutnih podgrafova pri čemu se podrazumeva da ako je podgraf i prisutan u grafu x, ali nije i u y, onda je $y_i = 0$ i obrnuto.

4.4 Profili formula zasnovani na izabranom skupu sintaksnih svojstava

Moguće je formulisati različita svojstva iskaznih formula u konjunktivnoj normalnoj formi. Neka od najjednostavnijih su broj klauza i broj promenljivih. Verovatno najpominjanija u literaturi je količnik ove dve veličine, pošto je

njegova vrednost korelirana sa težinom odgovarajućih formula (za ispitivanje zadovoljivosti). Pored pomenutih svojstava, u radu [31] se navodi još 88 potencijalno bitnih. U radu [38] se izdvaja podskup od 48 svojstava:

- Broj klauza c;
- Broj promenljivih v;
- Količnik $\frac{c}{v}$;
- Statistike stepena čvorova koji odgovaraju promenljivim u grafu promenljivih i klauza: prosek, disperzija, minimum, maksimum i entropija;
- Statistike stepena čvorova koji odgovaraju klauzama u grafu promenljivih i klauza: prosek, disperzija, minimum, maksimum i entropija;
- Statistike stepena čvorova u grafu promenljivih: prosek, disperzija, minimum, maksimum i entropija;
- Odnos pozitivnih i negativnih literala u svakoj klauzi: prosek, disperzija i entropija;
- Odnos pozitivnih i negativnih pojavljivanja svake promenljive: prosek, disperzija, minimum, maksimum i entropija;
- Udeo binarnih klauza;
- Udeo ternarnih klauza;
- Udeo Hornovih klauza;
- Broj pojavljivanja u Hornovim klauzama za svaku od promenljivih: prosek, disperzija, minimum, maksimum i entropija;
- Broj primena pravila *unit propagation* procedure DPLL pri dubinama pretrage od 1, 4, 16, 64 i 256;
- Ocena veličine prostora pretrage: prosečna dubina do konflikta i ocena logaritma broja čvorova;
- Broj koraka do najboljeg lokalnog minimuma pri lokalnoj pretrazi stohastičkim SAT rešavačem SAPS pri većem broju pokretanja: prosek, medijana, 10-ti i 90-ti percentil;
- Prosek po svim pokretanjima proseka poboljšanja po koraku do najboljeg rešenja koristeći SAPS;

- Prosek količnika poboljšanja do prvog lokalnog minimuma i ukupnog poboljšanja pri većem broju pokretanja korišćenjem stohastičkih SAT rešavača SAPS i GSAT;
- Prosek standardnih devijacija broja nezadovoljenih klauza u svakom lokalnom minimumu pri većem broju pokretanja rešavača SAPS.

U ovom radu biće korišćena prva 33 navedena svojstva koja su sva sintaksne prirode — zaključno sa statistikama vezanim za broj pojavljivanja promenljivih u Hornovim klauzama. Ostala svojstva su odbačena jer se sporije računaju. Profili formula predstavljaju vektore ovih karakterisitka.

Glava 5

Opis metodologije

U ovoj glavi je opisana metodologija izbora vrednosti parametara SAT rešavača na osnovu formule koja se rešava. U njoj će biti definisani svi aspekti metodologije koji su navedeni u glavi 1. Biće definisan skup parametara koji će biti podešavani, kao i skup dopustivih vrednosti za te parametre. Biće opisan korpus na kome će metodologija biti trenirana i testirana, kao i mere kvaliteta i principi evaluacije metodologije.

5.1 Pregled metodologije

Cilj ovog pristupa je da se za nepoznatu formulu izaberu pogodne vrednosti parametara sa kojima bi se SAT rešavač primenio na nju. Osnovni pristup se sastoji u prepoznavanju problema (familije iskaznih formula) čija je data formula instanca i primeni najboljih vrednosti parametara za taj problem. Najbolje vrednosti se nalaze iscrpnom pretragom prostora vrednosti parametara. Sam problem je predstavljen skupom formula za koje je poznato da su instance tog problema. Varijanta ovog pristupa je da se umesto prepoznavanja problema čiju instancu nepoznata formula predstavlja, odrede formule koje su joj u nekom smislu najsličnije i da se vrednosti parametara izaberu među vrednostima parametara koje su najbolje za te formule. Razlike između ova dva pristupa su u tome što su u prvom unapred određene familije formula kojima se nepoznata formula pridružuje, dok u drugom nisu, i što je u drugom potrebno precizirati način izbora vrednosti parametara.

Kao što je rečeno u glavi 1, metodologija se sastoji iz dve osnovne faze. U prvoj se vrši sistematično rešavanje nekog reprezentativnog korpusa iskaznih formula pri čemu se beleže njihova vremena rešavanja. One bivaju rešene za sve kombinacije vrednosti parametara. Kako je broj mogućih parametara i njihovih vrednosti veliki, moraju se napraviti određeni izbori kako bi računanje bilo izvodljivo. Stoga je na početku potrebno napraviti izbor parametara koji će biti razmatrani, kao i izbor njihovih dopustivih vrednosti. Zatim je potrebno izabrati korpus iskaznih formula koje će biti rešavane.

Faza treninga

Izlaz: D_{prof} - skup profila formula iz trening korpusa.
K - skup uređenih parova (Familija,Opt) gde je Opt skup najboljih vrednosti parametara za datu familiju.
$T \subseteq D \times \prod V_{p_i} \times \mathbf{R}$- tabela vremena rešavanja formula iz trening korpusa za sve dopustive vrednosti razmatranih parametara.

1. Izbor skupa razmatranih parametara P.
2. Izbor skupova dopusitivih vrednosti parametara V_{p_i}.
3. Izbor trening korpusa D.
4. Rešavanje svih formula za sve kombinacije vrednosti parametara, pri čemu se formira tabela T sortirana prema vremenima rešavanja u rastućem poretku.
5. Određivanje u proseku najboljih vrednosti parametara za familije formula i formiranje izlaznog skupa K.

Slika 5.1: Faza treninga.

Njegova reprezentativnost bi se oslikavala kroz veliki broj različitih familija formula koje bi bile zastupljene i dovoljan broj iskaznih formula u tim familijama. Takođe, zastupljene familije treba da pokrivaju i industrijske i veštački konstruisane probleme. Kad se definiše mera efikasnosti, određuju se najbolji parametri kako za pojedinačne formule, tako i za cele familije. Pod najboljim vrednostima parametara za neku familiju podrazumevaće se vrednosti za koje je rešen najveći broj formula iz te familije u nekom unapred zadatom vremenu. U slučaju da dva skupa vrednosti parametara rešavaju isti broj formula, boljim će se smatrati onaj koji ima manje ukupno vreme rešavanja. Ovu faza se može nazivati *treningom* sistema. Faza treninga je sumirana je na slici 5.1

Druga faza metodologije se sastoji u prilagođavanju vrednosti parametara SAT rešavača konkretnoj formuli koja se rešava. Ova faza se može nazivati fazom eksploatacije. Razmatraju se dve moguće verzije. U prvoj se vrši klasifikacija nepoznate formule u neku od unapred zadatih familija, a zatim se primenjuju najbolje vrednosti parametara za tu familiju. Ova verzija je sumirana na slici 5.2. Druga verzija je fleksibilnija. Bliske formule ne moraju biti grupisane u unapred fiksirane familije. Moguće je prvo naći u nekom smislu najsličnijih n formula, a onda se za svaku od njih sve kombinacije vrednosti parametara pomoću kojih se postiže isto vreme rešavanja grupišu u odvojene skupove. Za svaku od formula je moguće izabrati po

Faza eksploatacije - prva verzija

Ulaz: Nepoznata formula F,
Skup profila formula iz trening korpusa D_{prof},
Skup parova (Familija, Opt) gde je Opt skup najboljih vrednosti parametara za datu familiju.
Izlaz: SAT ukoliko je formula F zadovoljiva,
$UNSAT$ ukoliko je formula F nezadovoljiva.
Parametar: k - parametar metode k najbližih suseda.

1. Izračunati profil formule F.
2. Izvršiti klasifikaciju formule F na osnovu njenog profila u jednu od poznatih familija metodom k najbližih suseda.
3. Rešti formulu F sa vrednostima parametara koje su u fazi treninga određene kao najbolje za tu familiju.

Slika 5.2: Faza eksploatacije - prva verzija.

m skupova kojima odgovaraju najmanja vremena rešavanja. Potom se bira kombinacija vrednosti parametara koja se najčešće javlja u svim ovako izabranim skupovima za sve formule. Zbog grupisanja vrednosti parametara sa istim vremenom rešavanja broj vrednosti parametara koji ulaze u razmatranje za svaku od bliskih formula može biti veći od m. Ova verzija je opisana na slici 5.3.

5.2 Izbor SAT rešavača

Za SAT rešavač sa kojim će biti eksperimentisano izabran je ArgoSAT[1]. On je izabran jer podržava veliki broj različitih politika za razne aspekte funkcionisanja SAT rešavača i stoga omogućava raznovrisniji izbor parametara i njihovih dopustivih vrednosti.

5.3 Skup parametara čiji se uticaj na rešavanje formule razmatra

Za eksperimente su izabrana tri parametra. Taj broj bi mogao biti dovoljan da pokaže isplativost korišćenja predložene metodologije, uz prihvatljivu računsku zahtevnost. Ona naravno zavisi i od skupa njihovih dopustivih vrednosti. Izabrani parametri su *politika izbora promenljive*, *politika izbora polariteta promenljive* i *politika otpočinjanja iznova*. Ovi parametri i neke

[1] http://argo.marf.bg.ac.yu/software/ArgoSat-v2.0/argosat.zip

Faza eksploatacije - druga verzija

Ulaz: Nepoznata formula F,
Skup profila formula iz trening korpusa D_{prof},
$T \subseteq D \times \prod V_{p_i} \times \mathbf{R}$- tabela vremena rešavanja formula iz trening korpusa za sve dopustive vrednosti razmatranih parametara.
Izlaz: SAT ukoliko je formula F zadovoljiva,
$UNSAT$ ukoliko je formula F nezadovoljiva.
Parametri: m - broj najboljih vrednosti parametara za formulu koji se uzimaju u obzir.
n - broj bliskih formula koje se uzimaju u obzir.
Koristi se: S - multiskup vrednosti parametara.

1. Izračunati profil formule F.
2. Za nepoznatu formulu odrediti n najsličnijih formula iz trening korpusa.
3. Na osnovu tabele T, za svaku od odabranih formula odrediti najboljih m skupova kombinacija vrednosti parametara sa istim vremenom rešavanja formule. Svaku od kombinacija iz ovih skupova dodati u S.
4. Izabrati kombinaciju vrednosti parametara koja se najčešće javlja u S i rešiti nepoznatu formulu sa tako podešenim vrednostima parametara.

Slika 5.3: Faza eksploatacije - druga verzija.

Dopustive vrednosti	Parametri politike
VS_RANDOM	
VS_RANDOM p VS_MINISAT_INIT	p=0.05 `bumpAmount`=1.0 `decayFactor`=1.0/0.95
VS_MINISAT_INIT	`bumpAmount`=1.0 `decayFactor`=1.0/0.95

Tabela 5.1: Dopustive vrednosti za parametar *politika izbora promenljive*.

Dopustive vrednosti	Parametri politike
PS_TRUE	
PS_FALSE	
PS_RANDOM p	p=0.5
PS_POLARITY_SAVING	
PS_POLARITY_SAVING_INIT	

Tabela 5.2: Dopustive vrednosti za parametar *politika izbora polariteta promenljive*.

njihove dopustive vrednosti opisane su u podpoglavljima 2.2.2, 2.2.3, 2.2.4. Izabrani parametri su posebno pogodni po tome što dozvoljavaju eksperimentisanje sa politikama umesto sa parametrima nekih politika, pa se može očekivati da će razlike između njih biti značajnije.

5.4 Dopustive vrednosti izabranih parametara

Izabrane dopustive vrednosti su date u tabelama 5.1, 5.2 i 5.3. Ukupan broj kombinacija vrednosti parametara za koje treba rešiti svaku formulu je 60. Razmatrane dopustive vrednosti su politike koje se najčešće koriste. Za vrednosti parametara izabranih politika su uzete one koje se koriste u literaturi ili su podrazumevane u rešavačima iz kojih potiču.

Dopustive vrednosti	Parametri politike
RS_NO_RESTART	
RS_MINISAT	`numConflictsForFirstRestart`=100 `restartInc`=1.5
RS_LUBY	`sizeFactor`=512
RS_ARMIN	`numConflictsForFirstRestart`=100 `restartInc`=1.5

Tabela 5.3: Dopustive vrednosti za parametar *politika otpočinjanja iznova*.

5.5 Izbor korpusa formula za treniranje i evaluacija

Prilikom formulisanja detalja eksperimenata jedan od osnovnih izbora u fazi treninga je izbor reprezentativnog korpusa na kome će metodologija biti sprovedena i evaluirana. Više ovakvih korpusa je sastavljeno radi takmičenja SAT rešavača *SAT competition* koje je bilo organizovano više puta od 2002. godine. Za trening i evaluaciju izabran je korpus iz 2002. godine pošto se sastoji od velikog broja različitih familija iskaznih formula. Pod familijama se podrazumevaju skupovi formula koje predstavljaju instance istog praktičnog ili teorijskog problema. Većina familija ovog korpusa sadrži kako lake tako i teške formule, često sa postepenim prelazom od najlakših ka najtežim. Postoji četrdesetak familija formula. Ukupan broj formula je 1964. Broj formula koje su bile uključene u razmatranje pri različitim pristupima klasifikacije je bio različit pošto se pomoću nekih pristupa ne mogu obraditi sve formule. Formule ovog korpusa su sistematski rešavane za sve dopustive vrednosti izabranih parametara i stoga je nad njime bila moguća detaljna evaluacija. O ovome će biti više reči u nastavku.

Pored korupsa iz 2002. godine, biće korišćen i korpus iz 2007. Ovo je bitno kako bi se ocenila prenosivost rezultata primene metodologije na korpus sa različitom raspodelom formula. Ovaj korpus se značajno razlikuje od korpusa iz 2002. Postoji svega 12 zajedničkih formula, a skup zatupljenih familija je u značajnoj meri različit. Zbog toga se ovaj korpus može smatrati ozbiljnim izazovom za predloženi pristup. Korpus sadrži tridesetak familija i ukupno 906 instanci. Pored poređenja predloženog pristupa sa polaznim ArgoSAT rešavačem, na ovom korpusu biće urađeno i poređenje sa sistemom SATzilla. Ovo poređenje nije od suštinskog značaja pošto se u ovom radu evaluira poboljšanje performansi SAT rešavača postignuto primenom predložene metodologije u odnosu na polazni rešavač. Međutim, ipak je zanimljivo razmotriti rezultate poređenja.

5.6 Karakteristike iskaznih formula na osnovu kojih se vrši izbor vrednosti parametara

Jedna od stavki bitnih za kompletno formulisanje metodologije je identifikacija karakteristika iskaznih formula na osnovu kojih bi se vršio izbor pogodnih vrednosti parametara. Skupovi ovih karakteristika su već opisani u glavi 4. Izdvojeni skupovi karakteristika su frekvencije n-grama u tekstualnom zapisu formule (poglavlje 4.2), frekvencije podgrafova grafa klauza i promenljivih koji odgovara formuli (poglavlje 4.3) i izabrani skup sintaksnih svojstava formule (poglavlje 4.4). Funkcionisanje metodologije će biti ispitano za svaki od ovih skupova karakteristika.

5.7 Mere kvaliteta i principi evaluacije

Od mera kvaliteta biće formulisane mere za evaluaciju postupka klasifikovanja, kao i mere kvaliteta finalnih rezultata predložene metodologije.

5.7.1 Evaluacija klasifikovanja

Za evaluaciju kvaliteta kalsifikacije korišćene su dve mere. Prva je *preciznost klasifikacije* — broj pravilno klasifikovanih instanci podeljen ukupnim brojem instanci koje se klasifikuju. Pod *preciznošću za pojedinačnu familiju* podrazumevaćemo broj pravilno klasifikovanih formula iz te familije podeljen ukupnim brojem formula iz te familije. Kako broj formula po familiji ne mora biti isti, postojanje jedne ili više neproporcionalno brojnih familija sa visokom preciznošću po familiji može sakriti nisku preciznost na malim familijama ukoliko se koristi samo preciznost klasifikacije kao mera kvaliteta. Stoga se kao još jedna mera uvodi *prosek preciznosti po pojedinačnim familijama*. Ova mera je nezavisna od raspodele instanci po familijama i daje bolju sliku o kvalitetu klasifikatora ukoliko može doći do promene te raspodele.

5.7.2 Evaluacija finalnih rezultata predložene metodologije

Razni SAT rešavači koriste različite podrazumevane vrednosti parametara. Zbog toga se postavlja pitanje koje vrednosti parametara treba uzeti za referentne pri evaluaciji rezultata predložene metodologije. Kako ne postoji nikakvo čvrsto opravdanje za korišćenje bilo koje fiksirane kombinacije vrednosti parametara, za referentnu će biti uzeta kombinacija vrednosti parametara sa najboljim performansama. Rezultati dobijeni primenom predložene metodologije bi trebalo da budu bolji u smislu neke mere kvaliteta. Kako bi se mogla proceniti veličina dobijenog poboljšanja, prilikom evaluacije će biti prezentovane i vrednosti mera kvaliteta u slučaju kad bi za svaku formulu bile izabrane vrednosti parametara koje su najbolje za nju.

Izabrano je nekoliko mera kvaliteta. Cilj rešavača je da u što kraćem vremenu reši što više instanci. Razlike u vremenu rešavanja instanci su svakako značajne, ali razlika između uspešnog rešavanja i neuspeha je presudna. Stoga, za glavnu meru kvaliteta uzima se broj rešenih iskaznih formula. Ukoliko dva rešavača reše isti broj formula, onda se boljim smatra onaj koji je pri tome utrošio manje vremena. Ovaj način poređenja SAT rešavača koristi se i na takmičenjima SAT Competition.

Najjednostavniji načini merenja vremena su *ukupno vreme* utrošeno za rešavanje formula iz korpusa i *prosečno vreme* rešavanja formule. Pri računanju ukupnog vremena za formule koje nisu rešene podrazumeva se vreme rešavanja jednako vremenskom ograničenju. Prosečno vreme se dobija kada se ukupno vreme podeli brojem rešavanih formula.

Osnovni problem vezan za korišćenje vremena rešavanja kao mere kvaliteta je to što vreme rešavanja mora biti ograničeno. Neke formule neće biti rešene u zahtevanom vremenu. Za takve formule se ne može znati koliko bi vremena zahtevale za rešavanje, pa ih stoga ne možemo uzeti u obzir pri evaluaciji. To najčešće dovodi do toga da rešavači koji reše manje formula imaju bolja ukupna vremena. Što je još zanimljivije, formule koje su bolji rešavači u stanju da reše, a lošiji nisu, su i za bolje rešavače obično na granici isteka vremena, pa stoga prilikom računanja prosečnog vremena dovode do obrnutog rangiranja rešavača. Stoga, ukupno i prosečno vreme rešavanja nisu dobre mere kvaliteta. Zbog toga se kao mera uvodi *središnje vreme*. Ono predstavlja medijanu vremena rešavanja formula iz izabranog korpusa. Ukoliko je više od pola formula nerešeno i za njihovo vreme rešavanja se uzme vreme jednako ograničenju, medijana mora biti jednaka tom ograničenju i stoga središnje vreme nije tačno izračunato. Ukoliko je bar pola formula rešeno, središnje vreme će biti jednako vremenu rešavanja neke rešene formule i stoga će biti tačno izračunato bez obzira što potencijalno veliki broj formula nije rešen. Zbog toga ono predstavlja znatno pouzdaniju meru kvaliteta od ukupnog ili prosečnog vremena.

Pored ovih osnovnih mera, zanimljivo je analizirati i druge statistike procesa rešavanja. Za vremena rešavanja formula pomoću predložene metodologije biće kreiran histogram procenata u odnosu na vremena rešavanja za najbolje fiksirane vrednosti parametara za iste formule. Ostale statistike će biti računate za tri pristupa izboru vrednosti parametara — jedan koji odgovara predloženoj metodologiji i dva referentna — jedan kod kojeg se za svaku formulu biraju vrednosti parametara najbolje za nju i drugi kod kojeg se za svaku formulu biraju najbolje fiksirane vrednosti parametara. Biće izračunata kumulativna funkcija raspodele vremena rešavanja formula — funkcija koja je u svakoj tački x jednaka broju formula rešenih za x ili manje sekundi. Biće prikazana i funkcija čiji je argument indeks u sortiranom nizu vremena rešavanja pri određenom načinu izbora vrednosti parametara, a vrednost je upravo odgovarajuće vreme rešavanja. Ovakva funkcija će pokazati koliko je teško povećati broj rešenih formula. Još jedan skup statistika koje će biti prikazane se odnosi na kvalitet izabranih vrednosti parametara. Za svaki od 3 pominjana pristupa izboru vrednosti parametara, za svaku formulu, biće određen indeks izabranih vrednosti parametara u nizu vrednosti parametara koji je sortiran prema vremenu rešavanja formule sa tim vrednostima parametara. Najbolji indeks (indeks 1) odgovara vrednostima parametara sa najkraćim vremenom rešavanja. Pri tome se smatra da različitim vrednostima parametara sa istim vremenom rešavanja odgovara jedan indeks. Prosek ovakvih indeksa pokazuje koliko su birani indeksi pri nekom pravilu izbora bliski najboljim vrednostima parametara. Osim proseka, biće dat i histogram broja formula za koje su izabrane vrednosti parametara sa određenim indeksom, kao i kumulativna funkcija koja pokazuje za koliko formula su izabrane vrednosti parametara sa indeksom

manjim ili jednakim datom. Za svaki pristup biće prikazan i procenat formula za koje je su izabrane najbolje vrednosti parametara.

Glava 6

Evaluacija metodologije

U ovoj glavi biće opisani detalji izvršenih eksperimenata. Osnovni principi i mere pomoću kojih će predložena metodologija biti evaluirana su opisani u glavi 5. Dati su eksperimentalni rezultati koji ilustruju primenljivost predložene metodologije, kao i opravdanost osnovne teze ovog rada — da se inteligentnim biranjem vrednosti parametara SAT rešavača, zasnovanim na analizi sintakse formule koja se rešava, može značajno povećati njegova efikasnost. Opisani su i eksperimenti pomoću kojih se ispituje osnovanost drugih hipoteza datih u glavi 1. Na kraju je opisana implementacija sistema ArgoSmArT koji realizuje metodologiju predloženu u glavi 5 i izvršena je njegova evaluacija na nezavisnom korpusu.

6.1 Rešavanje formula iz korpusa

U poglavlju 5.4 su navedeni izabrani parametri i njihove dopustive vrednosti. Kako bi se izbegla pristrasnost prema određenim familijama, prilikom rešavanja je korišćen ceo korpus. Kasnije su izbacivane neke formule u pristupima klasifikaciji koji su računski previše zahtevni. Ovih formula je bilo najviše 21 i to samo u jednom pristupu (klasifikovanje bazirano na frekvencijama podgrafova), a nalazile su se u više familija, tako da njihovo izbacivanje ne može predstavljati značajan izvor pristrasnosti. U drugim pristupima izbačene su svega po dve. Ovi podaci su eksplicitno naglašeni u eksperimentalnim rezultatima. Vreme rešavanja jedne formule je ograničeno na 10 minuta. Sve formule su rešavane za svih 60 kombinacija parametara. Kako vreme rešavanja u praksi može zavisiti i od zapisa formule, svaka formula je transformisana preimenovanjem promenljivih i permutovanjem klauza. Kao vreme rešavanja formule uzeta je aritmetička sredina rešavanja permutovane i nepermutovane varijante. U slučaju da bar jedna od varijanti nije rešena u zadatom vremenskom roku, smatra se da formula nije rešena. Svaka formula i njena permutovana verzija su rešavane po 60 puta (za svaku kombinaciju parametara). Ukupan broj poziva rešavača je bio 235680. Za ovu količinu

posla, bili su potrebni veliki računarski resursi. Korišćen je klaster računar IBM Cluster 1350 Matematičkog instituta SANU sa 32 procesora. Ukupno vreme trajanja rešavanja je bilo oko mesec dana.

6.2 Evaluacija klasifikovanja formula

Glavni zadatak ovog poglavlja je testiranje prve hipoteze ovog rada navedene u glavi 1. Ona pretpostavlja da između formula iste familije postoji sintaksna sličnost koja se može automatski prepoznati i da se familija kojoj formula pripada može prepoznati na osnovu njene sintakse i znanja o drugim formulama. U ovom poglavlju su opisani eksperimenti vezani za sintaksnu sličnost formula koje pripadaju istoj familiji, kao i eksperimenti vezani za klasifikovanje iskaznih formula.

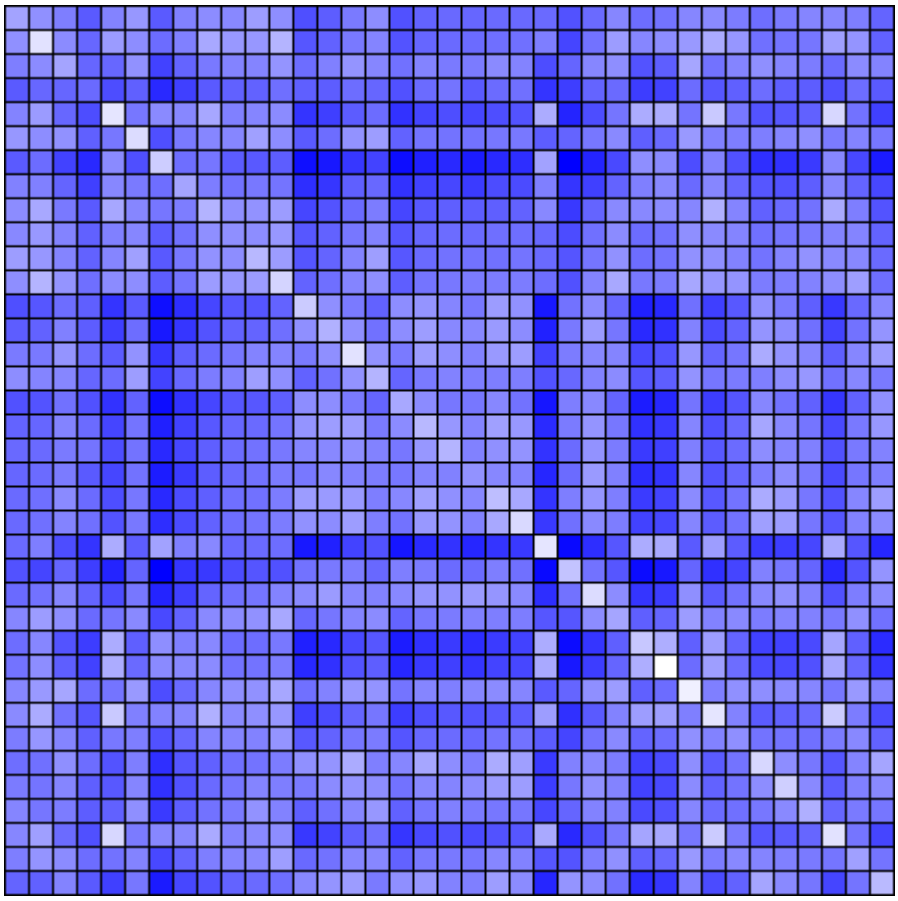
6.2.1 Eksperimenti vezani za prosečne distance između klasa

Prvi deo hipoteze koji pretpostavlja da između formula iste familije postoji sintaksna sličnost koja se može automatski prepoznati, može se evaluirati računanjem prosečnih distanci između formula iz različitih familija. One bi se za dve izabrane familije A i B računale kao prosek distanci između svake dve formule $a \in A$ i $b \in B$. Ovi proseci bi trebalo da budu najmanji prilikom poređenja familije sa samom sobom.

Slika 6.1 prikazuje prosečne distance između formula iz različitih familija. Svetlije nijanse označavaju manje vrednosti, a tamnije veće. Na slici i-ta kolona i i-ta vrsta odgovaraju istoj familiji. Prosek rastojanja između formula iz familije A i formulama iz familije B prikazan je u preseku vrste i kolone koje odgovaraju tim familijama. Na slici su prikazane vrednosti za funkciju rastojanja d_3 definisanu u poglavlju 4.1 i za profile dobijene na osnovu izabranog skupa sintaksnih svojstava (poglavlje 4.4). Ova funkcija rastojanja i ova vrsta profila su izabrani jer daju najbolje rezultate što se vidi iz podpoglavlja koje sledi. Zbog preglednosti su umesto stvarnih vrednosti predstavljeni njihovi logaritmi čije su vrednosti linearnom transformacijom preslikane u interval [0, 255].

Primećuje se da su kvadrati na dijagonali svetliji što znači da im odgovaraju manje vrednosti proseka, odnosno da je sintaksna sličnost između formula iste familije izraženija nego sličnost između različitih familija. Za svega 6 familija od 37 se minimalna vrednost proseka rastojanja postiže u poređenju sa nekom drugom familijom.

Ovi rezultati potvrđuje hipotezu da između formula koje pripadaju istoj familiji postoji sintaksna sličnost i da se ona može automatski prepoznati.



Slika 6.1: Proseci distanci između formula za svaki par familija. Svetlije nijanse označavaju manje vrednosti, a tamnije veće.

6.2.2 Eksperimenti vezani za klasifikovanje formula

Stepen osnovanosti drugog dela hipoteze koji pretpostavlja da se familija kojoj formula pripada može automatski prepoznati na osnovu njene sintakse i znanja o drugim formulama, može se direktno proveriti sprovođenjem klasifikacije formula iz korpusa. Metode za klasifikovanje iskaznih formula su opisane u poglavlju 4. Radi evaluacije kvaliteta klasifikacije, eksperimenti su dizajnirani tako da koriste unakrsno ocenjivanje sa maksimalnim brojem slojeva, odnosno u svakom sloju ocenjivanja izdvajana je po jedna formula koja je klasifikovana na osnovu ostatka korpusa. Ovakav pristup dozvoljava učešće svih formula korpusa u evaluaciji, trening skup je uvek maksimalan, a varijacije između modela dobijenih u različitim slojevima su minimalne. Eksperimenti su dizajnirani po jedinstvenom šablonu za sve načine klasifikovanja.

U eksperimentima sa n-gramima n je uzimalo vrednosti 1, 2, 3 i 5,

a dužina n-gramskog profila je bila ograničena na najfrekventnijih 1000 n-grama.

Za klasifikaciju zasnovanu na frekvencijama podgrafova, rađeni su eksperimenti sa podgrafovima od 4 i 5 čvorova. Podgrafovi sa 3 čvora nisu razmatrani zato što ima malo neizomorfnih grafova sa tim brojem čvorova. S druge strane, zbog računske složenosti procesa semplovanja, pravljenje profila za podgrafove veličine 6 ili više je previše zahtevno. Broj semplovanja u slučaju podgrafova veličine 4 bio je 20000, a u slučaju podgrafova veličine 5, bio je 100000. U frekventnim profilima zatupljene su frekvencije svih pronađenih podgrafova pošto ih obično nema više od 25.

U ovim, kao i u eksperimentima sa klasifikacijom zasnovanom na izabranom skupu sintaksnih svojstava formula, korišćene su distance date u poglavlju 4.1.

Prilikom korišćenja metode k najbližih suseda, k je uzimalo vrednosti 1, 3, 5 i 7 u svim eksperimentima.

Formule koje nisu mogle biti obrađene pomoću nekih pristupa zbog računske zahtevnosti su izbačene iz korpusa u tim eksperimentima, pa je stoga u eksperimentalnim rezultatima navedeno i koliko formula je moglo biti obrađeno pomoću tog pristupa.

Vremena potrebna za klasifikovanje različitim pristupima

Kreiranje profila instanci nije podjednako zahtevno za sve pristupe. Stoga ono predstavlja značajan faktor prilikom izbora najboljeg načina klasifikovanja. Za svaki od načina biće dati broj obrađenih instanci, prosečno vreme, ukupno vreme, vreme za najmanje zahtevnu instancu i vreme za najzahtevniju instancu. Ovi podaci su prikazani u tabeli 6.1.

	Instanci	Prosek	Ukupno	Min	Max
Monogrami	1964	0.01	15.92	0.002	0.52
Podgrafovi veličine 5	1945	21.18	41203.88	4.00	1980.80
Izabrani skup svojstava	1964	0.39	765.06	0.002	62.3

Tabela 6.1: Vremena izračunavanja profila instanci pomoću različitih pristupa izražena u sekundama.

Prvi i poslednji pristup su u stanju da obrade sve formule iz korpusa u kratkom vremenu. Drugi pristup to ne može. Stoga on mora biti u zaostatku u praktičnoj primeni.

Za klasifikovanje bazirano na izabranom skupu sintaksnih svojstava formule, vreme određivanja familije kojoj formula pripada nakon što je profil formule kreiran je manje od 0.01s. Kako su profili zasnovani na izabranom skupu sintaksnih svojstava formule najduži, vreme određivanja familije za druge vrste profila je manje ili jednako i stoga se generalno može smatrati zanemarljivim.

Rezultati klasifikovanja formula baziranog na n-gramima

Kako pravljenje n-gramskih profila nije mnogo računski zahtevno, samo 2 formule su isključene iz korpusa pošto predstavljaju jednočlane familije, odnosno korišćene su 1962 formule, a broj familija je 37. Jednočlane familije se izbacuju jer je prilikom klasifikovanja instance iz takve familije nemoguća tačna klasifikacija, odnosno ne postoji kvalitetan trening skup, što može značajno da utiče na prosek preciznosti po familijama. Rezultati su dati samo za distancu d_1, definisanu u poglavlju 4.1, pošto se ona u ovom slučaju pokazala kao najbolja. I u ostalim pristupima klasifikovanju će biti dati podaci samo za najbolje funkcije distnaci. Rezultati su navedeni u tabeli 6.2. Prilikom njihovog prezentovanja klasifikovanja dat je broj familija, broj formula koje su učestvovale u klasifikovanju, preciznost klasifikovanja i prosečna preciznost po pojedinačnim familijama. Ovi podaci su prezentovani za različite dužine n-gramskih profila i za različit broj suseda k u metodi k najbližih suseda.

n	k	Preciznost	Prosek preciznosti po familijama
1	1	82.8%	53.5%
1	3	75.4%	40.9%
1	5	70.4%	37.4%
1	7	68.2%	37.3%
2	1	47.1%	48.8%
2	3	43.3%	40.2%
2	5	41.5%	36.5%
2	7	41.8%	35.5%
3	1	53.7%	36.2%
3	3	53.8%	30.4%
3	5	56.0%	27.6%
3	7	56.5%	25.2%
5	1	26.7%	20.3%
5	3	27.0%	20.2%
5	5	26.2%	20.6%
5	7	24.1%	17.8%

Tabela 6.2: Rezultati klasifikovanja pomoću n-grama za distancu d_1.

Iz tabele se vidi da povećavanje parametara n i k dovodi do smanjivanja kvaliteta klasifikacije. Pri tome, ako se razmatra ukupna preciznost klasifikacije pad kvaliteta deluje nemonoton, ali prosek preciznosti po familijama, pada monotono. Preciznost od 82.8% za $n = 1$ i $k = 1$ se može smatrati zadovoljavajućom, posebno s obzirom na broj familija, ali imajući u vidu prosečnu preciznost po familijama od 53.5%, jasno je da je zavisna

od raspodele instanci.

Interesantan rezultat je da se najbolja preciznost postiže za dužinu n-gramskih profila $n = 1$. Za ovo se može ponuditi jednostavno objašnjenje. Svaka linija tekstualnog zapisa formule se završava karakterom 0. Stoga veća frekvencija ovog karaktera u ukupnom broju karaktera sugeriše da se formula sastoji iz većeg broja kraćih klauza, dok njeno ređe pojavljivanje sugeriše manji broj dužih klauza. Manja frekvencija razmaka sugeriše veći broj promenljivih. Na primer, u slučaju da postoji najviše 9 promenljivih, razmak se javlja posle svakog karaktera osim nule. U slučaju da postoji veći broj promenljivih, razmak se javlja samo posle poslednjeg karaktera u zapisu rednog broja promenljive. Ove informacije mogu da ukažu na neke zakonitosti u formulama, a moguće je da su na sličan način izražene i druge.

Zanimljiv je i rezultat da se najbolja preciznost postiže za broj najbližih suseda $k = 1$. Iz tabele 6.2 se vidi da prosečna preciznost po familijama brže opada nego ukupna preciznost, što znači da se povećavanjem parametra k više gubi na kvalitetu klasifikacije za formule iz manjih familija. Moguće je da za formule iz malih familija postoji manje dovoljno bliskih suseda za veće k, dok su formule iz većih klasa zbog svoje brojnosti gušće raspoređene u prostoru izabranih atributa. Ukoliko formule iz različitih familija nisu jasno razdvojene u prostoru atributa, ovo bi bilo očekivano ponašanje. Ova primedba važi i za ostale pristupe klasifikovanju.

Rezultati klasifikovanja grafova baziranog nza frekvencijama podgrafova

U ovim eksperimentima korišćene su 1943 formule. Bile su zastupljene 33 familije. Smanjenje u broju formula i broju familija potiče od izbacivanja formula koje su bile prevelike za obradu ovim metodom. Pri tome su iz korpusa izbačene i familije koje su tako postale jednočlane.

Rezultati za najbolju distancu — distancu d_4 definisanu u poglavlju 4.1 su navedeni u tabeli 6.3. Prezentovane su mere kvaliteta za različite brojeve čvorova podgrafova n i za različit broj suseda k u metodi k najbližih suseda.

Kvalitet klasifikacije je značajno bolji pri korišćenju podgrafova veličine 5. Razlog za to je što je broj neizomorfnih podgrafova veličine 5 znatno veći nego broj neizomorfnih podgrafova veličine 4, što omogućava veći broj dimenzija (frekvencija različitih podgrafova) po kojima bi se mogla izraziti razlika između formula iz različitih familija. S druge strane, sa povećanjem parametra k, kvalitet klasifikacije opada. Pri tome je dosta izraženiji pad proseka preciznosti po familijama, nego same preciznosti, što sugeriše da je klasifikovanje u ovom slučaju prisrasno u korist familija sa većim brojem instanci. Kvalitet klasifikacije je u ovom pristupu izuzetno dobar (posebno za $n = 5$ i $k = 1$). S druge strane, visoko vreme računanja profila instanci može predstavljati problem za praktičnu upotrebu.

n	k	Preciznost	Prosek preciznosti po familijama
4	1	93.0%	91.7%
4	3	93.0%	83.4%
4	5	93.1%	78.3%
4	7	92.1%	67.5%
5	1	98.7%	94.7%
5	3	97.7%	86.3%
5	5	97.1%	79.9%
5	7	96.4%	69.7%

Tabela 6.3: Rezultati klasifikovanja pomoću frekvencija podgrafova za distancu d_4.

Rezultati klasifikovanja formula baziranog na izabranom skupu sintaksnih svojstava

Pravljenje profila ni u ovom pristupu, kao ni u pristupu baziranom na n-gramima, nije mnogo računski zahtevno, pa su korišćene 1962 formule razvrstane u 37 familija. Rezultati za najbolju distancu — d_3, definisanu u poglavlju 4.1, su navedeni u tabeli 6.4.

k	Preciznost	Prosek preciznosti po familijama
1	98.6%	91.5%
3	97.1%	75.3%
5	96.6%	68.3%
7	95.8%	60.0%

Tabela 6.4: Rezultati klasifikovanja baziranog na izabranom skupu sintaksnih svojstava za distancu d_3.

Primećuje se da povećanje parametra k dovodi do smanjenja kvaliteta klasifikacije, kao i kod drugih pristupa. Pri tome, prosek preciznosti po familijama opada brže što znači da postoje velike favorizovane familije. Kvalitet klasifikacije se može smatrati izuzetnim kako po ukupnoj preciznosti, tako i po prosečnoj preciznosti po familijama. S obzirom na vreme potrebno za računanje profila instanci, ovaj pristup je praktično vrlo upotrebljiv.

Kako je pristup baziran na frekvencijama podgrafova vremenski značajno zahtevniji od ostalih, a po preciznosti klasifikacije je zanemarljivo bolji u odnosu na pristup baziran na izabranom skupu sintaksnih svojstava, on nije od praktičnog značaja. Pristup baziran na monogramima ima nižu preciznost klasifikacije, ali je izuzetno brz, tako da se može smatrati upotre-

bljivim. Pristup baziran na skupu izabranih sintaksnih svojstava ima oba potrebna kvaliteta — profili instanci se vrlo brzo računaju, a preciznost klasifikacije je visoka, pa je najbolji za praktičnu upotrebu.

Visoka preciznost klasifikacije, kao i slika 6.1 pokazuju da je prva hipoteza ovog rada tačna, odnosno da postoji sintaksna sličnost među formulama iz iste familije koja se može automatski prepoznati i da se ta sličnost može iskoristiti za prepoznavanje familije kojoj nepoznata formula pripada. Vreme potrebno za klasifikovanje instance kada je izračunat njen profil je zanemarljivo — manje od jedne sekunde.

6.3 Evaluacija svojstava vrednosti parametara

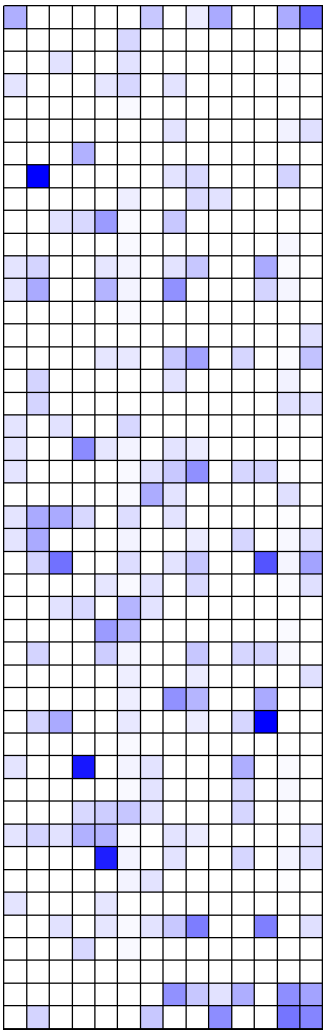
Druga hipoteza iz glave 1 pretpostavlja da za skupove formula koje su sintaksno slične postoje dominantno najbolje vrednosti parametara. Da bi se ovo proverilo, za svaku kombinaciju vrednosti parametara biće izračunato za koji procenat formula iz svake familije je ona najbolja. Pošto postoji značajan broj vrlo lakih formula koje bivaju rešene za manje od 0.001 sekundi za sve parametre, ne može se garantovati da bi ovakva tabela mogla da prikaže jasnu dominantnost neke kombinacije vrednosti parametara ukoliko ona postoji. Stoga će se računati procenti formula po familijama za koje su određene vrednosti parametara strogo bolje od ostalih.

Rezultati će biti prikazani za familije koje imaju bar 20 formula rešenih za bar jedne vrednosti parametara. Slikovit grafički prikaz dat je na slici 6.2. Tamnije nijanse označavaju veće vrednosti, a svetlije manje. Najveća vrednost je 30, a najmanja 0. Tačne vrednosti su date u tabeli 6.5.

Broj kombinacija vrednosti parametara koje su prikazane je 45 zbog toga što 15 kombinacija ni za jednu formulu nije strogo bolje od ostalih. Pri tome, 9 od njih nikad nisu najbolji ni zajedno sa drugim kombinacijama. Ono što je karakteristično za ove kombinacije je da je u svakoj politika izbora promenljive VS_RANDOM.

Iz tabele se vidi da ne postoje jedinstvene dominantne vrednosti parametara za formule iz neke familije, ali da često postoji nekoliko kombinacija vrednosti parametara koji dominiraju za određenu familiju dok se većina može zanemariti. Stoga druga hipoteza iz glave 1 nije u potpunosti potvrđena, ali jeste u izvesnoj meri.

Treća hipoteza pretpostavlja da su za sintaksno slične formule, najbolje kombinacije vrednosti parametara takođe slične. Sintaksna sličnost formula je već definisana distancama iz poglavlja 4.1. Može se izabrati jedna od njih, ili definisati nova. Potrebno je definisati neku distancu i nad vrednostima parametara. U tu svrhu je korišćena funkcija koja razlici u vrednosti svakog parametra pridružuje određenu težinu. Rastojanje je definisano kao zbir tih težina. O ovome će biti više reči u daljem tekstu. Kako bi se uporedile sličnosti formula i sličnosti najboljih vrednosti parametara biće sproveden sledeći postupak:



Slika 6.2: Grafički prikaz procenata broja formula po familijama za koje su određene vrednosti parametara strogo bolje od ostalih. Vrste odgovaraju vrednostima parametara, a kolone familijama. Prikazani su podaci samo za familije koje imaju bar 20 formula rešenih za bar jedne vrednosti parametara.

1. Za svake dve formule iz korpusa:
   (a) Izračunati rastojanje između njih.
   (b) Izračunati rastojanje između najboljih vrednosti parametara za te formule.
2. Izračunati Pirsonov koeficijent korelacije za izračunata rastojanja između formula i rastojanja između njihovih najboljih vrednosti parametara.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	9.38	0.00	0.00	0.00	0.00	0.00	6.45	0.00	2.17	10.00	0.00	0.00	9.82	17.86
2	0.00	0.00	0.00	0.00	0.00	4.67	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
3	0.00	0.00	3.33	0.00	0.00	3.33	0.00	0.00	0.00	0.00	0.00	0.00	0.18	0.00
4	3.12	0.00	0.00	0.00	2.94	4.67	0.00	3.23	0.00	0.00	0.00	0.00	0.18	0.00
5	0.00	0.00	0.00	0.00	0.00	0.67	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
6	0.00	0.00	0.00	0.00	0.00	0.00	0.00	3.23	0.00	0.00	0.00	0.00	1.45	3.57
7	0.00	0.00	0.00	9.09	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
8	0.00	30.00	0.00	0.00	0.00	0.00	0.00	3.23	4.35	0.00	0.00	0.00	5.09	0.00
9	0.00	0.00	0.00	0.00	0.00	2.00	0.00	0.00	4.35	3.33	0.00	0.00	0.00	0.00
10	0.00	0.00	3.33	4.55	11.76	0.67	0.00	6.45	0.00	0.00	0.00	0.00	0.00	0.00
11	0.00	0.00	0.00	0.00	0.00	0.67	0.00	0.00	0.00	0.00	0.00	0.00	0.91	0.00
12	3.12	5.00	0.00	0.00	2.94	1.33	0.00	3.23	6.52	0.00	0.00	10.00	0.55	0.00
13	3.12	10.00	0.00	0.00	8.82	1.33	0.00	12.90	0.00	0.00	0.00	5.00	1.09	0.00
14	0.00	0.00	0.00	0.00	0.00	0.67	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
15	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	3.57
16	0.00	0.00	0.00	0.00	2.94	2.67	0.00	6.45	10.87	0.00	4.76	0.00	0.36	7.14
17	0.00	5.00	0.00	0.00	0.00	0.00	0.00	3.23	0.00	0.00	0.00	0.00	1.45	0.00
18	0.00	5.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	3.45	3.57
19	3.12	0.00	3.33	0.00	0.00	4.67	0.00	0.00	0.00	0.00	0.00	0.00	0.18	0.00
20	3.12	0.00	0.00	13.64	2.94	1.33	0.00	3.23	2.17	0.00	0.00	0.00	0.00	0.00
21	3.12	0.00	0.00	0.00	0.00	0.67	3.23	6.45	13.04	0.00	4.76	5.00	0.18	0.00
22	0.00	0.00	0.00	0.00	0.00	0.67	9.68	3.23	0.00	0.00	0.00	0.00	3.64	0.00
23	3.12	10.00	10.00	4.55	0.00	4.00	0.00	3.23	0.00	0.00	0.00	0.00	0.18	0.00
24	3.12	10.00	0.00	0.00	0.00	1.33	0.00	0.00	2.17	0.00	4.76	0.00	0.73	3.57
25	0.00	5.00	16.67	0.00	0.00	4.00	0.00	3.23	6.52	0.00	0.00	20.00	1.09	10.71
26	0.00	0.00	0.00	0.00	2.94	0.67	3.23	0.00	4.35	0.00	0.00	0.00	0.36	3.57
27	0.00	0.00	3.33	4.55	0.00	8.67	3.23	0.00	0.00	0.00	0.00	0.00	0.00	0.00
28	0.00	0.00	0.00	0.00	11.76	8.00	0.00	0.00	0.00	0.00	0.00	0.00	0.55	0.00
29	0.00	5.00	0.00	0.00	5.88	1.33	0.00	0.00	6.52	0.00	4.76	5.00	0.73	0.00
30	0.00	0.00	0.00	0.00	0.00	2.00	0.00	0.00	2.17	0.00	0.00	0.00	0.00	3.57
31	0.00	0.00	0.00	0.00	0.00	2.00	0.00	12.90	8.70	0.00	0.00	10.00	0.00	0.00
32	0.00	5.00	10.00	0.00	0.00	2.67	0.00	0.00	2.17	0.00	4.76	30.00	0.00	0.00
33	0.00	0.00	0.00	0.00	0.00	0.67	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
34	3.12	0.00	0.00	27.27	0.00	1.33	3.23	0.00	0.00	0.00	9.52	0.00	0.55	0.00
35	0.00	0.00	0.00	0.00	0.00	0.67	3.23	0.00	0.00	0.00	4.76	0.00	0.73	0.00
36	0.00	0.00	0.00	4.55	5.88	6.67	3.23	0.00	0.00	0.00	4.76	0.00	0.18	0.00
37	3.12	5.00	3.33	9.09	8.82	0.67	0.00	3.23	2.17	0.00	0.00	0.00	0.00	3.57
38	0.00	0.00	0.00	0.00	26.47	1.33	0.00	3.23	0.00	0.00	4.76	0.00	1.27	3.57
39	0.00	0.00	0.00	0.00	0.00	1.33	3.23	0.00	0.00	0.00	0.00	0.00	0.18	0.00
40	3.12	0.00	0.00	0.00	2.94	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
41	0.00	0.00	3.33	0.00	2.94	0.67	3.23	6.45	15.22	0.00	0.00	15.00	0.00	3.57
42	0.00	0.00	0.00	4.55	0.00	0.67	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
43	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
44	0.00	0.00	0.00	0.00	0.00	0.00	0.00	12.90	6.52	3.33	9.52	0.00	13.27	10.71
45	0.00	5.00	0.00	0.00	0.00	0.00	6.45	0.00	0.00	13.33	0.00	0.00	16.18	14.29

Tabela 6.5: Tabela procenata broja formula po familijama za koje su određene vrednosti parametara strogo bolje od ostalih. Vrste odgovaraju vrednostima parametara, a kolone familijama. Prikazani su podaci samo za familije koje imaju bar 20 formula rešenih za bar jedne vrednosti parametara.

Osnovanost hipoteze biće procenjena na osnovu vrednosti koeficijenta korelacije. Prilikom njenog testiranja korišćeni su profili zasnovani na izabranom skupu sintaksnih svojstava formula (poglavlje 4.4) zbog visoke preciznosti pri klasifikaciji i jer se pomoću njih mogu obraditi sve formule iz korpusa. Kao distanca nad profilima korišćena je funkcija d_3 definisana u poglavlju 4.1. Težinski koeficijenti u funkciji rastojanja nad vrednostima parametara su podešeni tako da se dobije što veći koeficijent korelacije. Ovakav postupak je legitiman i predstavlja traženje odgovora na pitanje sa kojim su promenama u vrednostima parametara promene u sintaksnim svojstvima najviše korelirane.

Za vrlo lake formule postoji veliki broj najboljih vrednosti parametara.

Takođe, na vreme rešavanja formule osim njene težine i izabranih parametara može uticati i opterećenost sistema kroz prebacivanje konteksta procesora nevezano za rešavanje formule. Kako je to operacija čije trajanje ne zavisi od težine formule, dodatno vreme koje od nje potiče je izraženije za lake formule. Stoga su vremena rešavanja, a time i izbor najboljih parametara, za takve formule nepouzdana i one će biti zanemarene prilikom računanja koeficijenta korelacije između rastojanja među formulama i rastojanja među njihovim najboljim vrednostima parametara. Uzete su u obzir samo formule sa vremenom rešavanja dužim od 2 sekunde za najbolje vrednosti parametara, a koje su pomoću njih rešene u okviru ograničenja od 600 sekundi.

Izracunati koeficijent korelacije jednak je 0.51. Ova vrednost se može smatrati srednjom, ali značajnom korelacijom. Naime, proces rešavanja iskaznih formula je inherentno nestabilan. Može se desiti da se vreme rešavanja pri preimenovanju promenljivih razlikuje za red veličine u odnosu na polazno. Takođe, rezolucija prostora parametara se može smatrati niskom. U ovakvom kontekstu postignuta je značajna vrednost koeficijenta korelacije, polazna hipoteza se može u značajnoj meri smatrati potvrđenom.

U funkciji rastojanja nad vrednostima parametara najveći težinski koeficijenti odgovaraju promenama u politici otpočinjanja iznova, nešto manji promenama u politici izbora promenljive, a najmanji su vezano za politiku izbora polariteta promenljive. Iz toga sledi da su za sintaksno različite formule najrazličitije za njih najbolje politike otpočinjanja iznova, pa izbora promenljive, pa izbora polariteta promenljive i obratno — u slučaju sintaksno sličnih formula najbolja politika otpočinjanja iznova najmanje varira, politika izbora promenljive nešto više, a najmanje su bitne promene u politici izbora polariteta promenljive. To znači da za skup parametara koji je korišćen u ovom radu, pogodna politika otpočinjanja iznova predstavlja najvažniju zajedničku karakteristiku sintaksno sličnih formula.

6.4 Evaluacija pristupa za izbor vrednosti parametara

Do sada prikazani rezultati se odnose na različite delove analize prepoznavanja familije nepoznate formule i ponašanja parametara SAT rešavača. U ovom podpoglavlju su dati ključni rezultati primene metodologije iz kojih se može oceniti njena primenljivost u praksi. Rezultati eksperimenata su dati za pristupe bazirane na monogramima, frekvencijama podgrafova i izabranom skupu sintaksnih svojstava. Za vrednosti parametara k i n uzete su najbolje vrednosti navedene u poglavlju 6.2. Takođe su, radi poređenja, prikazane vrednosti mera kvaliteta za najbolje fiksirane i za najbolje vrednosti parametara.

Najvažnije statistike su date u tabeli 6.6. To su broj formula na kojima je pristup testiran, broj rešenih formula, središnje vreme rešavanja i ukupno

vreme rešavanja svih formula. U navedena vremena rešavanja nije uračunato vreme pravljenja profila instanci. Ti podaci su dati u tabeli 6.1. Ova tabela se pre svega odnosi na kvalitet interakcije između procesa klasifikovanja i primene izabranih vrednosti parametara. U tabeli 6.6, verzija metodologije (slike 5.2 i 5.3) je označena brojem u indeksu. Ukupan broj formula je 1964.

Pristup	Br. rešenih	Središnje vr.	Ukupno vr.
Najbolje fiksirane	1073	207.45s	162.64h
Monogrami_1	1113	112.60s	154.34h
Monogrami_2 $m = 36$, $n = 32$	1119	128.96s	156.28h
Podgrafovi_1	1126	88.81s	149.01h
Podgrafovi_2 $m = 34$, $n = 34$	1117	101.76s	151.52h
$\text{Sintaksna svojstva}_1$	1135	92.64s	151.14h
$\text{Sintaksna svojstva}_2$ $m = 20$, $n = 23$	1132	108.40s	153.07h
Najbolje	1187	46.08	141.50h

Tabela 6.6: Efikasnost rešavanja za različite načine izbora parametara.

U svim pristupima se primećuje značajno poboljšanje u odnosu na najbolje fiksirane vrednosti parametara. To znači da se svim pristupima mogu iskoristiti zakonitosti koje postoje u sintaksi formula. Takođe se zaključuje najbolje vrednosti parametara razlikuju od familije do familije ili od formule do formule, tako da se performanse najboljih fiksiranih vrednosti parametara ne mogu dovoljno približiti najboljim. To ostavlja prostor za primenu metoda koje prilagođavaju vrednosti parametara instanci koja se rešava.

Kao najbolji pristup, izdvaja se pristup baziran na izabranom skupu sintaksnih svojstava sa prvom verzijom metodologije koji je bolji od ostalih po broju rešenih formula. Središnje vreme je jedino bolje u pristupu baziranom na frekvencijama podgrafova, ali je taj pristup računski značajno zahtevniji u fazi klasifikovanja. Zanimljivo je da je pristup baziran na monogramima, iako vrlo jednostavan i računski isplativ, takođe postigao značajno poboljšanje u odnosu na najbolje fiksirane vrednosti parametara.

Može se primetiti da je u najefikasnijem pristupu rešeno preko 50% formula koje ostaju nerešene pri korišćenju najboljih fiksiranih vrednosti parametara, a mogu se rešiti korišćenjem nekog skupa vrednosti parametara. Ovaj procenat je posebno značajan s obzirom na eksponencijalni rast težine formula uočljiv na grafiku 6.6 koji će biti diskutovan kasnije.

Bitna primedba vezana za prikazane rezultate je da su za parametre m i n druge verzije metodologije (slika 5.3) nađene najbolje vrednosti na korišćenom korpusu pri unakrsnom ocenjivanju, odnosno za razne vrednosti ovih parametara, izvršen je postupak evaluacije, a rezultati su prikazani za

najbolje. Zbog ovakvog prilagođavanja parametara korišćenom korpusu, ne može se sa sigurnošću tvrditi da će kvalitet ovih pristupa biti održan na nekom drugom korpusu. Pristupi zasnovani na prvoj verziji metodologije nemaju ovakve parametre, pa se na njih ova primedba ne odnosi.

U tabeli 6.7 date su vrednosti statistika koje se odnose na indekse izabranih vrednosti parametara — prosečan indeks u nizu sortiranom po vremenu rešavanja i udeo najboljih vrednosti parametara u izabranim.

Pristup	Prosečan indeks	Udeo najboljih
Najbolje fiksirane	10.68	13.73%
Monogrami_1	6.80	27.21%
Monogrami_2 $k = 23$, $n = 20$	8.60	21.06%
Podgrafovi_1	5.79	28.35%
Podgrafovi_2 $k = 34$, $n = 34$	7.05	25.13%
$\text{Sintaksna svojstva}_1$	5.86	28.22%
$\text{Sintaksna svojstva}_2$ $k = 23$, $n = 20$	6.49	24.18%
Najbolje	1	100%

Tabela 6.7: Statistike vezane za indekse izabranih vrednosti parametara.

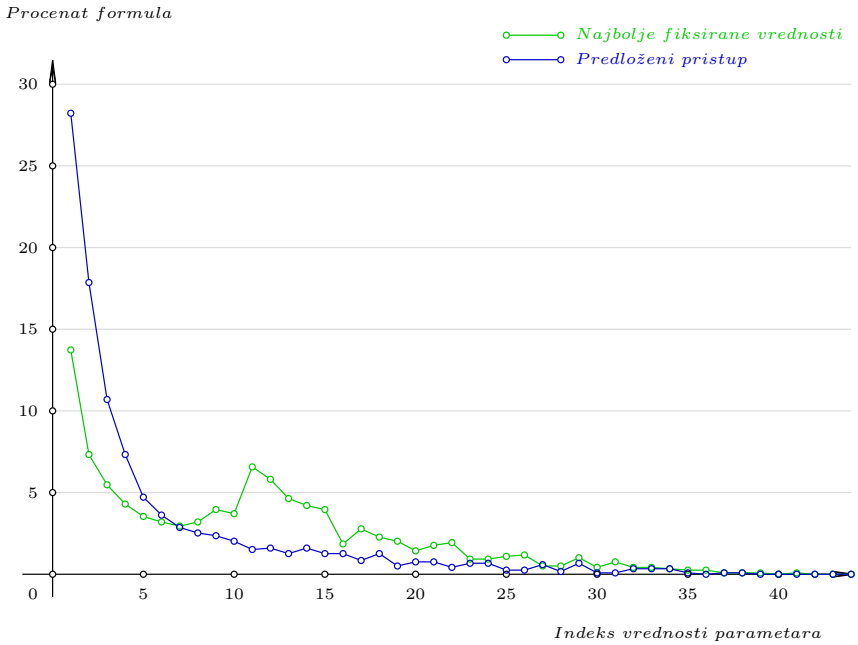
Poboljšanje u odnosu na najbolje fiksirane vrednosti parametara je prisutno i prema ovom skupu statistika.

Ostale statistike predložene u podpoglavlju 5.7.2 će zbog obima biti prikazane samo za najbolji pristup — zasnovan na izabranom skupu sintaksnih svojstava i prvoj verziji metodologije (slika 5.2).

Grafici koji prikazuju procenat formula za koje su pri određenom pravilu izbora izabrane vrednosti parametara sa određenim indeksom prikazani su na slici 6.3. Grafik za najbolje vrednosti parametara nije prikazan pošto ima maksimalnu vrednost za indeks 1 a vrednost 0 za sve ostale.

Raspodela za predloženi pristup pokazuje kako bolje rezultate tako i veću stabilnost u odnosu na najbolje fiksirane vrednosti parametara kod kojih se osim maksimuma na indeksu 1 javlja i značajan lokalni maksimum na indeksu 11. Primećuje se da je grafik koji odgovara predloženom pristupu za male vrednosti indeksa iznad, a za velike uvek ispod grafika koji odgovara pristupu baziranom na najboljim fiksiranim vrednostima. Ovo znači da se pri predloženom pristupu češće biraju dobre vrednosti parametara nego pri korišćenju najboljih fiksiranih vrednosti, dok se loše vrednosti parametara biraju ređe.

Dobra ilustracija procenta formula za koje se biraju pogodnije vrednosti parametara je kumulativna funkcija raspodele koja pokazuje za koji procenat formula su izabrane vrednosti parametara sa određenim ili manjim

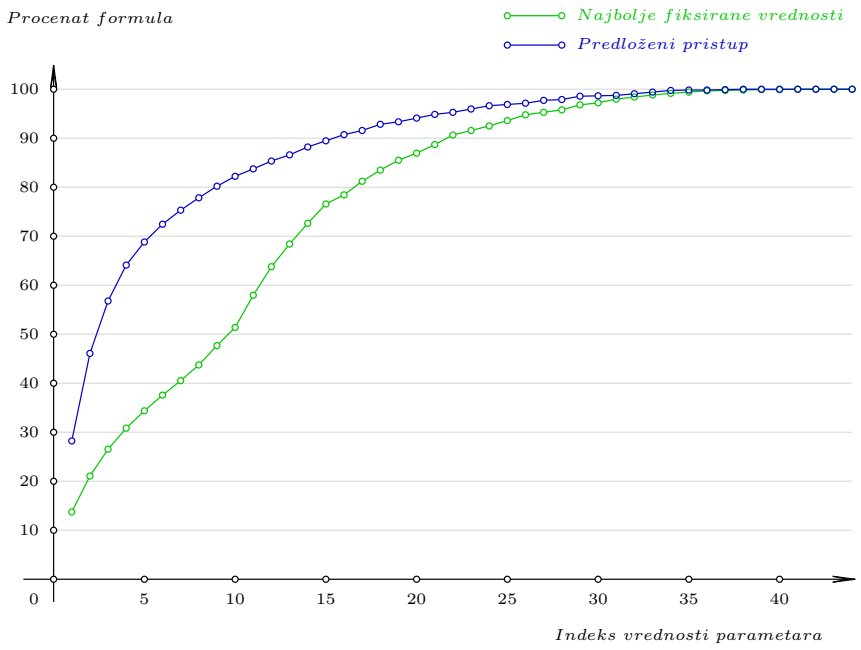


Slika 6.3: Raspodele indeksa izabranih parametara.

indeksom. Ove funkcije su prikazane na slici 6.4. Vidi se da predloženi pristup za značajno veći procenat formula bira bolje vrednosti parametara od pristupa koji koristi najbolje fiksirane vrednosti.

Histogram na slici 6.5 prikazuje procenat formula čije vreme rešavanja pomoću predloženog pristupa predstavlja određeni procenat njihovog vremena rešavanja pomoću najboljih fiksiranih vrednosti parametara. Na x osi se nalaze intervali procenata vremena, a na y osi, procenti formula sa takvim vremenom rešavanja. Procenti formula su računati u odnosu na ukupan broj rešivih formula. Procenti vremena rešavanja se kreću od 0 do 200. Interval od -10 do 0 se odnosi na formule koje su rešene pomoću predloženog pristupa, a nisu pomoću fiksiranih vrednosti parametara. Od 200 do 210 su svrstane formule sa procentom preko 200, bez obzira koliki je, a interval od 210 do 220 se odnosi na formule koje su rešene pomoću fiksiranih vrednosti parametara, ali ne i pomoću predloženog pristupa. Naravno, u razmatrenje su uzimane samo formule koje su rešene pomoću bar jednog od ova dva pristupa. Vidi se da su vrednosti na levoj polovini histograma ($< 100\%$) značajno veće nego vrednosti na desnoj, što znači da je rešavanje većeg broja formula ubrzano, a da je broj formula čije je rešavanje usporeno značajno manji.

Sortirani niz vremena rešavanja formula grafički je prikazan na slici 6.6. Crvena boja predstavlja vremena rešavanja pri najboljem izboru vrednosti

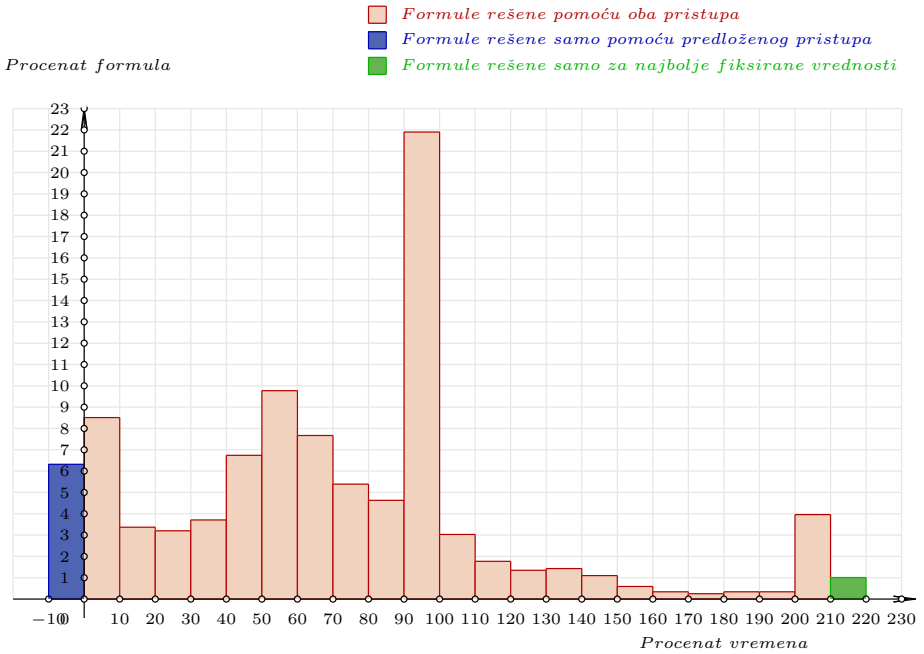


Slika 6.4: Kumulativna funkcija raspodele indeksa izabranih parametara.

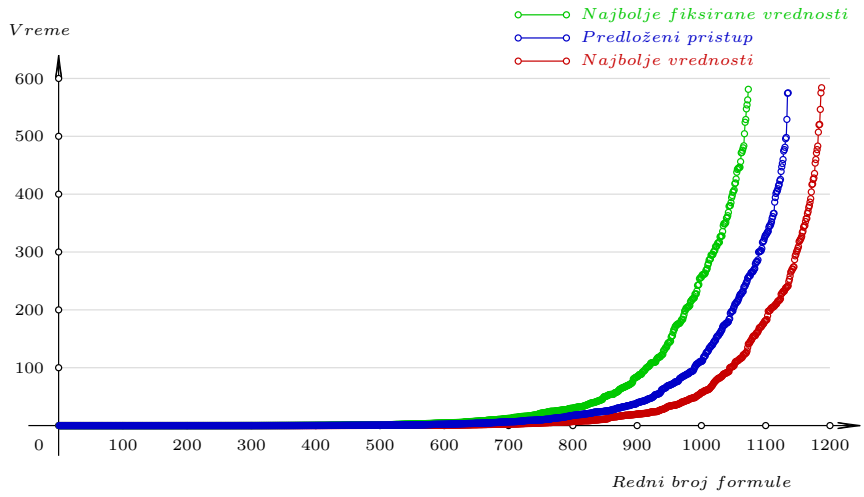
parametara. Plava se odnosi na predloženu metodologiju, a zelena na najbolje fiksirane vrednosti parametara. Ista značenja boja će važiti i na sledećem grafiku. U slučaju grafika na slici 6.6 treba primetiti da su nizovi vremena za različite načine izbora vrednosti parametara nezavisno sortirani, odnosno da za istu vrednost x koordinate mogu biti prikazana vremena rešavanja različitih formula. Uz ilustraciju koju pruža ovaj grafik može se bolje razumeti značaj ostvarnih rezultata. Sa njega se vidi da vrednosti niza pokazuju eksponencijalan rast i da je stoga bilo kakav napredak u broju rešenih formula dosta teško ostvariti.

Kumulativna funkcija raspodele data je na slici 6.7. Na x osi je prikazan logaritam za osnovu 10 vremena rešavanja, a na y osi broj formula rešiv u odgovarajućem vremenu. Sa svih grafika se vidi jasno poboljšanje u odnosu na najbolje fiksirane vrenosti parametara. I po vrednostima iz datih tabela i po graficima može se zaključiti da je postignuti napredak na nešto više od pola puta između najboljih fiksiranih i najboljih vrednosti parameatara.

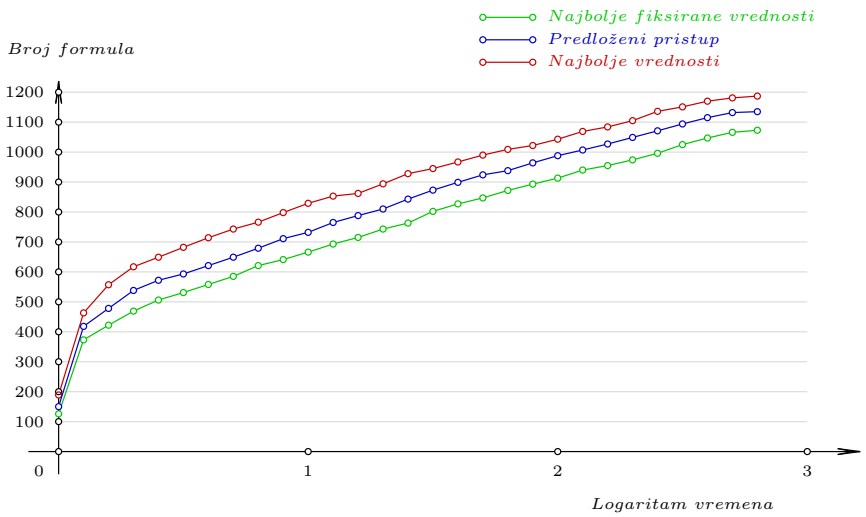
Prikazani rezultati potvrđuju osnovnu hipotezu ovog rada — da se inteligentnim biranjem vrednosti parametara SAT rešavača, zasnovanim na analizi sintakse formule koja se rešava, može povećati njegova efikasnost. Pri tome je pokazano da je najbolje koristiti profile zasnovane na skupu sintaksnih svojstava formule i funkciju rastojanja d_3 definisana u poglavlju 4.1.



Slika 6.5: Histogram broja formula čije je vreme rešavanja pomoću predloženog pristupa određeni procenat njihovog vremena rešavanja pomoću najboljih fiksiranih vrednosti parametara.



Slika 6.6: Vremena rešavanja formula iz korišćenog korpusa pri različitim načinima izbora vrednosti parametara.



Slika 6.7: Kumulativne funkcije raspodele broja rešenih formula u zavisnosti od ograničenja na vreme rešavanja.

6.5 Sistem ArgoSmArT

Sistem ArgoSmArT predstavlja proširenje rešavača ArgoSAT koje bira vrednosti parametara pogodne za instancu koja se rešava. U toku ovog istraživanja napravljen je veliki broj skripti u programskom jeziku PERL koje su izvršavale različite poslove kao što je klasifikacija, evaluacija, nalaženje najboljih vrednosti parametara za familije i slično. Nakon što su eksperimenti izvršeni pomoću ovih skripti, određeni njihovi delovi su prepisani u programskom jeziku *C* tako da čine objedinjen sistem. Izbori u dizajnu ovog sistema su doneti na osnovu eksperimentalnih rezultata — koriste se profili zasnovani na izabranom skupu sintaksnih svojstava i funkcija rastojanja d_3 definisana u poglavlju 4.1. Postoje dve varijante sistema — ArgoSmArT$_1$ i ArgoSmArT$_2$ od kojih svaka implementira jednu verziju faze eksploatacije predložene metodologije. Prva je opisana na slici 5.2, a druga na slici 5.3.

Zarad povećanja brzine izvršavanja, obe verzije sistema u svom kodu imaju profile instanci sa kojima se nepoznata instanca poredi. Takođe su ugrađene i informacije o najboljim parametrima za familije, odnosno pojedinačne instance. ArgoSmArT$_1$ kao ulaz uzima samo formulu koju je potrebno rešiti. Za parametar k u metodi k najbližih suseda se podrazumeva vrednost 1. Verzija ArgoSmArT$_2$ zahteva i zadavanje parametara n i m (ovi parametri objašnjeni su na slici 5.3). Po pokretanju bilo kog od ovih programa, za datu formulu se najpre određuje profil. Koriste se profili zasnovani na izabranom skupu sintaksnih svojstava. Pošto je određen profil formule, prepoznaju se pogodni parametri njegovim poređenjem sa profilima

drugih formula. Potom se formula rešava pokretanjem rešavača ArgoSAT za te vrednosti parametara.

6.5.1 Evaluacija sistema ArgoSmArT na korpusu SAT2007

Korpus formula sa takmičenja SAT rešavača iz 2007. godine će biti upotrebljen za proveru prenosivosti rezultata primene metodologije postignutih na jednom korpusu na drugi, kvalitativno različit korpus. Stoga će na njemu sistemom ArgoSmArT biti upoređen sa polaznim ArgoSAT rešavačem. Biće uzete u obzir obe verzije faze eksploatacije (slike 5.2 i 5.3) za pristup pomoću izabranog skupa sintaksnih svojstava. Druga verzija će biti testirana iako se pokazala kao lošija od prve. Pošto su vrednosti njenih parametara $m = 20$ i $n = 23$ izabrane tako da se dobija maksimalni broj rešenih formula pri unakrsnom ocenjivanju, postoji mogućnost da su one previše prilagođeni konkretnom trening skupu i da se pri prenosu na drugi korpus kvalitet sistema gubi. To se proverava na ovom korpusu. Pored toga, poređenje će biti izvršeno i sa sistemom SATzilla koji je opisan u poglavlju 3.2. Pri tome treba imati u vidu da su ti rezultati pre svega zanimljiva informacija, a da ne govore ništa o kvalitetu predložene metodologije, jer se on mora meriti u odnosu na polazni rešavač. Treba imati u vidu da cilj ovog rada nije konstrukcija najefikasnijeg SAT rešavača, već formulisanje metodologije kojom se može postići poboljšanje nekog konkretnog rešavača. Pristup na kome je sistem SATzilla baziran se ne bavi poboljšavanjem jednog SAT rešavača, već bira jedan od ponuđenih rešavača za koji prognozira da će najbrže rešiti formulu. Performanse oba pristupa zavise od toga na koje se rešavače primenjuju. Sistem SATzilla je uzet u obzir jer je jedini sistem koji se na bilo koji način prilagođava instanci koja se rešava, a pored toga, trenutno je najefikasniji sistem za rešavanje SAT problema.

Glavna mera kvaliteta je ponovo broj rešenih formula. Kako je ovaj korpus dosta teži od onog iz 2002. godine, središnje vreme ne može biti izračunato jer se u zadatom vremenskom intervalu od 600s ne može rešiti polovina formula iz korpusa. Stoga će središnje vreme biti zamenjeno 20-im percentilom skupa vremena rešavanja. Takođe je dato i ukupno vreme. Podaci su prikazani u tabeli 6.8.

Poboljšanje koje sistem ArgoSmArT$_1$ postiže na ovom korpusu ne zaostaje za onom koje je ostvareno na korpusu iz 2002. godine. Značajno poboljšanje je postignuto u odnosu na sve mere kvaliteta. Stoga se može smatrati da je pristup na kome je ovaj sistem baziran u potpunosti ispunio očekivanja. Sistem ArgoSmArT je treniran na korpusu iz 2002. Kao što je pomenuto u poglavlju 5.5, postoji 12 formula iz tog korpusa koje su sadržane u korpusu iz 2007. Kako ni jedna od ovih formula nije rešena, postignuto poboljšanje u odnosu na rešavač ArgoSAT nije posledica preklapanja trening i test skupova. Za razliku od sistema ArgoSmArT$_1$, ArgoSmArT$_2$ se pokazao značajno lošijim od polaznog rešavača. Moguće je dobiti nešto

Pristup	Br. rešenih	20-ti perc. vr.	Ukupno vr.
ArgoSAT	219	314.16s	123.60h
ArgoSmArT$_1$	239	237.24s	120.69h
ArgoSmArT$_2$ $m = 20$ $n = 23$	154	> 600s	132.19h
ArgoSmArT$_2$ $m = 16$ $n = 27$	178	> 600s	129.04h
SATzilla	364	14.99s	101.80h

Tabela 6.8: Rezultati poređenja na korpusu SAT2007.

bolje rezultate za vrednosti parametara $m = 16$ i $n = 27$ (koje na korpusu iz 2002. godine imaju nešto lošije rezultate). Ovo obrnuto rangiranje na korpusu iz 2007. godine predstavlja ilustraciju fenomena preteranog prilagođavanja naučenog modela trening podacima (eng. overfitting).

Razmatran je i jedan način poboljšanja performansi ArgoSmArT sistema za korpuse koji su po zastupljenim familijama različiti od onog na kome je trenirano. Naime, u slučaju da familija kojoj formula pripada nije zatupljena u trening korpusu, moguće je da vrednosti parametara koje sistem predlaže uopšte ne odgovaraju formuli koju je potrebno rešiti. U takvim slučajevima je rastojanje od razmatrane formule do njenog najbližeg suseda u trening skupu obično veliko i može se uzeti kao indikacija ovakvog problema. Razmatran je pristup u kome se odbacuju preporuke sistema ukoliko je to rastojanje iznad određenog praga i primenjuju se najbolje fiksirane vrednosti parametara. Kako je na taj način u slučaju ArgoSmArT$_1$ sistema bilo moguće rešiti samo još 2 formule, ova varijanta sistema se ne opisuje detaljnije i ne smatra se značajnom za dalja istraživanja.

Sistem SATzilla je superioran u odnosu na ostale kandidate, ali prikazano poređenje sa SATzilla sistemom zahteva dodatnu diskusiju. Kao što je rečeno, uspešnost oba pristupa zavisi od izabranih rešavača. Skup rešavača kojima SATzilla raspolaže predstavlja izbor najefikasnijih rešavača koji se pojavljuju na takmičenjima. Takođe, sistem SATzilla je treniran na višestruko većem korpusu. S druge strane, prostor vrednosti parametara kojima ArgoSmArT raspolaže se može smatrati prilično siromašnim. Sistem ArgoSmArT pre svega služi za demonstraciju praktične primenljivosti izložene metodologije, a još uvek ne kao visoko optimizovani sistem (iako je već pokazao značajna poboljšanja). O pravcima daljeg unapređenja se govori u glavi 7.

Glava 7

Dalji rad

Predstavljeni rezultati ostavljaju raznovrsne pravce daljek rada. Biće opisano nekoliko najznačajnijih.

7.1 Dalja analiza raspoloživih podataka

U ovom radu je sakupljena velika količina podataka o funkcionisanju SAT rešavača sa različitim vrednostima parametara i na različitim vrstama iskaznih formula. Kako je postupak sproveden sistematično, može se smatrati da se raspolaže kvalitetnim uzorkom nad kojim se mogu primenjivati različite tehnike istraživanje podataka ili statističke analize. Poznato je da se tehnike istraživanja podataka često mogu primenjivati i u cilju pronalaženja znanja bez unapred postavljenog pitanja o tome šta se traži. Jedno od važnih i jasno postavljenih pitanja na koje bi se mogao tražiti odgovor je koje vrednosti parametara se međusobno dobro slažu, a koje vrednosti ne bi trebalo kombinovati. Moguća tehnika za otkrivanje ovakvih zavisnosti bi bila nalaženje pravila pridruživanja.

Još jedna tema od potencijalnog istraživačkog značaja koja bi se mogla sprovesti na osnovu već sakupljenih podataka je proučavanje uticaja preimenovanja promenljivih i permutovanja klauza na vreme rešavanja formula. Bilo bi zanimljivo sprovesti takvu analizu u zavisnosti od familije kojoj formula pripada, kao i u zavisnosti od vrednosti parametara sa kojima je formula rešena.

7.2 Stohastička optimizacija parametara

Broj parametara i njihovih dopustivih vrednosti se u ovom radu može smatrati relativno malim, pa je verovatno da za razne formule postoje kombinacije vrednosti parametara za koje bi rešavač postizao bolje performanse. Najjednostavniji nastavak bi bio povećavanje broja razmatranih parametara i njihovih dopustivih vrednosti. Pri tome bi bilo potrebno paziti na

povećanje računske zahtevnosti. Dosta efikasnije rešenje bi bilo korišćenje stohastičkih metoda lokalne pretrage kao u radu [13], pošto se ovaj pristup pokazao kao vrlo uspešan u pronalaženju dobrih vrednosti parametara za datu familju. U kombinaciji sa metodom klasifikacije visoke preciznosti koja je opisana u ovom radu, takav pristup bi mogao da dâ vrlo dobre rezultate u praksi.

S obzirom na to da stohastičke metode pretrage omogućavaju rad sa mnogo većim brojem parametara nego sistematično rešavanje, za analizu odnosa ovih parametara bi se mogle koristiti statističke metode. Na taj način bi se moglo doći do znanja o tome koje vrednosti parametara dobro funkcionišu jedne sa drugima.

7.3 Ispitivanje stabilnosti najboljih vrednosti parametara u regionu fazne promene za 3-SAT

U stilu koji je opisan, planirano je iscrpno rešavanje instanci problema 3-SAT. Cilj ovog istraživanja je ispitivanje stabilnosti najboljih vrednosti parametara kada se instance problema približavaju tački fazne promene (skupu formula koje su najteže za rešavanje). Time bi se došlo do novih informacija o razlici u funkcionisanju SAT rešavača u blizini tačke fazne promene i daleko od nje.

7.4 Učenje upravljanja SAT rešavačem

Pomenuti pravci daljeg rada ili predstavljaju primenu iste metodologije sa drugačijim parametrima ili domenom ili se direktno oslanjaju na podatke prikupljene u dosadašnjem istraživanju. Nešto drugaciji pristup bi se umesto na prilagođavanju polaznih parametara bazirao na konstantnom prilagođavanju rešavača u toku procesa rešavanja. Naime, rad rešavača bi se mogao modelovati Markovljevim procesom odlučivanja. Za učenje optimalnih politika odlučivanja u Markovljevim procesima odlučivanja koristi se tehnika učenja uslovljavanjem. Na taj način bi se mogla naučiti optimalna politika upravljanja rešavačem ukoliko se definiše skup stanja procesa odlučivanja i skupovi akcija koje se mogu preduzeti u različitim stanjima. U svakom trenutku rada, rešavač se nalazi u nekom stanju koje je definisano vrednostima određenih veličina kao što su, na primer, broj promenljivih kojima je dodeljena vrednost, broj konflikata od poslednjeg otpočinjanja iznova, broj naučenih klauza i slično. Tačan izbor veličina koje bi trebalo uključiti u definiciju stanja rešavača je jedan od važnijih istraživačkih izazova pošto značajno utiče na broj stanja, a samim tim i na efikasnost učenja uslovljavanjem koje ima eksponencijalnu složenost u zavisnosti od broja stanja. Stoga je potrebno identifikovati mali skup veličina i njihovih dopustivih vrednosti koje bi definisale stanja. Bilo bi prirodno i da stanja

predstavljaju ne pojedinačne vrednosti ovih veličina već njihove skupove konstruisane na osnovu nekog teorijskog predznanja. Tako bi se broj stanja mogao značajno smanjiti. Što se tiče dopustivih akcija, one generalno mogu da variraju od stanja do stanja i potrebno ih je preciznije formulisati, ali se mogu razmatrati dva nivoa apstraktnosti odluka. U apstraktnijem pristupu bi se za akcije mogle uzeti promene politika koje se smatraju pogodnim u različitim stanjima. Alternativni pristup koji bi bio teži za dizajniranje i implementaciju, ali sa većim potencijalom, bi bio da se za akcije uzmu konkretne akcije u okviru rešavača, kao što su otpočinjanje iznova u datom trenutku, zaboravljanje klauza i slično. Za učenje bi ponovo bile korišćene formule iz nekog reprezentativnog korpusa.

Planirani pristup upravljanju rešavača učenjem bi se mogao kombinovati sa metodologijom izloženom u ovom radu kako bi se u startu izabrali povoljniji parametri, ali bi jedan od bitnih zahteva novog pristupa morao biti da se rešavač u prihvatljivom vremenu prilagodi datoj formuli bez obzira na polazne parametre.

Sličan pristup je već primenjen u upravljanju tehnikom *iterirane lokalne pretrage* (eng. iterated local search) i prikazan u radu [36] sa ohrabrujućim rezultatima.

Kako je istraživanje na ovom polju u svojoj ranoj fazi, izvesno je da će se vremenom pojaviti i nove istraživačke teme, kako zasnovane na već izloženim idejama, tako i na potpuno novim.

Glava 8

Zaključci

U ovom radu je predložena jedna metodologija za prilagođavanje vrednosti parametara SAT rešavača. Iznete su i testirane hipoteze vezane za ponašanje i svojstva najboljih vrednosti parametara za različite formule. Urađena je evaluacija metodologije na velikim, relevantnim korpusima iskaznih formula koji uključuju kako instance problema od teorijskog, tako i od praktičnog značaja. Dobijeni rezultati su dobri i pokazuju da se metodologija može praktično primeniti, ali i da je dalje istraživanje u ovoj oblasti isplativo. Ispitane su sve hipoteze navedene u glavi 1 i pri tom se došlo do sledećih zaključaka:

- Pokazno je da važi prva hipoteza ovog rada navedena u glavi 1 — da među formulama iz iste familije postoji sintaksna sličnost koja se može automatski prepoznati i da se ona može iskoristiti za prepoznavanje familije kojoj formula pripada. Osnovu metodologije predstavljaju metode za klasifikovanje iskaznih formula. Iako je ceo pristup inspirisan zakonitostima u grafovskoj reprezentaciji formule i te zakonitosti se mogu automatski prepoznati, grafovski pristup se pokazao kao računski prilično zahtevan i zbog toga ograničene primenljivosti. Veoma jednostavan pristup zasnovan na tekstualnim n-gramima se pokazao značajno kvalitetnijim nego što je očekivano, ali samo u jednom svom slučaju — za $n = 1$. Pristup baziran na izabranom skupu sintaksnih svojstava formule se pokazao kao najbolji. Profili instanci se brzo izračunavaju, a kvalitet kalsifikovanja sa velikim brojem familija je odličan.

- Druga hipoteza — da za skupove formula koje su sintaksno slične postoje dominantno najbolje vrednosti parametara, je delimično potvrđena. Pokazalo se da među najboljim vrednostima parametrara za pojedinačne formule nema dominantnih na celoj familiji, međutim, obično se samo manji deo dopustivih kombinacija vrednosti parametara javlja među najboljim za formule jedne familije. Taj rezultat je ohrabrujući

za pretpostavku da bi se sa većim brojem parametara i većim brojem njihovih dopustivih vrednosti moglo postići ubedljivije izdvajanje jedne kombinacije ili makar značajno manjeg broja kombinacija vrednosti u okviru jedne familije formula.

- Pokazano je da postoji značajna korelacija između sintaksnih sličnosti među formulama i sličnosti među vrednostima parametara najboljim za njih što predstavlja upravo treću hipotezu ovog rada. Ranije nije izvedeno istraživanje koje bi ovakav zaključak potkrepilo. Ovaj rezultat je od velikog značaja jer pokazuje da su faktori koji utiču na postupak rešavanja iskaznih formula u značajnoj meri uslovljeni sintaksom formule. Ovakav zaključak naizgled zvuči očekivano, ali nestabilnosti u procesu rešavanja ga čine netrivijalnim. Primer ovakve nestabilnosti su razlike u vremenu rešavanja iskaznih formula koje se razlikuju samo u preimenovanju promenljivih i promeni rasporeda klauza. Ove razlike mogu biti i u nivou reda veličine.

- Potvrđeno je da važi osnovna hipoteza ovog rada — da se inteligentnim biranjem vrednosti parametara SAT rešavača, zasnovanim na analizi sintakse formule koja se rešava, može povećati njegova efikasnost. Na raznovrsnom korpusu formula, rešen je veći broj formula nego bez primene nove metodologije, a središnje vreme je smanjeno više nego dvostruko u odnosu na najbolje fiksirane vrednosti parametara. Kada se poboljšanje razmatra u odnosu na mogući pomak između najboljih fiksiranih i najboljih vrednosti iz dopustivog skupa vrednosti parametara, rezultati se mogu smatrati još boljim.

Važan pokazatelj o primenljivosti metodologije je što se poboljšanje performansi dobijeno njenim treniranjem na jednom korpusu prenosi i na kvalitativno različit korpus.

Pozitivni rezultati ovog rada otvaraju i pravce daljih istraživanja i ukazuju na mogućnosti daljeg poboljšanja SAT rešavača.

Literatura

[1] Stuart Bain. *Evolving Algorithms for Over-Constrained and Satisfaction Problems*. PhD thesis, School of Information and Communication Technology, Griffith University, 2006.

[2] Vincent D. Blondel, Anahí Gajardo, Maureen Heymans, Pierre Senellart, and Paul Van Dooren. A measure of similarity between graph vertices: Applications to synonym extraction and web searching. *SIAM Rev.*, 46(4):647–666, 2004.

[3] H. Bunke, X. Jiang, and A. Kandel. On the minimum common supergraph of two graphs. *Computing*, 65(1):13–25, 2000.

[4] Stephen A. Cook. The complexity of theorem-proving procedures. In *STOC '71: Proceedings of the third annual ACM symposium on Theory of computing*, pages 151–158. ACM, 1971.

[5] Martin Davis, George Logemann, and Donald Loveland. A machine program for theorem-proving. *Commun. ACM*, 5(7):394–397, 1962.

[6] Martin Davis and Hilary Putnam. A computing procedure for quantification theory. *J. ACM*, 7(3):201–215, 1960.

[7] Niklas Eén and Niklas Sörensson. An extensible sat-solver. In *Theory and Applications of Satisfiability Testing*, 2004.

[8] Ian P. Gent and Toby Walsh. The search for satisfaction. Technical report, University of Strathclyde, 1999.

[9] David E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley Longman Publishing Co., Inc., 1989.

[10] R. L. Graham. A similarity measure for graphs. *Los Alamos Science*, 1987.

[11] John H. Holland. *Adaptation in natural artificial systems*. University of Michigan Press, 1975.

[12] F. Hutter, H. H. Hoos, and T. Stützle. Automatic algorithm configuration based on local search. In *Proc. of the Twenty-Second Conference on Artifical Intelligence (AAAI '07)*, pages 1152–1157, 2007.

[13] Frank Hutter, Domagoj Babic, Holger H. Hoos, and Alan J. Hu. Boosting verification by automatic tuning of decision procedures. In *FMCAD '07: Proceedings of the Formal Methods in Computer Aided Design*, pages 27–34, Washington, DC, USA, 2007. IEEE Computer Society.

[14] Frank Hutter, Youssef Hamadi, Holger H. Hoos, and Kevin Leyton-Brown. Performance prediction and automated tuning of randomized and parametric algorithms. In *CP*, pages 213–228, 2006.

[15] Frank Hutter, Dave A. D. Tompkins, and Holger H. Hoos. Scaling and probabilistic smoothing: Efficient dynamic local search for sat. pages 233–248. Springer Verlag, 2002.

[16] Predrag Janičić. *Matematička logika u računarstvu.* Matematički fakultet Univerziteta u Beogradu, 2006.

[17] Toni Jussila, Armin Biere, Carsten Sinz, Daniel Krning, and Christoph M. Wintersteiger. A first step towards a unified proof checker for qbf. In *Proc. of SAT. To Appear*, 2007.

[18] N. Kashtan, S. Itzkovitz, R. Milo, and U. Alon. Efficient sampling algorithm for estimating subgraph concentrations and detecting network motifs. *Bioinformatics*, 20(11):1746–1758, 2004.

[19] N. Kashtan, S. Itzkovitz, R. Milo, and U. Alon. *mfinder Tool Guide.* Department of Molecluar Cell Biology and Computer Science & Applied Mathematics, Weizmann Institute of Science, 2005.

[20] Vlado Keselj, Fuchun Peng, Nick Cercone, and Calvin Thomas. N-gram-based author profiles for authorship attribution. In *Proceedings of the Conference Pacific Association for Computational Linguistics, PACLING'03*, pages 255–264, Dalhousie University, Halifax, Nova Scotia, Canada, August 2003.

[21] Michail G. Lagoudakis and Michael L. Littman. Learning to select branching rules in the dpll procedure for satisfiability. In *In LICS/SAT*, pages 344–359, 2001.

[22] Ken Lang. NewsWeeder: learning to filter netnews. In *Proceedings of the 12th International Conference on Machine Learning*, pages 331–339. Morgan Kaufmann publishers Inc.: San Mateo, CA, USA, 1995.

[23] Hans leo Teulings, Lambert R. B. Schomaker, Jan Gerritsen, Hans Drexler, and Marc Albers. An on-line handwriting-recognition system

based on unreliable modules. In *Computer Processing of Handwriting*, pages 167–185. World Scientific, 1990.

[24] Michael Luby, Alistair Sinclair, and David Zuckerman. Optimal speedup of las vegas algorithms. *Information Processing Letters*, 47:173–180, 1993.

[25] Alistair Manning, Andrew Ireland, and Alan Bundy. Incresing the versatility of heuristic based theorem provers. In *LPAR '93: Proceedings of the 4th International Conference on Logic Programming and Automated Reasoning*, pages 194–204, London, UK, 1993. Springer-Verlag.

[26] Filip Marić. *Formalization and Implementation of Modern SAT Solvers*. PhD thesis, Matematički fakultet Univerziteta u Beogradu, 2008.

[27] Filip Marić and Predrag Janičić. ARGO-LIB v3.5: System description for smtcomp'07. Technical report, Matematički fakultet Univerziteta u Beogradu, 2007.

[28] David Mcallester, Bart Selman, and Henry Kautz. Evidence for invariants in local search. In *In Proceedings of AAAI-97*, pages 321–326, 1997.

[29] Nenad Mladenovic and Pierre Hansen. Variable neighborhood search. *Computers & OR*, 24(11):1097–1100, 1997.

[30] Matthew W. Moskewicz, Conor F. Madigan, Ying Zhao, Lintao Zhang, and Sharad Malik. Chaff: Engineering an efficient SAT solver. In *Proceedings of the 38th Design Automation Conference (DAC'01)*, 2001.

[31] Eugene Nudelman, Kevin L. Brown, Holger H. Hoos, Alex Devkar, and Yoav Shoham. Understanding random sat: Beyond the clauses-to-variables ratio. In Mark Wallace, editor, *Principles and Practice of Constraint Programming - CP 2004, 10th International Conference, CP 2004, Toronto, Canada, September 27 - October 1, 2004, Proceedings*, volume 3258 of *Lecture Notes in Computer Science*, pages 438–452. Springer, 2004.

[32] Bart Selman, Henry A. Kautz, and Bram Cohen. Noise strategies for improving local search. pages 337–343. MIT press, 1994.

[33] Bart Selman, Hector Levesque, and David Mitchell. A new method for solving hard satisfiability problems. pages 440–446, 1992.

[34] Carsten Sinz. Visualizing sat instances and runs of the dpll algorithm. *J. Autom. Reasoning*, 39(2):219–243, 2007.

[35] Andrija Tomovic, Predrag Janicic, and Vlado Keselj. n-gram-based classification and unsupervised hierarchical clustering of genome sequences. *Computer Methods and Programs in Biomedicine*, 81(2):137–153, 2006.

[36] Klaus E. Varrentrapp. *A Practical Framework for Adaptive Metaheuristics*. PhD thesis, Technical University Darmstadt, 2005.

[37] Miodrag Živković. *Algoritmi*. Matematički fakultet Univerziteta u Beogradu, 2000.

[38] Lin Xu, Frank Hutter, Holger H. Hoos, and Kevin Leyton-Brown. SATzilla: portfolio-based algorithm selection for SAT. *Journal of Artificial Intelligence Research*, 32:565–606, June 2008.

[39] Laura Zager. Graph similarity and matching. Master's thesis, Massachusetts Institute of Technology, 2005.