

UNIVERZITET U BEOGRADU
MATEMATIČKI FAKULTET



Ivan Drecun

ALGORITMI ZA ISPITIVANJE IZOMORFIZMA
GRAFOVA

master rad

Beograd, 2021.

Mentor:

dr Filip MARIĆ, vanredni profesor
Univerzitet u Beogradu, Matematički fakultet

Članovi komisije:

dr Miodrag ŽIVKOVIĆ, redovan profesor
Univerzitet u Beogradu, Matematički fakultet

dr Vesna MARNIKOVIĆ, docent
Univerzitet u Beogradu, Matematički fakultet

Datum odbrane: _____

Naslov master rada: Algoritmi za ispitivanje izomorfizma grafova

Rezime: U ovom radu opisan je i realizovan algoritam za rešavanje problema izomorfizma grafova **GI**. Dat je formalan opis opšteg okvira za konstrukciju algoritma za rešavanje **GI** prikazan u radu [7]. Konstruisan je algoritam zasnovan na proceduri profinjavanja bojenja grafa i uvedeno je jedno uopštenje te procedure. Algoritam je implementiran u programskom jeziku C++ u okviru programa *morphi* čije je vreme izvršavanja testirano na raznovrsnom skupu grafova i napravljeno je poređenje rezultata sa rezultatima nekih najuspešnijih rešavača za problem izomorfizma grafova.

Ključne reči: izomorfizam grafova, kanonska forma grafa, grupa automorfizama, obojen graf, profinjavanje bojenja

Sadržaj

1	Uvod	1
2	Opšti algoritam	3
2.1	Osnovni pojmovi	3
2.2	Stablo pretrage	9
2.3	Invarijanta stabla	11
2.4	Odsecanje pretrage	15
3	Realizacija algoritma	18
3.1	Reprezentacija podataka	18
3.2	Funkcija odabira ciljne ćelije	20
3.3	Funkcija profinjavanja	21
3.4	Invarijanta stabla	29
3.5	Automorfizmi	33
3.6	Pretraga	34
3.7	Invarijanta grafa	38
4	Rezultati	40
4.1	Detalji testiranja	40
4.2	Nasumično generisani grafovi	41
4.3	Grafovi donje granice nekih algoritama	43
4.4	Mreže	45
4.5	Grafovi nekih kombinatornih objekata	46
5	Zaključak i dalji rad	49
	Bibliografija	51

Glava 1

Uvod

Dva grafa su izomorfna ukoliko su jednaki do na promenu imenovanja čvorova. Problem ispitivanja izomorfizma grafova **GI** je problem određivanja postojanja takvog preimenovanja. Trenutno nije poznat algoritam polinomijalne vremenske složenosti koji rešava ovaj problem u opštem slučaju [1]. Ipak, sa praktičnog stanovišta bi se moglo kazati da je problem rešen. Najuspešniji rešavači za **GI** (nauty [6, 7], Traces [11, 7], bliss [4], conauto [5], saucy [3]) sposobni su da reše velike instance teških klasa grafova u realnom vremenu. Pritom, sastavljanje teških klasa grafova za ove rešavače nije trivijalno i u realnoj upotrebi se takvi grafovi veoma retko javljaju. Na grafovima bez konkretne kombinatorne strukture se ti algoritmi čak ponašaju linearno [10].

Postoje dva pristupa praktičnom rešavanju problema izomorfizma grafova. Prvi, neposredniji pristup je direktna pretraga izomorfizma između dva zadata grafa. Algoritmi zasnovani na ovom pristupu imaju korisnu osobinu da je moguće prekinuti pretragu pre ispitivanja velikog dela prostora pretrage, na primer kada je izomorfizam pronađen ili kada je pronađeno neko svojstvo grafova iz kog je moguće zaključiti njihov neizomorfizam. Drugi pristup oslanja se na mogućnost svođenja grafa na neki kanonski oblik. Time je omogućena jednostavna provera izomorfizma dva grafa - ispitivanjem jednakosti tako dobijenih kanonskih grafova. Ovakav pristup je znatno pogodniji od prvog u situacijama kada je neophodno efikasno ispitivanje izomorfizma u radu sa većim brojem grafova, na primer ukoliko je potrebno odrediti sve parove izomorfnih grafova u nekoj kolekciji grafova. Uprkos tome što algoritmi za određivanje kanonske forme grafa ne poseduju mogućnost ranijeg zaustavljanja pretrage, neki od najuspešnijih modernih alata za rešavanje **GI** zasnovani su upravo na tom pristupu.

Centralna tema ovog rada je Mekejev algoritam za određivanje kanonske forme. Pojmovi i tvrđenja uvedeni su u skladu sa opisom opšteg okvira za konstrukciju algoritma prikazanim u radu [7] na osnovu kog je zatim konstruisan konkretan algoritam. Uvedeno je i originalno uopštenje algoritma i prikazani su rezultati vremena izvršavanja implementiranog algoritma sa i bez tog uopštenja.

Glava 2

Opšti algoritam

U ovoj glavi predstavljeni su osnovni matematički pojmovi neophodni za dalje razumevanje konstrukcije opšteg algoritma za određivanje kanonske forme grafa. Uvedeni su pojmovi *bojenja* i *obojenog grafa*, nakon čega je prikazana konstrukcija stabla pretrage koja leži u osnovi algoritma. Nad stablom je pogodno definisana invarijanta koja omogućava određivanje grupe automorfizama grafa i kanonske forme. Na kraju su prikazani i mehanizmi odsecanja pretrage koji omogućavaju praktično izvršavanje algoritma u razumnom vremenu.

2.1 Osnovni pojmovi

Permutacija i particija

Permutacija konačnog skupa A je bilo koja bijekcija $g : A \rightarrow A$. Skup svih takvih permutacija označavamo sa S_A . Specijalno, ako je $A = \{1, 2, \dots, n\}$ za neki prirodan broj n , tada koristimo oznaku S_n . Zajedno sa operacijom kompozicije preslikavanja skup S_n je grupa koju zovemo *simetrična grupa stepena n* . Identičnu permutaciju označavamo sa id , a inverz permutacije g sa g^{-1} . Permutaciju g možemo zapisati kao preslikavanje, npr. $g : \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 3 & 4 & 1 & 5 & 2 \end{pmatrix}$ ili kao kompoziciju ciklusa $(1\ 3)(2\ 4\ 5)$.

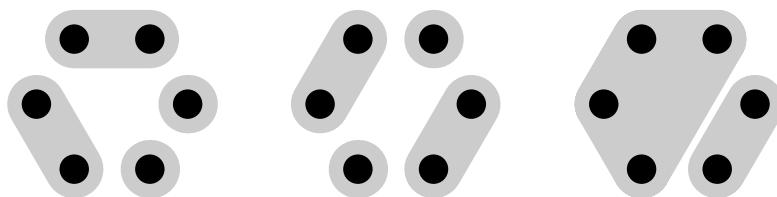
Particija konačnog skupa A je skup nepraznih podskupova od A takav da se svaki element skupa A nalazi u tačno jednom od tih podskupova. Te podskupove zovemo *klasama* ili *čelijama* particije. Relacija \approx na skupu A je *relacija ekvivalencije* ukoliko je refleksivna, simetrična i tranzitivna. *Klasa ekvivalencije* elementa $a \in A$ je skup $C_a = \{b \in A \mid a \approx b\}$. Skup klasa ekvivalencije relacije \approx čini jednu

particiju $P = A/\approx$ skupa A . Slično, svaka particija P je skup klasa ekvivalencije jedne relacije ekvivalencije \approx_P . Zbog ovoga u daljem tekstu poistovećujemo pojmove particije skupa A i relacije ekvivalencije na skupu A .

Neka su α i β particije skupa A . Particija α je *finija* od particije β (u oznaci $\alpha \leq \beta$) ukoliko za sve $a, b \in A$ važi $a \approx_\alpha b \implies a \approx_\beta b$. Ovaj uslov je ekvivalentan tome da je svaka klasa particije α podskup neke klase particije β .

Neka su α i β particije skupa A . Particija $\alpha \vee \beta$ definisana kao tranzitivno zatvorenje relacije $a \approx b \iff a \approx_\alpha b \vee a \approx_\beta b$, odnosno $a \approx_{\alpha \vee \beta} b \iff x_1 \approx x_2 \approx \dots \approx x_k$ za neke $a = x_1, x_2, \dots, x_k = b$ naziva se *supremum particija α i β* . Primetimo da su particije α i β finije od $\alpha \vee \beta$, kao i da je $\alpha \vee \beta$ finije od svakog γ za koje važi $\alpha, \beta \leq \gamma$, odnosno da je $\alpha \vee \beta$ zaista njihovo najmanje gornje ograničenje.

Primer 1. Na slici 2.1 prikazane su dve particije skupa od šest elemenata, kao i njihov supremum. Sa slike se vidi i da su obe particije finije od supremuma.



Slika 2.1: Particije (levo) i supremum (desno).

Obojen graf

Bojenje skupa A je surjektivna funkcija $\pi : A \rightarrow \{1, 2, \dots, k\}$ za neki prirodan broj $k > 0$. Označimo skup svih bojenja skupa sa n elemenata sa Π_n (nadalje Π). Broj k zovemo *brojem boja* i označavamo ga sa $|\pi|$. *Ćelija bojenja π* boje c je skup svih čvorova te boje, odnosno $\pi^{-1}(c)$ za $c \in \{1, 2, \dots, k\}$. Bojenje je *diskretno* ukoliko je $|\pi| = n$ i tada je π permutacija skupa A .

Označimo sa \sim_π binarnu relaciju na skupu A definisanu sa $a \sim_\pi b \iff \pi(a) = \pi(b)$. U pitanju je jedinstvena particija čije klase odgovaraju upravo ćelijama bojenja π i nazivamo je *particija bojenja π* . Primetimo da nije moguće poistovetiti bojenje i particiju zato što za proizvoljnu particiju bojenje čije ćelije odgovaraju klasama particije nije jedinstveno - postoji ih onoliko koliko je različitih rasporeda klasa. Bojenje možemo poistovetiti sa *uređenom particijom*, odnosno particijom čije su klase uređene u niz.

Bojenje π_1 je *finije* od bojenja π_2 (u oznaci $\pi_1 \leq \pi_2$) ukoliko za sve $a, b \in A$ važi implikacija $\pi_2(a) < \pi_2(b) \implies \pi_1(a) < \pi_1(b)$. Primetimo da za bojenja π_1 i π_2 važi $\pi_1 \leq \pi_2 \implies \sim_{\pi_1} \leq \sim_{\pi_2}$, ali ne i obrnuto.

Primer 2. Neka je $\pi : \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 3 & 1 & 2 & 2 & 1 & 2 \end{pmatrix}$ jedno bojenje. Koristan je zapis po ćelijama bojenja $\pi = [2 \ 5 \mid 4 \ 3 \ 6 \mid 1]$, pri čemu redosled brojeva unutar ćelija nije bitan. Bojenje $\pi_1 = [5 \mid 2 \mid 4 \ 6 \mid 3 \mid 1]$ je finije od π . Bojenje $\pi_2 = [2 \mid 3 \mid 4 \ 6 \mid 5 \mid 1]$ nije finije od π iako važi da je svaka njegova ćelija podskup neke ćelije bojenja π . To je moguće videti iz toga što se bojenje π može dobiti spajanjem susednih ćelija bojenja π_1 , ali ne i spajanjem susednih ćelija bojenja π_2 .

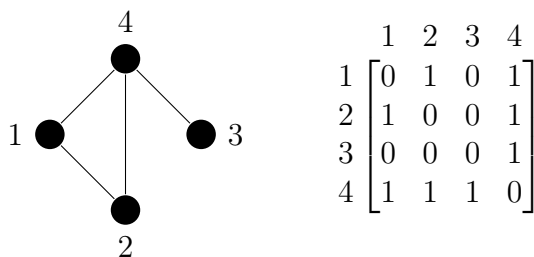
Primer 3. Neka je $\pi = [4 \mid 1 \mid 3 \mid 2]$ diskretno bojenje. Odgovarajuća permutacija je $\pi : \begin{pmatrix} 1 & 2 & 3 & 4 \\ 2 & 4 & 3 & 1 \end{pmatrix}$, a njen inverz je $\pi^{-1} : \begin{pmatrix} 1 & 2 & 3 & 4 \\ 4 & 1 & 3 & 2 \end{pmatrix}$.

Graf $G = (V, E)$ je uređeni par konačnog skupa čvorova V i skupa grana $E \subseteq V \times V$ pri čemu za sve $u, v \in V$ važi $(v, v) \notin E$ i ako je $(u, v) \in E$ onda je $(v, u) \in E$. U nastavku pretpostavljamo da je $V = \{1, 2, \dots, n\}$ za neki prirodan broj $n > 0$. Označimo skup svih grafova sa n čvorova sa \mathcal{G}_n (nadalje \mathcal{G}).

Matrica povezanosti grafa je preslikavanje $\psi : \mathcal{G} \times V \times V \rightarrow \{0, 1\}$ takvo da važi $\psi(G, u, v) = 1 \iff (u, v) \in E$. Binarna reprezentacija $\text{bin}(G)$ grafa G je binarni niz dobijen nadovezivanjem redova gornjeg trougla matrice povezanosti grafa G .

Obojen graf je uređeni par (G, π) gde je π jedno bojenje skupa čvorova V grafa G . Kažemo i da je π bojenje grafa G .

Primer 4. Na slici 2.2 je prikazan jedan graf G zajedno sa njegovom matricom povezanosti. Njegova binarna reprezentacija je $\text{bin}(G) = 101011$.

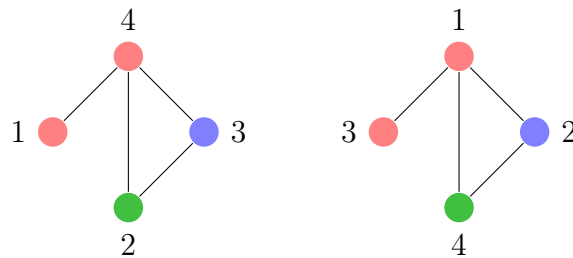


Slika 2.2: Graf i matrica povezanosti.

Izomorfizam

Obojeni grafovi (G_1, π_1) i (G_2, π_2) su *izomorfni* (u oznaci $(G_1, \pi_1) \cong (G_2, \pi_2)$) ukoliko postoji $g \in S_n$ tako da je $(G_1, \pi_1) = (G_2, \pi_2)^g$. Takvo g zovemo *izomorfizam*.

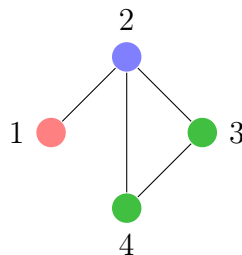
Primer 6. Neka su dati grafovi (G_1, π_1) i (G_2, π_2) kao na slici 2.4. Ako je redosled boja crvena, zelena, plava, tada je $\pi_1 = [1\ 4\ |\ 2\ |\ 3]$ i $\pi_2 = [3\ 1\ |\ 4\ |\ 2]$. Ovi grafovi su izomorfni za izomorfizam $g : \begin{pmatrix} 1 & 2 & 3 & 4 \\ 3 & 4 & 2 & 1 \end{pmatrix}$.



Slika 2.4: Izomorfni grafovi (G_1, π_1) i (G_2, π_2) .

Automorfizam obojenog grafa (G, π) je izomorfizam tog grafa sa samim sobom, odnosno $g \in S_n$ za koje važi $(G, \pi) = (G, \pi)^g$. Skup automorfizama grafa (G, π) označavamo sa $Aut(G, \pi)$. Zajedno sa operacijom kompozicije preslikavanja skup $Aut(G, \pi)$ čini grupu automorfizama.

Primer 7. Na slici 2.5 prikazan je graf sa jednim automorfizmom.



Slika 2.5: Graf sa automorfizmom $g : \begin{pmatrix} 1 & 2 & 3 & 4 \\ 1 & 2 & 4 & 3 \end{pmatrix}$.

Kanonska forma

Neka je $f : \mathcal{G} \times \Pi \rightarrow S$ preslikavanje iz skupa svih obojenih grafova u proizvoljan skup S . Kažemo da je f funkcija invarijantna na imenovanje čvorova ukoliko za svaki obojen graf (G, π) i svaku permutaciju $g \in S_n$ važi $f(G^g, \pi^g) = f(G, \pi)$. Neformalno, to znači da vrednost funkcije f ne zavisi od konkretnog imenovanja čvorova grafa, već samo od njegove unutrašnje strukture.

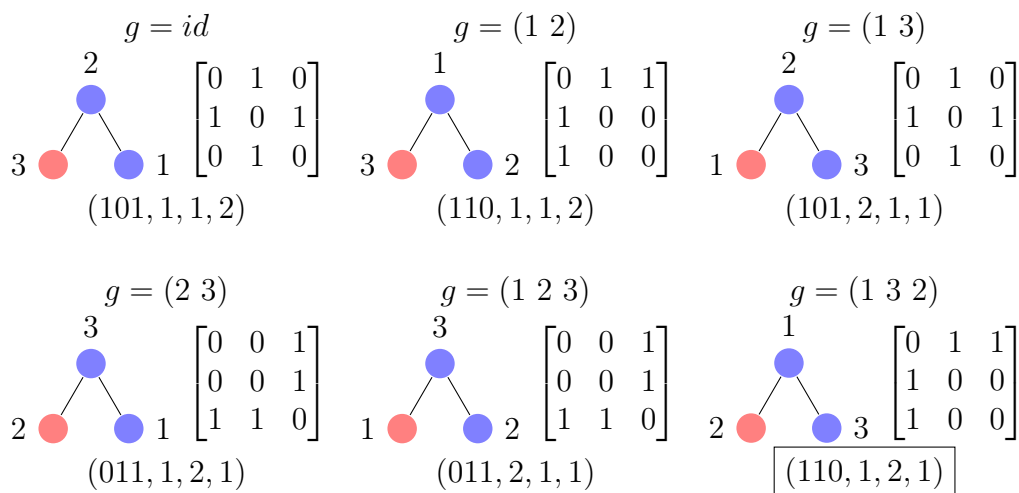
Ako na skupu S postoji definisano dejstvo grupe S_n , kažemo da je f transformacija invarijantna na imenovanje čvorova ukoliko za svaki obojen graf (G, π) i svaku permutaciju $g \in S_n$ važi $f(G^g, \pi^g) = f(G, \pi)^g$.

Definicija. Kanonska forma je preslikavanje $\mathcal{C} : \mathcal{G} \times \Pi \rightarrow \mathcal{G} \times \Pi$ koje ispunjava sledeće uslove:

(C1) Za svaki obojen graf (G, π) važi $\mathcal{C}(G, \pi) \cong (G, \pi)$

(C2) \mathcal{C} je funkcija invarijantna na imenovanje čvorova

Primer 8. Neka je definisano uređenje na skupu obojenih grafova leksikografskim poređenjem odgovarajućih nizova $(bin(G), \pi(1), \dots, \pi(n))$. Tada je preslikavanje definisano sa $\mathcal{M}(G, \pi) = \max_{g \in S_n} (G^g, \pi^g)$ kanonska forma. Na slici 2.6 prikazane su sve permutacije jednog obojenog grafa i istaknut je maksimum u prethodno opisanom smislu.



Slika 2.6: Sve permutacije jednog obojenog grafa.

2.2 Stablo pretrage

Označimo sa V^* skup svih konačnih nizova elemenata skupa V . Ako je $\nu \in V^*$ sa $|\nu|$ označavamo dužinu niza ν . Ako je $\nu = (v_1, v_2, \dots, v_k) \in V^*$ i $w \in V$, onda $\nu \| w$ označava niz $(v_1, v_2, \dots, v_k, w)$. Za $0 \leq s \leq k$ prefiks niza ν dužine s označavamo sa $[\nu]_s = (v_1, v_2, \dots, v_s)$. Uređenje \leq na skupu V^* predstavlja leksikografski poredak.

Čvorovi stabla pretrage predstavljeni su nizovima elemenata skupa V , pri čemu korenu stabla odgovara prazan niz. U nastavku definišemo funkcije na osnovu kojih ćemo opisati pravila grananja u stablu.

Definicija. *Funkcija profinjavanja* je bilo koje preslikavanje $R : \mathcal{G} \times \Pi \times V^* \rightarrow \Pi$ koje za svaki obojen graf (G, π) i svako $\nu \in V^*$ zadovoljava sledeće uslove:

$$(R1) \quad R(G, \pi, \nu) \leq \pi$$

$$(R2) \quad \text{Ako je } v \in \nu, \text{ onda je } \{v\} \text{ ćelija bojenja } R(G, \pi, \nu)$$

$$(R3) \quad \text{Za svako } g \in S_n \text{ važi } R(G^g, \pi^g, \nu^g) = R(G, \pi, \nu)^g$$

Definicija. *Funkcija odabira ciljne ćelije* je bilo koje preslikavanje $T : \mathcal{G} \times \Pi \times V^* \rightarrow \mathcal{P}(V)$ koje za svaki obojen graf (G, π) i svako $\nu \in V^*$ zadovoljava sledeće uslove:

$$(T1) \quad \text{Ako je } R(G, \pi, \nu) \text{ diskretno, onda je } T(G, \pi, \nu) = \emptyset$$

$$(T2) \quad \text{Ako } R(G, \pi, \nu) \text{ nije diskretno, onda je } T(G, \pi, \nu) \text{ višečlana ćelija od } R(G, \pi, \nu)$$

$$(T3) \quad \text{Za svako } g \in S_n \text{ važi } T(G^g, \pi^g, \nu^g) = T(G, \pi, \nu)^g$$

Kako je graf fiksna, ove funkcije možemo smatrati funkcijama čvorova stabla. Funkcija profinjavanja obezbeđuje postojanje bojenja pridruženog svakom čvoru stabla ν , koje zbog uslova (R2) razdvaja sve čvorove grafa niza ν u zasebne ćelije. Funkcija odabira ciljne ćelije nam omogućava da odaberemo skup čvorova grafa koji nam služi za konstrukciju dece tog čvora u stablu. Treći uslov u obe definicije govori da su u pitanju transformacije invarijantne na imenovanje čvorova.

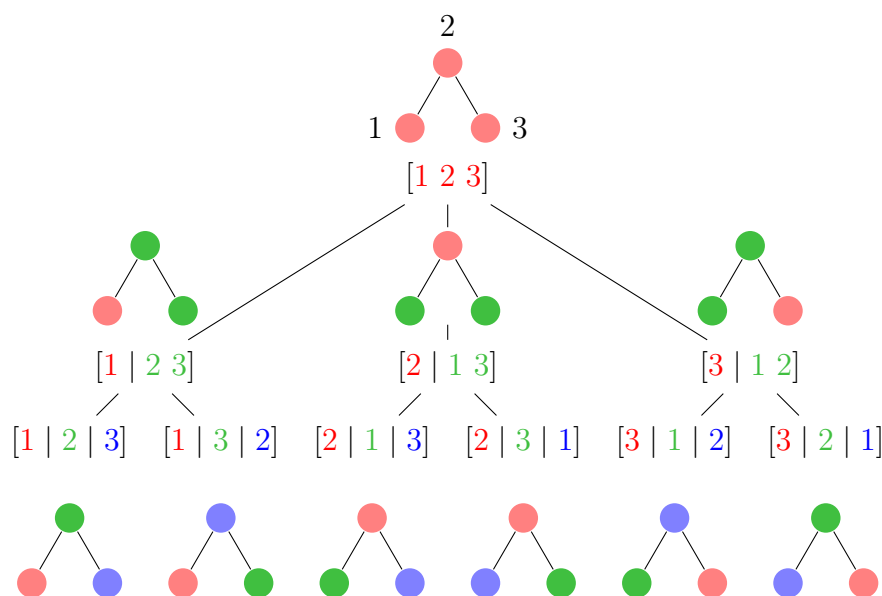
Definicija. *Stablo pretrage* $\mathcal{T}(G, \pi)$ određeno je sledećim uslovima:

$$(\mathcal{T}1) \quad \text{Koren stabla } \mathcal{T}(G, \pi) \text{ je prazan niz } ()$$

$$(\mathcal{T}2) \quad \text{Ako je } \nu \text{ čvor stabla } \mathcal{T}(G, \pi), \text{ njegova deca u stablu su } \{\nu \| w \mid w \in T(G, \pi, \nu)\}$$

Iz definicije se vidi da je neki čvor list stabla ako i samo ako je njemu pridruženo bojenje diskretno. Primetimo da je ovako definisano stablo konačno zato što za svako ν koje je permutacija skupa V važi da je bojenje pridruženo nekom njegovom prefiksu diskretno ili je njemu pridruženo bojenje diskretno.

Primer 9. Prikažimo jedno jednostavno stablo pretrage. Za funkciju odabira cilj-
ne ćelije možemo uzeti funkciju koja bira prvu višočlanu ćeliju bojenja. Za funkciju
profinjavanja možemo uzeti funkciju koja dodeljuje boje čvorovima u skladu sa po-
četnim bojenjem π , pri čemu čvorove niza ν izdvaja u zasebne ćelije. Na slici 2.7 je
prikazano stablo pretrage jednog grafa.



Slika 2.7: Stablo pretrage koje generiše sve permutacije skupa čvorova.

Dejstvo grupe S_n na stablo definiše se slično kao za bilo koju drugu strukturu. Naredna lema pokazuje da je ovako definisano stablo invarijantno na imenovanje čvorova grafa.

Lema 1. Za svaki obojen graf (G, π) i svako $g \in S_n$ važi $\mathcal{T}(G^g, \pi^g) = \mathcal{T}(G, \pi)^g$.

Dokaz. Dokažimo da za svaki čvor ν stabla $\mathcal{T}(G, \pi)$ važi da je ν^g čvor stabla $\mathcal{T}(G^g, \pi^g)$. Dokaz izvodimo indukcijom po strukturi stabla.

Baza indukcije Prazan niz je koren stabla $\mathcal{T}(G^g, \pi^g)$, pa tvrđenje trivijalno važi.

Induktivni korak Pretpostavimo da tvrđenje važi za čvor ν . Neka je $\nu||w$ dete čvora ν za neko $w \in T(G, \pi, \nu)$. Tada je $(\nu||w)^g = \nu^g||w^g$, ali kako važi $w^g \in T(G, \pi, \nu^g) \stackrel{(T3)}{=} T(G^g, \pi^g, \nu^g)$ to je $\nu^g||w^g$ dete čvora ν^g u stablu $\mathcal{T}(G^g, \pi^g)$.

Time smo dokazali da je stablo $\mathcal{T}(G, \pi)^g$ podstablo od $\mathcal{T}(G^g, \pi^g)$ ($\mathcal{T}(G, \pi)^g \subseteq \mathcal{T}(G^g, \pi^g)$). Prema prethodno dokazanom važi $\mathcal{T}(G^g, \pi^g)^{g^{-1}} \subseteq \mathcal{T}((G^g)^{g^{-1}}, (\pi^g)^{g^{-1}}) = \mathcal{T}(G, \pi)$, pa primenom g na obe strane konačno dobijamo $\mathcal{T}(G^g, \pi^g) \subseteq \mathcal{T}(G, \pi)^g$. \square

Posledica 1. Neka je ν čvor stabla $\mathcal{T}(G, \pi)$ i neka $\mathcal{T}(G, \pi, \nu)$ označava njegovo podstablo sa korenom u ν . Ako je $g \in \text{Aut}(G, \pi)$, onda je ν^g čvor stabla $\mathcal{T}(G, \pi)$ i važi $\mathcal{T}(G, \pi, \nu^g) = \mathcal{T}(G, \pi, \nu)^g$.

Lema 2. Neka je ν čvor stabla $\mathcal{T}(G, \pi)$ i $\pi_\nu = R(G, \pi, \nu)$. Tada je grupa automorfizama obojenog grafa (G, π_ν) jednaka stabilizatoru niza ν u grupi automorfizama grafa (G, π) , odnosno $\text{Aut}(G, \pi_\nu) = \Sigma_\nu^{\text{Aut}(G, \pi)}$.

Dokaz. Neka je $g \in \text{Aut}(G, \pi_\nu)$. Tada je $\pi_\nu^g = \pi_\nu$, pa za svako v važi $\pi_\nu^g(v) = \pi_\nu(v)$. Na osnovu uslova (R2) za $v \in \nu$ važi $\{v\} = \pi_\nu^{-1}(\pi_\nu(v)) = \pi_\nu^{-1}(\pi_\nu^g(v)) = \pi_\nu^{-1}(\pi_\nu(g^{-1}v)) \ni g^{-1}v$ odakle sledi $v^g = v$, odnosno g stabilizuje svako $v \in \nu$.

Sa druge strane, neka $g \in \text{Aut}(G, \pi)$ stabilizuje ν . Tada po (R3) važi $\pi_\nu^g = R(G, \pi, \nu)^g = R(G^g, \pi^g, \nu^g) = R(G, \pi, \nu) = \pi_\nu$, pa je $g \in \text{Aut}(G, \pi_\nu)$. \square

2.3 Invarijanta stabla

Definicija. *Invarijanta stabla* je preslikavanje $\phi : \mathcal{G} \times \Pi \times V^* \rightarrow F$ za neki potpuno uređen skup F koje za sve obojene grafove (G, π) i različite čvorove $\nu_1, \nu_2 \in \mathcal{T}(G, \pi)$ ispunjava sledeće uslove:

($\phi 1$) Ako su $\nu_1, \nu_2 \in \mathcal{T}(G, \pi)$ takvi da je $|\nu_1| = |\nu_2|$ i $\phi(G, \pi, \nu_1) < \phi(G, \pi, \nu_2)$, onda za sve $\omega_1 \in \mathcal{T}(G, \pi, \nu_1)$ i $\omega_2 \in \mathcal{T}(G, \pi, \nu_2)$ važi $\phi(G, \pi, \omega_1) < \phi(G, \pi, \omega_2)$

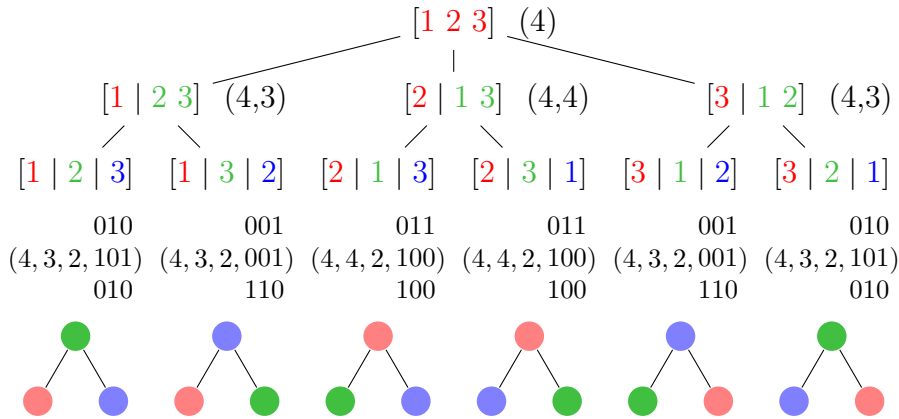
($\phi 2$) Ako su $\nu_1, \nu_2 \in \mathcal{T}(G, \pi)$ takvi da su $\pi_1 = R(G, \pi, \nu_1)$ i $\pi_2 = R(G, \pi, \nu_2)$ diskretna bojenja, onda je $\phi(G, \pi, \nu_1) = \phi(G, \pi, \nu_2) \implies G^{\pi_1} = G^{\pi_2}$

($\phi 3$) ϕ je funkcija invarijantna na imenovanje čvorova grafa

Listovi ν_1 i ν_2 su *ekvivalentni* ako i samo ako $\phi(G, \pi, \nu_1) = \phi(G, \pi, \nu_2)$.

Neformalno, prvi uslov obezbeđuje da ako je vrednost invarijante u čvoru ν_1 manja od vrednosti invarijante u čvoru ν_2 istog nivoa, tada to važi i za sve parove čvorova njihovih podstabala. Drugi uslov garantuje da su grafovi dobijeni transformacijom polaznog grafa primenom bojenja u ekvivalentnim listovima jednaki.

Primer 10. Definišimo $\phi(G, \pi, \nu)$ koristeći pomoćnu funkciju f kao niz vrednosti $(f(G, \pi, [\nu]_0), f(G, \pi, [\nu]_1), \dots, f(G, \pi, [\nu]_{|\nu|}))$, pri čemu u slučaju da je ν list možemo na taj niz nadovezati i binarnu reprezentaciju grafa permutovanog bojenjem $R(G, \pi, \nu)$. Ovako definisano ϕ ispunjava uslove $(\phi 1-2)$. Ako za pomoćnu funkciju f odaberemo npr. proizvod zbirova stepena čvorova ćelija, dobijamo jednu jednostavnu invarijantu stabla (npr. za bojenje $[2 \mid 1 \ 3]$ zbir stepena čvorova prve ćelije je 2, a druge je $1 + 1$, pa je vrednost funkcije $2(1 + 1) = 4$ za ovo bojenje). Vrednosti invarijante stabla iz primera 2.7 prikazane su na slici 2.8.



Slika 2.8: Invarijanta stabla.

U nastavku ćemo kroz dve teoreme prikazati značaj ovako definisane invarijante stabla. Označimo za proizvoljan čvor stabla ν njegovo bojenje dobijeno funkcijom profinjavanja sa $\pi_\nu = R(G, \pi, \nu)$.

Lema 3. Neka je $g \in \text{Aut}(G, \pi)$ i listovi ν_1 i ν_2 takvi da je $\nu_1^g = \nu_2$. Tada su ν_1 i ν_2 ekvivalentni i $g = \pi_{\nu_2}^{-1} \pi_{\nu_1}$.

Dokaz. Na osnovu svojstva $(\phi 3)$ invarijante stabla i činjenice da je g automorfizam sledi $\phi(G, \pi, \nu_1) =_{(\phi 3)} \phi(G^g, \pi^g, \nu_1^g) =_{g \in \text{Aut}(G, \pi)} \phi(G, \pi, \nu_1^g) = \phi(G, \pi, \nu_2)$, odnosno ν_1 i ν_2 su ekvivalentni. Dalje, važi $\pi_{\nu_2}^{-1} \pi_{\nu_1} = \pi_{\nu_1^g}^{-1} \pi_{\nu_1} =_{(R3)} (\pi_{\nu_1}^g)^{-1} \pi_{\nu_1} = (\pi_{\nu_1} g^{-1})^{-1} \pi_{\nu_1} = g \pi_{\nu_1}^{-1} \pi_{\nu_1} = g$. \square

Lema 4. Neka su α i β diskretna bojenja finija od bojenja π . Tada je $\pi^\alpha = \pi^\beta$.

Dokaz. Dokaz izvodimo nizom pomoćnih tvrđenja.

1. $id \leq \pi^\alpha$

Neka su x i y proizvoljne vrednosti. Važi $\pi^\alpha(x) < \pi^\alpha(y) \iff \pi(\alpha^{-1}(x)) < \pi(\alpha^{-1}(y))$, pa kako je $\alpha \leq \pi$ sledi $\alpha(\alpha^{-1}(x)) < \alpha(\alpha^{-1}(y))$ odnosno $x < y$. Kontrapozicijom dobijamo i $x \leq y \implies \pi^\alpha(x) \leq \pi^\alpha(y)$.

2. $(\pi^\alpha)^{-1}(c) = [n, m]$ za neko $n, m \in \mathbb{N}$ gde je $[n, m] = \{k \in \mathbb{N} \mid n \leq k \leq m\}$

Za svako x, y, z važi $x \leq y \leq z \implies \pi^\alpha(x) \leq \pi^\alpha(y) \leq \pi^\alpha(z)$, pa ako je $\pi^\alpha(x) = \pi^\alpha(z) = c$, onda je i $\pi^\alpha(y) = c$.

3. $\pi^\alpha(n+1) = \pi^\alpha(n)$ ili $\pi^\alpha(n+1) = \pi^\alpha(n) + 1$

Neka je $\pi^\alpha(n+1) \neq \pi^\alpha(n)$. Tada je $n+1 \geq n \implies \pi^\alpha(n+1) \geq \pi^\alpha(n)$, pa je $\pi^\alpha(n+1) > \pi^\alpha(n)$ jer su po pretpostavci različiti. Pretpostavimo da je $\pi^\alpha(n+1) > \pi^\alpha(n) + 1$. Tada postoji m takvo da je $\pi^\alpha(m) = \pi^\alpha(n) + 1$, ali iz $\pi^\alpha(n) < \pi^\alpha(m) < \pi^\alpha(n+1) \implies n < m < n+1$ sledi kontradikcija.

4. $|(\pi^\alpha)^{-1}(c)| = |\pi^{-1}(c)|$

Važi da je $(\pi^\alpha)^{-1}(c) = \pi^{-1}(c)^\alpha$ odakle je $|(\pi^\alpha)^{-1}(c)| = |\pi^{-1}(c)^\alpha| = |\pi^{-1}(c)|$.

5. $(\pi^\alpha)^{-1}(c) = (\pi^\beta)^{-1}(c)$

Dokaz izvodimo indukcijom po c .

Baza indukcije Kako je $1 \leq x$ za sve x , onda iz $id \leq \pi^\alpha$ sledi $\pi^\alpha(1) \leq \pi^\alpha(x)$ za sve x pa je $\pi^\alpha(1) = 1$. Dalje, kako je $|(\pi^\alpha)^{-1}(1)| = |\pi^{-1}(1)|$, mora važiti $(\pi^\alpha)^{-1}(1) = [1, |\pi^{-1}(1)|]$. Analogno se pokazuje i za β .

Induktivni korak Neka je po induktivnoj pretpostavci $(\pi^\alpha)^{-1}(c) = (\pi^\beta)^{-1}(c) = [n, m]$. Tada je $\pi^\alpha(m+1) \neq \pi^\alpha(m)$, pa je $\pi^\alpha(m+1) = \pi^\alpha(m) + 1$ i $m+1$ je najmanje u $(\pi^\alpha)^{-1}(c+1)$. Odatle važi $(\pi^\alpha)^{-1}(c+1) = [m+1, m+|\pi^{-1}(c+1)|]$. Analogno se pokazuje i za β .

□

Teorema 1. Za svaki list ν važi $Aut(G, \pi) = \{\pi_\omega^{-1}\pi_\nu \mid \nu \text{ i } \omega \text{ su ekvivalentni}\}$.

Dokaz. Neka je $g \in \text{Aut}(G, \pi)$. Tada je po posledici 1 ν^g list stabla $\mathcal{T}(G, \pi)$. Po prethodno dokazanoj lemi 3 su ν i ν^g ekvivalentni i $g = \pi_{\nu^g}^{-1} \pi_\nu$ što je element skupa sa desne strane jednakosti. Sa druge strane, ako su ν i ω ekvivalentni, onda je $G^{\pi_\nu} = G^{\pi_\omega}$, pa je $\pi_\omega^{-1} \pi_\nu \in \text{Aut}(G, \pi)$. \square

Prethodna teorema pokazuje da je otkrivanjem svih listova ekvivalentnih jednom listu moguće odrediti grupu automorfizama datog grafa. Naravno, ovakav način određivanja grupe automorfizama nije veoma efikasan pošto se grupa generiše član po član. Ovo se može poboljšati odsecanjem pretrage o čemu će biti reči u narednom odeljku.

Primer 11. *Posmatrajmo prvi list ζ stabla sa slike 2.8. Na osnovu prethodne teoreme je moguće odrediti skup svih automorfizama grafa. Kako je list ζ ekvivalentan samom sebi, $id \in \text{Aut}(G, \pi)$. Kako je ekvivalentan poslednjem listu ν stabla, $\pi_\nu^{-1} \pi_\zeta = \begin{pmatrix} 1 & 2 & 3 \\ 3 & 2 & 1 \end{pmatrix}^{-1} \begin{pmatrix} 1 & 2 & 3 \\ 1 & 2 & 3 \end{pmatrix} = (1 \ 3) \in \text{Aut}(G, \pi)$. Ne postoje drugi automorfizmi jer ne postoje drugi listovi ekvivalentni prvom listu stabla, pa je $\text{Aut}(G, \pi) = \{id, (1 \ 3)\}$.*

Definicija. Neka je ν^* list stabla $\mathcal{T}(G, \pi)$ u kom invarijanta $\phi(G, \pi, \nu)$ dostiže maksimum. *Kanonska forma određena invarijantom stabla obojenog grafa (G, π) je funkcija $\mathcal{C}(G, \pi) = (G, \pi)^{\pi_{\nu^*}}$.*

Primetimo da zbog uslova ($\phi 2$) definicija ne zavisi od izbora lista ν^* . Naredna teorema opravdava naziv i oznaku funkcije.

Teorema 2. *Funkcija $\mathcal{C}(G, \pi)$ je kanonska forma.*

Dokaz. Dokazujemo da ovako definisana funkcija ispunjava uslove kanonske forme za svaki obojen graf (G, π) .

(C1) Kako je $\mathcal{C}(G, \pi) = (G, \pi)^{\pi_{\nu^*}}$ to je $\mathcal{C}(G, \pi) \cong (G, \pi)$ za izomorfizam π_{ν^*} .

(C2) Za svako $g \in S_n$ i svako $\nu \in \mathcal{T}(G, \pi)$ važi $\nu^g \in \mathcal{T}(G, \pi)^g = \mathcal{T}(G^g, \pi^g)$ kao i $\phi(G^g, \pi^g, \nu^g) = \phi(G, \pi, \nu)$, pa je ν^g list u kom invarijanta stabla $\mathcal{T}(G^g, \pi^g)$ dostiže maksimalnu vrednost. Odatle sledi $\mathcal{C}(G^g, \pi^g) = (G^g, \pi^g)^{R(G^g, \pi^g, \nu^{*g})} = (G^g, \pi^g)^{\pi_{\nu^{*g}}} = (G, \pi)^{\pi_{\nu^*}} = \mathcal{C}(G, \pi)$ pa je \mathcal{C} funkcija invarijantna na imenovanje čvorova.

\square

Primer 12. Posmatrajmo stablo sa slike 2.8. Na osnovu prethodne teoreme je moguće odrediti kanonsku formu grafa. U pitanju je graf koji se dobija bilo kojom od permutacija $g_1 = [2 \mid 1 \mid 3]$ tj. $g_1 : \begin{pmatrix} 1 & 2 & 3 \\ 2 & 1 & 3 \end{pmatrix}$ ili $g_2 = [2 \mid 3 \mid 1]$ tj. $g_2 : \begin{pmatrix} 1 & 2 & 3 \\ 3 & 1 & 2 \end{pmatrix}$ jer su to permutacije sa najvećom invarijantom u stablu.

2.4 Odsecanje pretrage

Stablo pretrage može biti veoma veliko, pa pretraga kompletnog stabla nije poželjna. To možemo rešiti uvođenjem tri različite operacije odsecanja.

- Neka su ν_1 i ν_2 različiti čvorovi stabla $\mathcal{T}(G, \pi)$ takvi da je $|\nu_1| = |\nu_2|$ i $\phi(G, \pi, \nu_1) > \phi(G, \pi, \nu_2)$. Operacija $P_A(\nu_1, \nu_2)$ podrazumeva odsecanje podstabla $\mathcal{T}(G, \pi, \nu_2)$.
- Neka su ν_1 i ν_2 različiti čvorovi stabla $\mathcal{T}(G, \pi)$ takvi da je $|\nu_1| = |\nu_2|$ i $\phi(G, \pi, \nu_1) \neq \phi(G, \pi, \nu_2)$. Operacija $P_B(\nu_1, \nu_2)$ podrazumeva odsecanje podstabla $\mathcal{T}(G, \pi, \nu_2)$.
- Neka su ν_1 i ν_2 različiti čvorovi stabla $\mathcal{T}(G, \pi)$ takvi da je $\nu_1 < \nu_2$ i $\nu_1^g = \nu_2$ za neko $g \in \text{Aut}(G, \pi)$. Operacija $P_C(\nu_1, g)$ podrazumeva odsecanje podstabla $\mathcal{T}(G, \pi, \nu_2)$.

Operacije P_A i P_C su definisane tako da proizvoljna primena ovih operacija ne uklanja mogućnost određivanja kanonske forme u stablu. Slično, proizvoljna primena operacija P_B i P_C ne uklanja mogućnost određivanja grupe automorfizama grafa. Naredna teorema opravdava uvođenje ovih operacija i pokazuje da one ne narušavaju rezultate teorema o određivanju grupe automorfizama i kanonske forme.

Teorema 3. Neka je (G, π) obojen graf.

1. Neka je nad stablom $\mathcal{T}(G, \pi)$ izvršen proizvoljan niz operacija P_A i P_C . Tada u dobijenom stablu postoji bar jedan list ν takav da je $\phi(G, \pi, \nu) = \phi(G, \pi, \nu^*)$.
2. Neka je ν_0 list stabla $\mathcal{T}(G, \pi)$ i neka je nad stablom izvršen proizvoljan niz operacija $P_B(\nu_1, \nu_2)$ i P_C gde je $\phi(G, \pi, \nu_2) \neq \phi(G, \pi, [\nu_0]_s)$ za sve $0 \leq s \leq |\nu_0|$ i neka su g_1, \dots, g_k svi automorfizmi korišćeni u izvršenim operacijama P_C . Tada je grupa automorfizama $\text{Aut}(G, \pi)$ generisana skupom $\{g_1, \dots, g_k\} \cup \{g \in \text{Aut}(G, \pi) \mid \nu_0^g \text{ nije uklonjen}\}$.

Dokaz. Dokažimo za početak nekoliko pomoćnih tvrđenja.

Nijedna operacija P_A ne uklanja listove u kojima je vrednost invarijante maksimalna. Pretpostavimo suprotno. Neka je ν_1 list u kom invarijanta stabla dostiže maksimum i neka je ν'_1 predak od ν_1 . Operacija $P_A(\nu'_2, \nu'_1)$ uklanja ν'_1 ako je $\phi(G, \pi, \nu'_1) < \phi(G, \pi, \nu'_2)$, pa po svojstvu ($\phi 1$) za proizvoljan list ν_2 u $\mathcal{T}(G, \pi, \nu'_2)$ važi $\phi(G, \pi, \nu_1) < \phi(G, \pi, \nu_2)$, što je u kontradikciji sa pretpostavkom da je vrednost invarijante maksimalna u ν_1 .

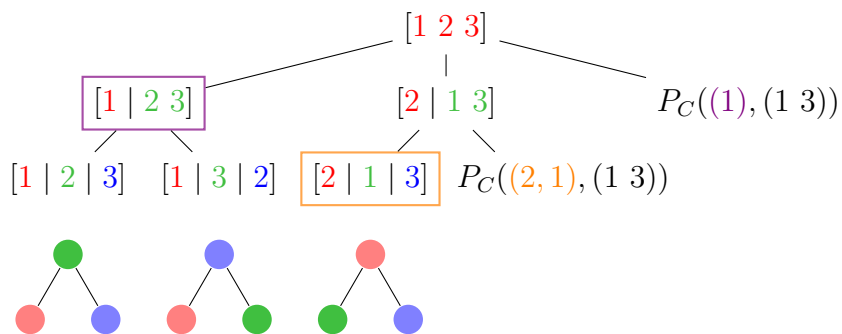
Nijedna operacija P_B ne uklanja nijedan list ν ekvivalentan listu ν_0 (iz drugog dela teoreme). Iz pretpostavke teoreme nijedna operacija $P_B(\nu_1, \nu_2)$ ne uklanja čvor ν_2 takav da je $\phi(G, \pi, [\nu_0]_{|\nu_2|}) = \phi(G, \pi, \nu_2)$, pa samim tim ne uklanja nijedan čvor $[\nu]_s$ za $0 \leq s \leq |\nu|$.

Nijedna operacija P_C ne uklanja leksikografski najmanji među ekvivalentnim listovima. Štaviše, nijedna operacija P_C ne uklanja leksikografski najmanji čvor iz $\Omega_\nu^{\langle g_1, \dots, g_k \rangle}$ za bilo koje ν . Neka je bez umanjenja opštosti ν leksikografski najmanji list u svojoj orbiti. Operacija $P_C(\omega, g)$ uklanja čvor ω^g ako je $\omega < \omega^g$, pa je $[\nu]_{|\omega|} \neq \omega^g$ pošto je ili $[\nu]_{|\omega|} < \omega$ ili su ω i $[\nu]_{|\omega|}$ iz različitih orbita.

1. Na osnovu dokazanih svojstava operacija P_A i P_C iz stabla se ne uklanja leksikografski najmanji list ν ekvivalentan listu ν^* .
2. Ako je $g \in \text{Aut}(G, \pi)$, onda na osnovu dokazanih svojstava operacija P_B i P_C važi da iz stabla nije uklonjen leksikografski najmanji list oblika ν_0^{hg} za neko $h \in \langle g_1, \dots, g_k \rangle$, izborom $\omega = \nu_0^g$. Odatle sledi da je $hg \in \langle \{g_1, \dots, g_k\} \cup \{g \in \text{Aut}(G, \pi) \mid \nu_0^g \text{ nije uklonjen} \} \rangle$, pa je i g element generisane grupe.

□

Primer 13. Prilikom obilaska stabla sa slike 2.8 moguće je izvršiti odsecanje nekih delova stabla na osnovu automorfizma (1 3) (ukoliko je on poznat unapred). Rezultujuće stablo je dato na slici 2.9. Čvor (2, 3) moguće je odseći na osnovu čvora (2, 1). Čvor (3) moguće je odseći na osnovu čvora (1).



Slika 2.9: Odsecanje stabla na osnovu automorfizama.

Glava 3

Realizacija algoritma

U ovoj glavi konkretizovani su parametri opšteg algoritma. Uvedeni su pojmovi *ekvitabilnog bojenja* i *količničkog grafa* ključni za određivanje funkcije profinjavanja i invarijante stabla i prikazani su efikasni postupci za njihovo izračunavanje. Prikazane su strukture podataka za reprezentaciju objekata uvedenih u prethodnoj glavi i opisan je kompletan algoritam pretrage za određivanje kanonske forme. Na kraju, uveden je pojam *invarijante grafa* koji indukuje klasu moćnijih funkcija profinjavanja.

3.1 Reprezentacija podataka

Bojenje

Bojenje $\pi \in \Pi_n$ predstavljeno je permutacijom p i nizom $cells$. Permutacija p predstavlja niz koji se dobija nadovezivanjem ćelija $\pi^{-1}(1), \dots, \pi^{-1}(k)$ redom, gde je k broj boja i pri čemu su elementi jedne ćelije uređeni rastuće. Na primer, bojenje $\pi = [2\ 6\ | \ 4\ 3\ 5\ | \ 1]$ predstavljeno je permutacijom $p : \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 2 & 6 & 3 & 4 & 5 & 1 \end{pmatrix}$ koja se dobija kada se poređaju rastuće elementi druge ćelije. Primetimo da u slučaju diskretnog bojenja π važi $p = \pi^{-1}$.

Niz $cells$ dužine n upotpunjava permutaciju p podacima o granicama ćelija bojenja π . Ako pozicija u predstavlja početak nove ćelije, tada je $cells(u)$ takvo da ta ćelija obuhvata tačno pozicije $[u, cells(u))$ permutacije p , a u suprotnom je $cells(u) = -1$. Na istom primeru, počeci novih ćelija nalaze se na pozicijama 1, 3 i 6, pa je $cells(1) = 3$, $cells(3) = 6$ i $cells(6) = 7$, dok su vrednosti na ostalim

pozicijama -1 , odnosno $cells : \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 3 & -1 & 6 & -1 & -1 & 7 \end{pmatrix}$ jer prva ćelija obuhvata elemente na pozicijama $[1, 3)$, druga $[3, 6)$, a treća $[6, 7)$.

Jedna od ključnih operacija nad bojenjem je profinjavanje jedne ćelije. Neka je $\pi^{-1}(c)$ jedna ćelija bojenja π i neka je svakom čvoru v te ćelije dodeljena proizvoljna vrednost $t(v)$. Profinjavanje ćelije podrazumeva formiranje bojenja $\pi' \leq \pi$ takvog da za u i v iz $\pi^{-1}(c)$ važi $\pi'(u) < \pi'(v) \iff t(u) < t(v)$, dok za sve ostale parove u i v važi $\pi'(u) = \pi'(v) \iff \pi(u) = \pi(v)$.

Algoritam 1 Profinjavanje ćelije

Ulaz Obojen graf (G, π) pri čemu je π predstavljeno pomoću p_π i $cells_\pi$,
ćelija koju treba profiniti C , niz proizvoljnih vrednosti t

Izlaz Profinjeno bojenje π' , niz ćelija dobijenih profinjavanjem C_1, \dots, C_k ,
indeks prve ćelije među C_1, \dots, C_k sa najvećim brojem elemenata s

```

1: procedure REFINE_CELL( $G, \pi, C, t$ )
2:    $\pi' \leftarrow \pi$ 
3:    $k \leftarrow 1$ 
4:    $i \leftarrow \min p_{\pi'}^{-1}(C)$ 
5:   for  $t' \in \text{sorted}(\{t(v) \mid v \in C\})$  do
6:      $C_k \leftarrow \{v \in C \mid t(v) = t'\}$ 
7:      $j \leftarrow i$ 
8:     for  $v \in \text{sorted}(C_k)$  do
9:        $p_{\pi'}(i) \leftarrow v$ 
10:       $i \leftarrow i + 1$ 
11:       $cells_{\pi'}(j) \leftarrow i$ 
12:       $k \leftarrow k + 1$ 
13:    $s \leftarrow$  indeks prvog skupa među  $C_{i, 1 \leq i \leq k}$  sa najvećim brojem elemenata
14:   return  $\pi', C_1, \dots, C_k, s$ 

```

Opisana reprezentacija bojenja je korisna zato što omogućava da se ovaj algoritam implementira tako da mu složenost bude $O(|C| \log |C|)$, pri čemu dominira složenost sortiranja čvorova na osnovu vrednosti $t(v)$.

Primer 14. Neka je $\pi = [2 \ 8 \mid 1 \ 3 \ 4 \ 5 \ 7 \mid 6]$ i potrebno je profiniti ćeliju $C = \{1, 3, 4, 5, 7\}$ u odnosu na vrednosti $t : \begin{pmatrix} 1 & 3 & 4 & 5 & 7 \\ 2 & 4 & 1 & 2 & 4 \end{pmatrix}$. Rezultujuće bojenje je $\pi' = [2 \ 8 \mid 4 \mid 1 \ 5 \mid 3 \ 7 \mid 6]$. Nove ćelije su $C_1 = \{4\}$, $C_2 = \{1, 5\}$ i $C_3 = \{3, 7\}$ i važi $s = 2$ kako ćelije C_2 i C_3 imaju najveći broj elemenata, a C_2 je prva među njima.

Graf

Pošto se graf ne menja tokom izvršavanja samog algoritma, moguće je čuvati više različitih reprezentacija grafa. Ovde ćemo istaći one delove reprezentacije koji su neophodni za dalju diskusiju.

Obojen graf je predstavljen bojenjem π i matricom povezanosti ψ . Zapazimo sledeća svojstva matrice povezanosti:

$$(\psi 1) \quad \psi(G, u, v) = \psi(G, v, u)$$

$$(\psi 2) \quad \psi(G^g, u^g, v^g) = \psi(G, v, u) \text{ za proizvoljno } g \in S_n$$

Prvo svojstvo je simetričnost matrice povezanosti koja proizilazi iz činjenice da je u neusmerenom grafu $(u, v) \in E$ akko $(v, u) \in E$. Drugo svojstvo govori da je matrica povezanosti funkcija invarijantna na imenovanje čvorova. Ovo svojstvo će biti značajno u definicijama i dokazima koji slede u ovom poglavlju. Invarijantnost važi zato što je (u, v) grana grafa G akko je $(u, v)^g = (u^g, v^g)$ grana grafa G^g .

3.2 Funkcija odabira ciljne ćelije

Postoji više strategija za realizaciju funkcije odabira ciljne ćelije. Ovde ćemo predstaviti dve takve strategije.

Jedna mogućnost je odabir neke (konkretno prve) višečlane ćelije bojenja najmanje veličine. Obrazloženje za ovakav izbor je da manje ćelije imaju veću šansu da sadrže mali broj orbita (ili čak jednu) [7] i da je samim tim očekivani faktor grananja manji.

Druga mogućnost, koja se ispostavila kao bolja u većini slučajeva je odabir prve višečlane ćelije bez obzira na njenu veličinu [7]. U daljem radu pretpostavljamo postojanje procedure `Target_cell(π)` koja implementira ovu strategiju.

3.3 Funkcija profinjavanja

Definiciju funkcije profinjavanja ćemo podeliti na dve komponente. Prva komponenta je transformacija koja određuje kako se vrši prelaz sa čvora ν na čvor $\nu \parallel w$ u stablu, odnosno kako se izbor čvora w odražava na bojenje.

Definicija. Preslikavanje $I : \Pi \times V \rightarrow \Pi$ definisano sa

$$I(\pi, v)(w) = \begin{cases} \pi(w), & \text{ako je } \pi(w) < \pi(v) \text{ ili } w = v \\ \pi(w) + 1, & \text{inače} \end{cases}$$

naziva se *funkcija individualizacije*.

Uloga funkcije individualizacije je da od zadatog bojenja formira novo, finije bojenje koje se od polaznog razlikuje samo u tome što je odabrani čvor izdvojen u zasebnu ćeliju. Ovo će obezbediti ispunjenost uslova (R2) funkcije profinjavanja.

Primer 15. Neka je $\pi = [1 \ 3 \mid 2 \ 4 \ 5]$. Tada je $I(\pi, 4) = [1 \ 3 \mid 4 \mid 2 \ 5]$.

Naredno tvrđenje ističe tri važna svojstva funkcije individualizacije. Štaviše, za funkciju I moguće bi bilo odabrati bilo koje preslikavanje koje ispunjava ta svojstva, ali nam je cilj da individualizaciju razdvojimo od ostatka definicije funkcije profinjavanja, pa zbog toga nećemo razmatrati druga preslikavanja.

Lema 5. *Funkcija individualizacije I ispunjava naredne uslove:*

(I1) *Za svako bojenje π i čvor u važi $I(\pi, u) \leq \pi$.*

(I2) *$\{u\}$ je ćelija bojenja $I(\pi, u)$.*

(I3) *I je transformacija invarijantna na imenovanje čvorova.*

Dokaz. Dokažimo redom tražena svojstva.

(I1) Neka je $\pi(v) < \pi(w)$. Tada nije istovremeno $I(\pi, u)(v) = \pi(v) + 1$ i $I(\pi, u)(w) = \pi(w)$ jer to povlači da je $\pi(v) \geq \pi(u)$ i $\pi(w) \leq \pi(u)$, odnosno da je $\pi(w) \leq \pi(v)$ što je kontradikcija. U svim ostalim slučajevima iz pretpostavke sledi $I(\pi, u)(v) < I(\pi, u)(w)$.

(I2) Ako je $\pi(w) = \pi(v)$ i $w \neq v$ onda je $I(\pi, v)(w) = I(\pi, v)(v) + 1$, a kako je $I(\pi, v) \leq \pi$ onda je $\pi(v) \neq \pi(w) \iff I(\pi, v)(v) \neq I(\pi, v)(w)$ pa je $I(\pi, v)(v) \neq I(\pi, v)(w)$ za sve $w \neq v$.

(I3) Neka je $g \in S_n$ proizvoljno. Tada je $I(\pi^g, v^g)(w^g) = \pi(w) \iff I(\pi^g, v^g)(w^g) = \pi^g(w^g) \iff \pi^g(w^g) < \pi^g(v^g) \vee w^g = v^g \iff \pi(w) < \pi(v) \vee w = v \iff I(\pi, v)(w) = \pi(w)$. Slično je i $I(\pi^g, v^g)(w^g) = \pi(w) + 1 \iff I(\pi, v)(w) = \pi(w) + 1$, odnosno $I(\pi^g, v^g)(w^g) = I(\pi, v)(w) = I(\pi, v)^g(w^g)$.

□

Druga komponenta definicije funkcije profinjavanja je transformacija koja se vrši nad bojenjem dobijenim individualizacijom. Cilj ove transformacije je da što više profini bojenje kako bi stepen grananja prilikom pretrage bio što manji. Idealna transformacija je takva da ćelije rezultujućeg bojenja odgovaraju orbitama čvorova u grupi automorfizama početnog bojenja.

Lema 6. *Neka je $F : \mathcal{G} \times \Pi \times V^* \rightarrow \Pi$ transformacija invarijantna na imenovanje čvorova takva da je $F(G, \pi, \nu) \leq \pi$. Tada je preslikavanje definisano sa*

$$\begin{aligned} R(G, \pi, ()) &= F(G, \pi, ()) \\ R(G, \pi, \nu \parallel w) &= F(G, I(R(G, \pi, \nu), w), \nu \parallel w) \end{aligned}$$

funkcija profinjavanja.

Dokaz. Dokažimo indukcijom da ovako definisana funkcija R ispunjava uslove funkcije profinjavanja.

Baza indukcije

- (R1) $R(G, \pi, ()) = F(G, \pi, ()) \leq \pi$
- (R2) Tvrđenje trivijalno važi zato što je $()$ prazan niz
- (R3) Za svako $g \in S_n$ važi $R(G^g, \pi^g, ()^g) = F(G^g, \pi^g, ()^g) = F(G, \pi, ())^g = R(G, \pi, ())^g$

Induktivni korak Pretpostavimo da tvrđenje važi za ν .

- (R1) $R(G, \pi, \nu \parallel w) = F(G, I(R(G, \pi, \nu), w), \nu \parallel w) \leq I(R(G, \pi, \nu), w) \leq R(G, \pi, \nu) \leq \pi$
- (R2) $R(G, \pi, \nu \parallel w) \leq I(R(G, \pi, \nu), w)$ pa je w ćelija bojenja $R(G, \pi, \nu \parallel w)$. Ako je $v \in \nu$, onda je v ćelija $R(G, \pi, \nu)$, pa je ćelija i bojenja $R(G, \pi, \nu \parallel w)$ jer je $R(G, \pi, \nu \parallel w) \leq R(G, \pi, \nu)$.

(R3) Za svako $g \in S_n$ važi

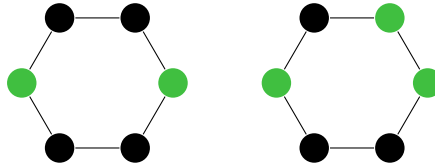
$$\begin{aligned}
 R(G^g, \pi^g, (\nu||w)^g) &= F(G^g, I(R(G^g, \pi^g, \nu^g), w^g), (\nu||w)^g) \\
 &= F(G^g, I(R(G, \pi, \nu)^g, w^g), (\nu||w)^g) \\
 &= F(G^g, I(R(G, \pi, \nu), w)^g, (\nu||w)^g) \\
 &= F(G, I(R(G, \pi, \nu), w), (\nu||w))^g \\
 &= R(G, \pi, \nu||w)^g
 \end{aligned}$$

□

Sada umesto funkcije profinjavanja R kao parametar algoritma posmatramo funkciju F . Kako bi stablo pretrage bilo što manje, i samim tim pretraga efikasnija, ključno je odabrati što moćniju funkciju. U svrhu odabira konkretne funkcije F uvodimo pojam *ekvitabilnog bojenja*.

Definicija. Označimo sa $\tilde{\psi}(G, u, W) = \sum_{w \in W} \psi(G, u, w)$ broj grana grafa G koje povezuju čvor u i skup čvorova W . Particija \sim skupa čvorova V je *ekvitabilna* ako za svaki par čvorova u i v takvih da je $u \sim v$ i svaku klasu C particije \sim važi $\tilde{\psi}(G, u, C) = \tilde{\psi}(G, v, C)$. Bojenje π je *ekvitabilno* ako je particija \sim_π ekvitabilna.

Primer 16. Na slici 3.1 su prikazana dva bojenja istog grafa. Levo bojenje je ekvitabilno zato što je svaki zeleni čvor povezan sa tačno dva crna, a svaki crni čvor sa tačno jednim zelenim i tačno jednim crnim. Desno bojenje nije ekvitabilno zato što postoje zeleni čvorovi sa različitim brojem zelenih suseda (levi zeleni čvor nije povezan ni sa jednim zelenim, dok su preostala dva zelena čvora međusobno povezana).



Slika 3.1: Primer ekvitabilnog (levo) i neekvitabilnog (desno) bojenja.

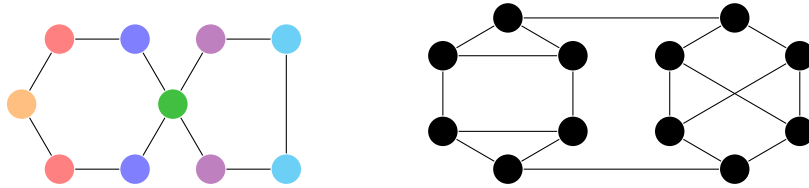
Lema 7. Za proizvoljno bojenje π grafa G postoji jedinstvena najgrublja particija \sim_γ koja je ekvitabilna i finija od \sim_π .

Dokaz. Očigledno postoji bar jedna ekvitabilna particija finija od \sim_π ; diskretna particija ispunjava te uslove. Neka su \sim_α i \sim_β dve takve particije. Pokažimo da je particija $\sim_\alpha, \sim_\beta \leq \sim_{\alpha\vee\beta} \leq \sim_\pi$ ekvitabilna.

Neka je $u \sim_\alpha v$ i neka je C proizvoljna klasa iz $\sim_{\alpha\vee\beta}$. Kako je \sim_α finije od $\sim_{\alpha\vee\beta}$, to je $C = \bigcup_{i=1}^n A_i$ za neke klase A_1, \dots, A_n particije \sim_α , pa je $\tilde{\psi}(G, u, C) = \sum_{i=1}^n \tilde{\psi}(G, u, A_i)$. Kako je α ekvitalno, to je dalje jednako $\sum_{i=1}^n \tilde{\psi}(G, v, A_i) = \tilde{\psi}(G, v, C)$. Analogno se pokazuje i za \sim_β . Označimo $u \sim v \iff u \sim_\alpha v \vee u \sim_\beta v$. Tada važi $u \sim v \implies \tilde{\psi}(G, u, C) = \tilde{\psi}(G, v, C)$. Konačno, ako je $u \sim_{\alpha\vee\beta} v$, onda je $u = x_1 \sim x_2 \sim \dots \sim x_n = v$, pa je $\tilde{\psi}(G, u, C) = \tilde{\psi}(G, x_1, C) = \tilde{\psi}(G, x_2, C) = \dots = \tilde{\psi}(G, x_n, C) = \tilde{\psi}(G, v, C)$.

Kako za bilo koje dve particije postoji grublja ekvitalna particija finija od \sim_π , skup svih takvih particija ima najveći element. To je tražena particija \sim_γ . \square

Primer 17. Na slici 3.2 prikazana su dva grafa obojena svojim najgrubljim ekvitalnim bojenjem. Čelije ekvitalnog bojenja levog grafa su upravo orbite čvorova u grupi automorfizama grafa što znači da ne postoji bolje (finije) bojenje koje je moguće konstruisati bilo kojom transformacijom invarijantnom na imenovanje čvorova. To naravno ne mora uvek biti slučaj, što se može videti iz primera desnog grafa čije je najgrublje ekvitalno bojenje jednobožno uprkos tome što postoji više orbita.



Slika 3.2: Primer najgrubljeg ekvitalnog bojenja dva grafa.

Ovim smo pokazali da postoji najgrublje ekvitalno bojenje finije od π određeno do na raspored ćelija. Definišimo onda funkciju $F(G, \pi, \nu)$ kao rezultat izvršavanja algoritma određivanja jednog takvog bojenja.

Algoritam 2 Profinjavanje bojenja

Ulaz Obojen graf (G, π) , čvor stabla ν

Izlaz Najgrublje ekvitalno bojenje finije od π

```

1: procedure REFINE( $G, \pi, \nu$ )
2:    $\alpha \leftarrow \emptyset$ 
3:   if  $\nu = \nu' || w$  then
4:     push( $\alpha, \{w\}$ )
5:   else
6:     push( $\alpha, C$ ) za sve  $C \in \pi$ 
7:   while  $\alpha \neq \emptyset$  do
8:      $W \leftarrow \text{pop}(\alpha)$ 
9:     for  $C \in \pi$  do
10:       $\pi, C_1, \dots, C_k, s \leftarrow \text{Refine\_cell}(G, \pi, C, t(v) = \tilde{\psi}(G, v, W))$ 
11:      if  $C \in \alpha$  then
12:        remove( $\alpha, C$ )
13:        push( $\alpha, C_i$ ) za sve  $1 \leq i \leq k$ 
14:      else
15:        push( $\alpha, C_i$ ) za sve  $1 \leq i \leq k, i \neq s$ 
16:   return  $\pi$ 

```

Potrebno je da procedure koje modifikuju strukturu α budu implementirane na način koji garantuje invarijantnost na imenovanje čvorova. Ovo se jednostavno postiže bilo kojom implementacijom koja ne zavisi od sadržaja elemenata unutar skupa α poput steka ili reda, ili opštije implementacijom koja zavisi jedino od svojstava elemenata invarijantnih na imenovanje čvorova.

Teorema 4. *Neka je γ bojenje dobijeno izvršavanjem algoritma profinjavanja bojenja za ulaz (G, π, ν) u obojenom grafu (G, π_0) . Tada γ ispunjava uslove leme 7.*

Dokaz. Označimo sa ξ particiju koja ispunjava uslove leme 7. Dokažimo indukcijom po i da je $\xi \leq \sim_{\pi_i}$ gde π_i predstavlja bojenje π nakon i izvršenih koraka spoljne petlje algoritma. Iz ovoga će slediti $\xi \leq \sim_{\gamma}$.

Baza indukcije Po definiciji ξ je $\xi \leq \sim_{\pi}$.

Induktivni korak Neka je $\xi \leq \sim_{\pi_j}$ za sve $j \leq i$. Tada je svaka ćelija C svakog bojenja π_j , $j \leq i$ unija nekih klasa particije ξ , pa to važi i za ćeliju W odabranu u liniji 8 algoritma. Neka su x i y proizvoljni elementi iste klase particije ξ . Tada je $\tilde{\psi}(G, x, W) = \tilde{\psi}(G, y, W)$, pa oni ne mogu biti razdvojeni nakon izvršavanja koraka spoljne petlje. Prema tome je $\xi \leq \sim_{\pi_{i+1}}$.

Neka je C ćelija nekog bojenja i neka su C_1, C_2, \dots, C_k ćelije dobijene netrivialnim profinjavanjem C u odnosu na neku ćeliju W ili individualizacijom ćelije C . Možemo formirati šumu (skup stabala) čiji su čvorovi ćelije na sledeći način:

1. Koren svakog stabla šume je jedna ćelija bojenja π_0 ako je $\nu = ()$, odnosno $R(G, \pi_0, \nu')$ ako je $\nu = \nu' || w$.
2. Deca čvora C su $\{C_1, \dots, C_k\}$.

Pokažimo da je γ ekvitalno. Dokaz izvodimo iz dva dela. Prvo, pokažimo da važi sledeće tvrđenje. Ako je ćelija C prilikom izvršavanja algoritma dodata u α u liniji 4, 6, 13 ili 15 algoritma (u oznaci $C \in \mathcal{A}$), tada se ona može predstaviti kao disjunktna unija $C = W_1 \sqcup \dots \sqcup W_m$ nekih ćelija $W_1, \dots, W_m \in \mathcal{W}$ gde \mathcal{W} predstavlja skup svih ćelija uklonjenih iz α linijom 8 algoritma. Dokaz izvodimo indukcijom po strukturi stabla, odozdo naviše.

Baza indukcije Ukoliko je $C \in \mathcal{W}$, tvrđenje važi trivijalno.

Induktivni korak Kako je po završetku izvršavanja algoritma α prazan skup, C je moralo biti uklonjeno. Kako C nije uklonjeno linijom 8, moralo je biti uklonjeno linijom 12. U tom slučaju je $C = C_1 \sqcup \dots \sqcup C_k$ i C_1, \dots, C_k su dodate u α linijom 13. Po induktivnoj hipotezi se svaki od C_1, \dots, C_k može predstaviti kao disjunktna unija skupova iz \mathcal{W} , pa to važi i za C .

Definišimo $\mathcal{B} = \{C \in \mathcal{P}(V) \mid \forall x, y \gamma(x) = \gamma(y) \implies \tilde{\psi}(G, x, C) = \tilde{\psi}(G, y, C)\}$. Jasno je da je $\mathcal{W} \subseteq \mathcal{B}$. Prema prethodno pokazanom je onda i $\mathcal{A} \subseteq \mathcal{B}$. Pokažimo da za svaku ćeliju C u šumi važi da je $C \in \mathcal{B}$. Dokaz izvodimo indukcijom po strukturi stabla, odozgo naniže.

Baza indukcije Neka je C koren nekog stabla šume.

- 1° Ako je $\nu = ()$, važi $C \in \mathcal{B}$ jer je dodato u α u liniji 6 algoritma.
- 2° Ako je $\nu = \nu' || w$, bojenje $R(G, \pi_0, \nu')$ je ekvitalno, pa kako je $\gamma \leq R(G, \pi_0, \nu')$, sve njegove ćelije su u \mathcal{B} , a samim tim je i C .

Induktivni korak Neka je $C \in \mathcal{B}$ i $C = C_1 \sqcup \dots \sqcup C_k$.

- 1° Ako je $C \in \mathcal{A}$, onda su i $C_1, \dots, C_k \in \mathcal{A} \subseteq \mathcal{B}$.
- 2° Ako je $C \notin \mathcal{A}$, onda su $C_1, \dots, C_{s-1}, C_{s+1}, \dots, C_k \in \mathcal{A} \subseteq \mathcal{B}$. Tada je $\tilde{\psi}(G, x, C_s) = \tilde{\psi}(G, x, C) - \sum_{1 \leq i \leq k, i \neq s} \tilde{\psi}(G, x, C_i)$, pa je i $C_s \in \mathcal{B}$.

Kako su sve ćelije šume u \mathcal{B} , to su i sve ćelije bojenja γ , pa je γ ekvitabilno. \square

Teorema 5. *Neka je ν list stabla $\mathcal{T}(G, \pi)$. Tada je postupak određivanja $R(G, \pi, [\nu]_i)$ za sve $0 \leq i \leq |\nu|$ ukupne vremenske složenosti $O(n^2 \log n)$, pri čemu je n broj čvorova grafa.*

Dokaz. Pretpostavimo da su operacije **push**, **pop** i **remove** složenosti $O(1)$. Za fiksno W i C broj koraka jedne iteracije unutrašnje petlje je $|C||W| + |C| \log |C| + k \leq |C||W| + |C|(1 + \log |C|)$ za određivanje vrednosti $t(v)$ za svaki čvor ćelije C , izvršavanje profinjavanja ćelije C i dodavanje svih potrebnih C_i u α . Dakle, za fiksno W je broj koraka unutrašnje petlje manji ili jednak $\sum_{C \in \pi} (|C||W| + |C|(1 + \log |C|)) \leq \sum_{C \in \pi} (|C||W| + |C|(1 + \log n)) = n|W| + n(1 + \log n)$. Odatle je broj koraka za određivanje $R(G, \pi, [\nu]_i)$ najviše $\sum_{W \in \mathcal{W}_i} (n|W| + n(1 + \log n)) = n \sum_{W \in \mathcal{W}_i} |W| + |\mathcal{W}_i|n(1 + \log n)$ gde je \mathcal{W}_i skup svih ćelija na osnovu kojih je vršeno profinjavanje prilikom određivanja $R(G, \pi, [\nu]_i)$, odnosno svih ćelija W iz linije 8 algoritma. Ukupna složenost svih profinjavanja je onda $\sum_{i=1}^{|\nu|} (n \sum_{W \in \mathcal{W}_i} |W| + |\mathcal{W}_i|n(1 + \log n)) = O(n \sum_{W \in \mathcal{W}} |W| + |\mathcal{W}|n \log n)$ pri čemu je $\mathcal{W} = \bigcup_{i=1}^{|\nu|} \mathcal{W}_i$.

Neka je C ćelija nekog bojenja i neka su C_1, C_2, \dots, C_k ćelije dobijene netrivialnim profinjavanjem C u odnosu na neku ćeliju W ili individualizacijom ćelije C . Možemo formirati šumu (skup stabala) čiji su čvorovi ćelije na sledeći način:

1. Koren svakog stabla šume je jedna ćelija bojenja $R(G, \pi, ())$.
2. Deca čvora C su $\{C_1, \dots, C_k\}$.

Kako su ćelije u listovima šume sve različite jednočlane ćelije (jer je bojenje $R(G, \pi, \nu)$ diskretno), ima ih tačno n . Kako svaka ćelija šume ima bar dva deteta, broj ćelija u šumi je manji od $2n$, a pošto ona mora sadržati i sve ćelije iz \mathcal{W} , to je $|\mathcal{W}| < 2n$.

Neka je v čvor grafa. Posmatrajmo niz ćelija $\{v\} = C_m \subseteq C_{m-1} \subseteq \dots \subseteq C_1$ koje čine put od lista $\{v\}$ do korena odgovarajućeg stabla. Neka je ćelija C_i jedna od ćelija dodatih u α u liniji 4, 6, ili 15 algoritma. Tada je $|C_i| \leq \frac{|C_{i-1}|}{2}$ ako je $i > 1$, odnosno nakon individualizacije ili profinjavanja. Označimo broj ovakvih ćelija sa k_v . Očigledno je $k_v \leq 1 + \log_2 |C_1| \leq 1 + \log_2 n$.

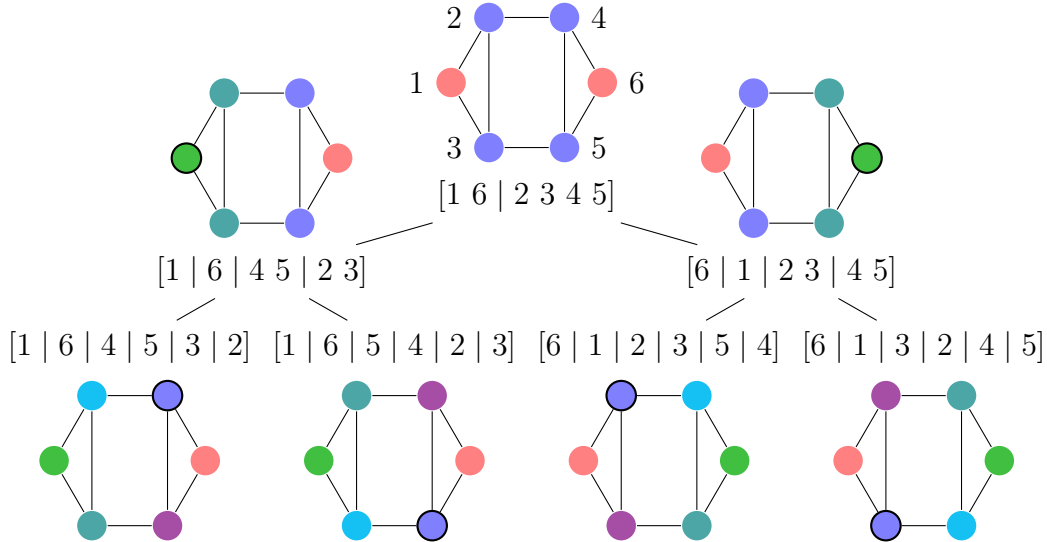
Primetimo da su u svakom koraku algoritma ćelije koje se nalaze u α međusobno disjunktne. Odavde sledi da se istovremeno u α ne mogu naći dve različite ćelije C_i i C_j . Prisetimo, takođe, da jedino linije 4, 6, 8 i 15 algoritma mogu da promene tačnost rečenice $\exists i C_i \in \alpha$ (12 i 13 ne mogu ukoliko se posmatraju zajedno). Kako

je na početku i na kraju izvršavanja algoritma $\alpha = \emptyset$, mora važiti da je broj k_v dodavanja ćelija C_i u α jednak broju uklanjanja ćelija C_i iz α , odnosno broju onih ćelija za koje važi $C_i \in \mathcal{W}$.

Konačno,

$$\begin{aligned} \sum_{W \in \mathcal{W}} |W| &= \sum_{W \in \mathcal{W}} \sum_{v \in W} 1 \\ &= \sum_{W \in \mathcal{W}} \sum_{v \in V} \chi_W(v) \\ &= \sum_{v \in V} \sum_{W \in \mathcal{W}} \chi_W(v) \\ &= \sum_{v \in V} k_v \\ &\leq \sum_{v \in V} (1 + \log_2 n) \\ &= n(1 + \log_2 n) \end{aligned}$$

gde je $\chi_W : V \rightarrow \{0, 1\}$ karakteristična funkcija skupa W . □



Slika 3.3: Stablo pretrage generisano definisanom funkcijom profinjavanja.

Primer 18. Na slici 3.3 prikazano je stablo pretrage jednobojnog grafa koje koristi definisanu funkciju profinjavanja. Prikazan je obojen graf nakon svakog profinjavanja i zaokruženi su čvorovi korišćeni prilikom individualizacije. U korenu stabla se profinjavanjem dobija ekvitalno bojenje sa dve boje, razdvajanjem čvorova stepena

dva i tri. U narednom nivou stabla se vrši individualizacija jednog od čvorova stepena dva pri čemu se profinjavanjem čvorovi stepena tri dele u dve ćelije po tome da li su povezani sa individualizovanim čvorom ili ne. U poslednjem nivou se vrši individualizacija jednog od čvorova stepena tri koji nije sused prethodno individualizovanom čvoru i profinjavanjem se dolazi do diskretnog bojenja.

3.4 Invarijanta stabla

Lema 8. Neka je $f : \mathcal{G} \times \Pi \times V^* \rightarrow F$ funkcija invarijantna na imenovanje čvorova, pri čemu je F neki potpuno uređen skup i neka je $\text{bin}(G)$ binarna reprezentacija gornjeg trougla matrice povezanosti grafa G . Tada je funkcija definisana sa

$$\phi(G, \pi, \nu) = \begin{cases} (f(G, \pi, [\nu]_0), \dots, f(G, \pi, [\nu]_{|\nu|})), & \text{ako } \pi_\nu \text{ nije diskretno} \\ (f(G, \pi, [\nu]_0), \dots, f(G, \pi, [\nu]_{|\nu|}), \text{bin}(G^{\pi_\nu})), & \text{inače} \end{cases}$$

invarijanta stabla pri leksikografskom poretku.

Dokaz. Dokažimo da tako definisana funkcija ϕ ispunjava uslove invarijante stabla.

($\phi 1$) Za svaki čvor ω podstabla $\mathcal{T}(G, \pi, \nu)$ važi da je $\phi(G, \pi, \nu) = [\phi(G, \pi, \omega)]_{|\nu|}$. Neka su ν_1 i ν_2 čvorovi stabla takvi da je $|\nu_1| = |\nu_2|$ i $\phi(G, \pi, \nu_1) < \phi(G, \pi, \nu_2)$. Tada za čvorove $\omega_1 \in \mathcal{T}(G, \pi, \nu_1)$ i $\omega_2 \in \mathcal{T}(G, \pi, \nu_2)$ važi $[\phi(G, \pi, \omega_1)]_{|\nu_1|} < [\phi(G, \pi, \omega_2)]_{|\nu_2|}$ pa je po leksikografskom poretku i $\phi(G, \pi, \omega_1) < \phi(G, \pi, \omega_2)$.

($\phi 2$) Ako za listove ν_1 i ν_2 važi $\phi(G, \pi, \nu_1) = \phi(G, \pi, \nu_2)$, onda je $\text{bin}(G^{\pi_1}) = \text{bin}(G^{\pi_2})$, pa je $G^{\pi_1} = G^{\pi_2}$.

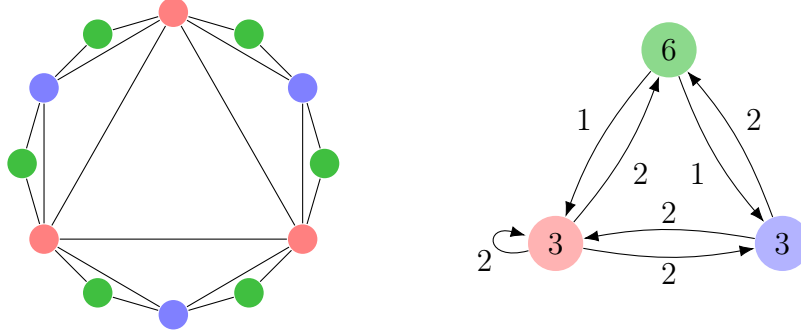
($\phi 3$) Neka je $g \in \text{Aut}(G, \pi)$ i ν čvor stabla. Tada je $f(G^g, \pi^g, [\nu^g]_i) = f(G^g, \pi^g, [\nu]_i^g) = f(G, \pi, [\nu]_i)$, kao i $\text{bin}((G^g)^{R(G^g, \pi^g, \nu^g)}) = \text{bin}((G^g)^{\pi^g}) = \text{bin}(G^{\pi_\nu})$, pa su nizovi $\phi(G^g, \pi^g, \nu^g)$ i $\phi(G, \pi, \nu)$ jednaki.

□

Još je potrebno odabrati konkretnu funkciju f . Uvedimo za početak pojam *količničkog grafa*.

Definicija. Neka je (G, π) obojen graf i π ekvitabilno. *Količnički graf* $Q(G, \pi) = (V_Q, d_Q, \psi_Q)$ je struktura takva da je V_Q skup od $|\pi|$ čvorova, $d_Q : V_Q \rightarrow \mathbb{N}$ preslikavanje takvo da je $d_Q(c) = |\pi^{-1}(c)|$ i $\psi_Q : V_Q^2 \rightarrow \mathbb{N}$ preslikavanje takvo da važi $\psi_Q(c_1, c_2) = \tilde{\psi}(G, v, \pi^{-1}(c_2))$ za bilo koje $v \in \pi^{-1}(c_1)$.

Primer 19. Na slici 3.4 prikazan je graf sa ekvitalnim bojenjem i odgovarajući količnički graf. Vrednosti $d_Q(c)$ odgovaraju broju čvorova grafa boje c i upisane su u čvorove količničkog grafa. Vrednosti $\psi_Q(c_1, c_2)$ odgovaraju broju čvorova boje c_2 grafa sa kojim je povezan svaki čvor boje c_1 i napisane su pored odgovarajućih usmerenih grana, dok grane kojima je ta vrednost nula nisu prikazane.



Slika 3.4: Ekvitalno bojenje (levo) i količnički graf (desno).

Primetimo da je definicija dobra zbog ekvitalnosti bojenja π . Sledeća lema pokazuje značaj uvedenog pojma.

Lema 9. Neka je $Q : \mathcal{G}_{eq} \rightarrow \mathcal{Q}$ prethodno definisano preslikavanje iz skupa obojenih grafova sa ekvitalnim bojenjem u skup količničkih grafova. Q je funkcija invarijantna na imenovanje čvorova.

Dokaz. Neka je $g \in S_n$ proizvoljno. Pokažimo da je $Q(G^g, \pi^g) = Q(G, \pi)$. Kako je $|\pi^g| = |\pi|$ to su skupovi čvorova količničkih grafova jednaki. Dalje, važi $d_{Q(G^g, \pi^g)}(c) = |(\pi^g)^{-1}(c)| = |\pi^{-1}(c)| = d_{Q(G, \pi)}(c)$. Konačno, neka je $u' \in (\pi^g)^{-1}(c_1)$. Tada važi

$$\begin{aligned}
 \psi_{Q(G^g, \pi^g)}(c_1, c_2) &= \tilde{\psi}(G^g, u', (\pi^g)^{-1}(c_2)) \\
 &= \sum_{v' \in (\pi^g)^{-1}(c_2)} \psi(G^g, u', v') \\
 &= \sum_{v^g \in (\pi^g)^{-1}(c_2)} \psi(G^g, u^g, v^g) && \text{smena } u' = u^g, v' = v^g \\
 &= \sum_{v \in \pi^{-1}(c_2)} \psi(G^g, u^g, v^g) && (\pi^g)^{-1}(c_2) = \pi^{-1}(c_2)^g \\
 &= \sum_{v \in \pi^{-1}(c_2)} \psi(G, u, v) && (\psi 2) \\
 &= \tilde{\psi}(G, u, \pi^{-1}(c_2)) \\
 &= \psi_{Q(G, \pi)}(c_1, c_2) && (\pi^g)^{-1}(c_1) = \pi^{-1}(c_1)^g
 \end{aligned}$$

Dakle, preslikavanje Q je funkcija invarijantna na imenovanje čvorova. \square

Posledica 2. *Preslikavanje $f_Q : \mathcal{G} \times \Pi \times V^* \rightarrow \mathcal{Q}$ definisano sa $f_Q(G, \pi, \nu) = Q(G, R(G, \pi, \nu))$ je funkcija invarijantna na imenovanje čvorova.*

Jasno je da za preslikavanje f možemo uzeti upravo ceo količnički graf, pri čemu je uređenje količničkih grafova moguće realizovati uređivanjem njihovih binarnih reprezentacija. Ovakva definicija preslikavanja f u praksi nije korisna zbog velike složenosti neophodne za njeno izračunavanje - potrebno je realizovati ceo količnički graf. Ovo je moguće rešiti posmatranjem manjeg dela količničkog grafa i to bez gubitka korisnih informacija.

Definicija. Neka je (G, π) obojen graf i $\nu = \nu' || w$ čvor stabla različit od korena. Uvedimo oznake $\pi_1 = R(G, \pi, \nu')$ i $\pi_2 = R(G, \pi, \nu)$. Neka su c_1, \dots, c_k boje takve da za sve $1 \leq i \leq k$ važi da $\pi_2^{-1}(c_i)$ nije ćelija bojenja π_1 . Označimo sa f_i niz vrednosti $(c_i, d_{Q(G, \pi_2)}(c_i), \psi_{Q(G, \pi_2)}(c_i, c_1), \dots, \psi_{Q(G, \pi_2)}(c_i, c_k))$. Preslikavanje $f(G, \pi, \nu)$ čija je vrednost prazan niz $()$ u slučaju korena ν , odnosno dobijena nadovezivanjem nizova f_1, \dots, f_k u suprotnom, zovemo *razlikom količničkih grafova*.

Primer 20. *Posmatrajmo najlevlji list i njegovog roditelja u stablu pretrage prikazanom na slici 3.3. Bojenje $[1 | 6 | 4 | 5 | 3 | 2]$ ima četiri ćelije koje nisu u bojenju $[1 | 6 | 4 | 5 | 2 | 3]$. To su $\{4\}, \{5\}, \{3\}, \{2\}$, redom za boje $c_1 = 3, c_2 = 4, c_3 = 5, c_4 = 6$. Tada je $f_1 = (c_1, d_Q(c_1), \psi_Q(c_1, c_1), \psi_Q(c_1, c_2), \psi_Q(c_1, c_3), \psi_Q(c_1, c_4)) = (3, 1, 0, 1, 0, 1)$. Dalje, $f_2 = (4, 1, 1, 0, 1, 0)$, $f_3 = (5, 1, 0, 1, 0, 1)$, $f_4 = (6, 1, 1, 0, 1, 0)$, pa je $f(G, \pi, \nu) = (3, 1, 0, 1, 0, 1, 4, 1, 1, 0, 1, 0, 5, 1, 0, 1, 0, 1, 6, 1, 1, 0, 1, 0)$.*

Lema 10. *Razlika količničkih grafova f je funkcija invarijantna na imenovanje čvorova.*

Dokaz. Pretpostavimo da za neko $g \in S_n$ i neke boje c i d važi $\pi_2^{-1}(c) = \pi_1^{-1}(d)$, odnosno da je $\pi_2^{-1}(c)$ ćelija bojenja π_1 . Tada je $R(G^g, \pi^g, \nu^g)^{-1}(c) = (\pi_2^g)^{-1}(c) = \pi_2^{-1}(c)^g = \pi_1^{-1}(d)^g = (\pi_1^g)^{-1}(d) = R(G^g, \pi^g, \nu^g)^{-1}(d)$. Analogno se pokazuje i obrnuta implikacija. Ovim smo pokazali da je niz boja c_1, \dots, c_k jednak za $f(G^g, \pi^g, \nu^g)$ i $f(G, \pi, \nu)$. Tvrđenje onda jednostavno važi pošto je količnički graf funkcija invarijantna na imenovanje čvorova. \square

Naredna teorema opravdava upotrebu invarijante stabla koja se ne oslanja na ceo količnički graf i pokazuje da ne postoji gubitak informacije prilikom njenog izračunavanja.

Teorema 6. *Neka je (G, π) obojen graf i $\nu||w_1$ i $\nu||w_2$ različiti čvorovi stabla. Tada je $f(G, \pi, \nu||w_1) = f(G, \pi, \nu||w_2)$ ako i samo ako $f_Q(G, \pi, \nu||w_1) = f_Q(G, \pi, \nu||w_2)$.*

Dokaz. Dokaz implikacije ulevo je trivijalan. Za smer udesno dokazaćemo kontrapoziciju tvrđenja.

Uvedimo oznake $\pi_1 = R(G, \pi, \nu||w_1)$ i $\pi_2 = R(G, \pi, \nu||w_2)$. Pretpostavimo da je $Q(G, \pi_1) \neq Q(G, \pi_2)$. Ako je $V_{Q(G, \pi_1)} \neq V_{Q(G, \pi_2)}$ ili $d_{Q(G, \pi_1)} \neq d_{Q(G, \pi_2)}$ onda je skup ćelija bojenja π_1 i π_2 koje nisu ćelije u π_ν različit, pa je $f(G, \pi, \nu||w_1) \neq f(G, \pi, \nu||w_2)$. Ako je $\psi_{Q(G, \pi_1)}(c_1, c_2) \neq \psi_{Q(G, \pi_2)}(c_1, c_2)$ za neke c_1 i c_2 čije odgovarajuće ćelije iz π_1 i π_2 nisu ćelije bojenja π_ν , očigledno je $f(G, \pi, \nu||w_1) \neq f(G, \pi, \nu||w_2)$.

Pretpostavimo da je $\pi_1^{-1}(c_2)$ ćelija bojenja π_ν . Tada zbog ekvitalnosti bojenja π_ν za proizvoljno c_1 važi $\psi_{Q(G, \pi_1)}(c_1, c_2) = \psi_{Q(G, \pi_\nu)}(d_1, d_2)$ gde je d_1 takvo da je $\pi_1^{-1}(c_1) \subseteq \pi_\nu^{-1}(d_1)$, a d_2 takvo da je $\pi_1^{-1}(c_2) = \pi_\nu^{-1}(d_2)$. Odatle sledi da ne važi $\psi_{Q(G, \pi_1)}(c_1, c_2) \neq \psi_{Q(G, \pi_2)}(c_1, c_2)$ jer je $\psi_{Q(G, \pi_1)}(c_1, c_2) = \psi_{Q(G, \pi_\nu)}(d_1, d_2) = \psi_{Q(G, \pi_2)}(c_1, c_2)$, a znamo da su u obe jednakosti isti d_1 i d_2 u pitanju jer su $\pi_1, \pi_2 \leq \pi_\nu$.

Pretpostavimo da je $\pi_1^{-1}(c_1)$ ćelija bojenja π_ν . Tada ne važi $\psi_{Q(G, \pi_1)}(c_1, c_2) \neq \psi_{Q(G, \pi_2)}(c_1, c_2)$ zato što po prethodno pokazanom važi $|\pi_1^{-1}(c_1)|\psi_{Q(G, \pi_1)}(c_1, c_2) = |\pi_1^{-1}(c_2)|\psi_{Q(G, \pi_1)}(c_2, c_1) = |\pi_2^{-1}(c_2)|\psi_{Q(G, \pi_2)}(c_2, c_1) = |\pi_2^{-1}(c_1)|\psi_{Q(G, \pi_2)}(c_1, c_2)$.

Dokažimo još korišćeno svojstvo $|C_1|\psi_{Q(G, \pi)}(c_1, c_2) = |C_2|\psi_{Q(G, \pi)}(c_2, c_1)$ za $C_1 = \pi^{-1}(c_1)$ i $C_2 = \pi^{-1}(c_2)$. Ono važi zato što za svako $u \in C_1$ važi $|C_1|\psi_{Q(G, \pi)}(c_1, c_2) = |C_1|\tilde{\psi}(G, u, C_2) = \sum_{u \in C_1} \tilde{\psi}(G, u, C_2) = \sum_{u \in C_1} \sum_{v \in C_2} \psi(G, u, v)$. Iz simetričnosti ($\psi 1$) i mogućnosti zamene mesta suma sledi tražena jednakost. \square

Primetimo da je vremenska složenost određivanja invarijante u čvoru $\nu||w$ u klasi $O((|\pi_2| - |\pi_1|)n)$ gde je $\pi_2 = R(G, \pi, \nu||w)$, $\pi_1 = R(G, \pi, \nu)$ i n broj čvorova grafa. Ovo se jednostavno vidi iz toga što je broj ćelija π_2 koje nisu ćelije π_1 upravo $|\pi_2| - |\pi_1|$ i iz toga što je za određivanje svakog niza f_i dovoljno $O(n)$ koraka. To znači da ukupna složenost određivanja svih invarijanti od korena do jednog od listova stabla pripada klasi $O(n^2)$.

3.5 Automorfizmi

Prilikom pretrage otkrivaju se automorfizmi koji se dalje koriste za odsecanje. Ako je poznat skup automorfizama $\{g_1, \dots, g_n\}$, u operaciji odsecanja P_C moguće je primeniti ne samo neki od tih automorfizama, već i proizvoljan automorfizam iz grupe generisane tim skupom.

Operacija odsecanja P_C se primenjuje što je ranije u stablu moguće. Konkretno, neka je $\nu = (v_1, \dots, v_{|\nu|})$ i neka je k dužina najdužeg zajedničkog prefiksa nizova ν i ν^g za neki automorfizam g . Tada je moguće primeniti operaciju odsecanja $P_C([\nu]_{k+1}, g)$. Primetimo da g stabilizuje niz $[\nu]_k$, odnosno $g \in \Sigma_{[\nu]_k}$ i da stoga v_{k+1} i v_{k+1}^g pripadaju istoj orbiti tog stabilizatora.

Zbog ovoga je korisno na efikasan način realizovati određivanje stabilizatora (ili nekog njegovog podskupa) i određivanje orbita nekog skupa automorfizama. Pored samog skupa automorfizama čuvaju se podaci koji ovo i omogućavaju.

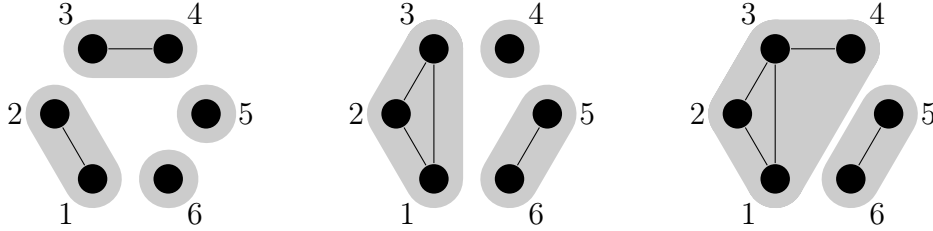
Neka je $S = \{g_1, \dots, g_n\}$ skup automorfizama. Definišimo $stab_S(v) = \{g \in S \mid v^g = v\}$. Grupa automorfizama generisana skupom $S_\nu = \bigcap_{1 \leq j \leq k} stab_S(v_j)$ je podgrupa stabilizatora niza $\nu = (v_1, \dots, v_k)$, odnosno $\langle S_\nu \rangle \leq \Sigma_\nu^{Aut(G, \pi)} = Aut(G, \pi_\nu)$.

Definišimo $mcr_S(v) = \min \Omega_v^{(S)}$. Neka je ν čvor stabla. Kako je prema prethodno pokazanom P_C moguće primenjivati za bilo koja dva čvora iz iste orbite stabilizatora, možemo odseći pretragu u bilo kom čvoru $\nu \parallel v$ za koji važi $v \neq mcr_{S_\nu}(v)$.

Prilikom dodavanja novog automorfizma g u skup potrebno je ažurirati ove podatke. Skupovi $stab_S(v)$ se jednostavno ažuriraju dodavanjem g tamo gde je potrebno. Kako bi se ažuriranje predstavnika orbita moglo efikasno izvršiti, pamti se pomoćna particija Ω čvorova po orbitama. Prilikom dodavanja se nova particija određuje kao supremum particije Ω i particije ciklusa (orbita) automorfizma g . Proizvoljnu particiju moguće je preslikati u odgovarajući graf formiranjem prostog ciklusa nad elementima svake klase particije. Određivanje supremuma dve particije moguće realizovati pretragom unije tako formiranih grafova i formiranjem particije od komponenti povezanosti dobijenog grafa. Primetimo da je broj grana ovako formiranog grafa linearno ograničen u odnosu na broj čvorova n , pa je određivanje supremuma dve particije moguće izvršiti u vremenskoj složenosti $O(n)$.

Primer 21. Neka je $S = \{(1\ 2), (1\ 3), (5\ 6)\}$ jedan skup automorfizama nekog grafa sa 6 čvorova. Tada je $stab_S(1) = \{(5\ 6)\}$, $stab_S(2) = \{(1\ 3), (5\ 6)\}$, $stab_S(3) = \{(1\ 2), (5\ 6)\}$, $stab_S(4) = \{(1\ 2), (1\ 3), (5\ 6)\}$, $stab_S(5) = stab_S(6) = \{(1\ 2), (1\ 3)\}$ i $\Omega = \{\{1, 2, 3\}, \{4\}, \{5, 6\}\}$. Dodavanjem automorfizma $(1\ 2)(3\ 4)$ dobija se $S' =$

$\{(1\ 2), (1\ 3), (5\ 6), (1\ 2)(3\ 4)\}$ i $stab_{S'}(5) = stab_{S'}(6) = \{(1\ 2), (1\ 3), (1\ 2)(3\ 4)\}$, dok ostale vrednosti $stab_S$ ostaju nepromenjene. Particija Ω' dobija se kao supremum particije Ω i $\Omega^{(1\ 2)(3\ 4)} = \{\{1, 2\}, \{3, 4\}, \{5\}, \{6\}\}$, odnosno $\Omega' = \Omega \vee \Omega^{(1\ 2)(3\ 4)} = \{\{1, 2, 3, 4\}, \{5, 6\}\}$. Na slici 3.5 prikazana su dva grafa formirana na osnovu particije $\Omega^{(1\ 2)(3\ 4)}$ i Ω , kao i njihova unija.



Slika 3.5: Grafovi koji odgovaraju particijama $\Omega^{(1\ 2)(3\ 4)}$, Ω i $\Omega' = \Omega^{(1\ 2)(3\ 4)} \vee \Omega$.

3.6 Pretraga

Generisanje i pretraga stabla realizovani su pretragom u dubinu. Osnovni algoritam generisanja stabla dat je u nastavku.

Algoritam 3 Generisanje stabla pretrage

Ulaz Obojen graf (G, π_0)

- 1: **procedure** SEARCH(G, π_0, π, ν)
 - 2: $\pi \leftarrow \text{Refine}(G, \pi, \nu)$
 - 3: $cell \leftarrow \text{Target_cell}(\pi)$
 - 4: **for** $v \in cell$ **do**
 - 5: SEARCH($G, \pi_0, I(\pi, v), \nu || v$)
 - 6: SEARCH($G, \pi_0, \pi_0, ()$)
-

Izračunavanjem invarijante stabla možemo iskoristiti prethodni algoritam za određivanje kanonske forme. Pored toga, poznavanje invarijante stabla omogućava odsecanje operacijom P_A . Tokom obilaska stabla pamti se list ρ čija je vrednost invarijante $\phi(G, \pi_0, \rho)$ najveća među svim do sada obrađenim listovima, odnosno $()$ ukoliko do sada nije obrađen nijedan list. Nakon završenog obilaska stabla čvor ρ je upravo čvor ν^* neophodan za određivanje kanonske forme.

Algoritam 4 Određivanje kanonske forme

Ulaz Obojen graf (G, π_0)

Izlaz Permutacija π_ρ

```

1: procedure SEARCH( $G, \pi_0, \pi, \nu$ )
2:   global  $\rho$ 
3:    $\pi \leftarrow \text{Refine}(G, \pi, \nu)$ 
4:   if  $|\rho| \geq |\nu| \wedge \phi(G, \pi_0, \nu) < \phi(G, \pi_0, [\rho]_{|\nu|})$  then
5:     return
6:    $cell \leftarrow \text{Target\_cell}(\pi)$ 
7:   for  $v \in cell$  do
8:     Search( $G, \pi_0, I(\pi, v), \nu || v$ )
9:   if  $cell = \emptyset$  then
10:    if  $\rho = () \vee \phi(G, \pi_0, \nu) > \phi(G, \pi_0, \rho)$  then
11:       $\rho \leftarrow \nu$ 
12:  $\rho \leftarrow ()$ 
13: Search( $G, \pi_0, \pi_0, ()$ )

```

Kako bi bilo moguće odrediti generatore grupe automorfizama i primeniti operaciju odsecanja P_C neophodno je tokom pretrage otkrivati i pamtititi otkrivene automorfizme. Održava se skup otkrivenih automorfizama S . Jedan od načina da se primeni operacija P_C je da se pretraga rekurzivno pokreće samo za one čvorove odabrane ćelije za koje važi $v = mcr_{S_\nu}(v)$.

Otkrivanje novih automorfizama omogućeno je pamćenjem jednog lista ζ , konkretno prvog otkrivenog lista. Kada se prilikom pretrage otkrije list ν , u slučaju da su ζ i ν ekvivalentni listovi $g = \pi_\zeta^{-1}\pi_\nu$ je otkriven automorfizam i on se dodaje u skup S . Kako se novi automorfizmi otkrivaju samo u listovima ekvivalentnim listu ζ , moguće je u velikoj meri primenjivati operaciju odsecanja P_B . Ovde dolazimo i do drugog načina za primenu operacije P_C . Kako je $\nu^g = \zeta$, g je automorfizam koji stabilizuje $lca(\nu, \zeta)$ (najduži zajednički prefiks za ν i ζ). Ako je njegova dužina $k = |lca(\nu, \zeta)|$, tada je $[\nu]_{k+1}^g = [\zeta]_{k+1}$ i moguće je primeniti operaciju odsecanja $P_C([\nu]_{k+1}, g)$. Ovo je moguće realizovati skokom unazad - povratna vrednost određuje do kog nivoa u pretrazi je potrebno vratiti se. Nakon završenog obilaska stabla skup S predstavlja skup generatora grupe $Aut(G, \pi_0)$.

Algoritam 5 Određivanje generatora grupe automorfizama

Ulaz Obojen graf (G, π_0)

Izlaz Skup generatora S

```

1: procedure SEARCH( $G, \pi_0, \pi, \nu$ )
2:   global  $S, \zeta$ 
3:    $\pi \leftarrow \text{Refine}(G, \pi, \nu)$ 
4:   if  $\zeta \neq () \wedge (|\zeta| < |\nu| \vee \phi(G, \pi_0, \nu) \neq \phi(G, \pi_0, [\zeta]_{|\nu|}))$  then
5:     return  $|\nu|$ 
6:    $cell \leftarrow \text{Target\_cell}(\pi)$ 
7:    $mcr \leftarrow cell$ 
8:   for  $v \in cell$  do
9:     if  $v \in mcr$  then
10:       $b \leftarrow \text{Search}(G, \pi_0, I(\pi, v), \nu||v)$ 
11:      if  $|\nu| > b$  then
12:        return  $b$ 
13:       $mcr \leftarrow mcr \cap \{v \in V \mid v = mcr_{S_v}(v)\}$ 
14:   if  $cell = \emptyset$  then
15:     if  $\zeta = ()$  then
16:        $\zeta \leftarrow \nu$ 
17:        $g = \pi_\zeta^{-1} \pi_\nu$ 
18:       if  $G^g = G$  then
19:          $\text{insert}(S, g)$ 
20:       return  $|lca(\nu, \zeta)|$ 
21:   return  $|\nu|$ 
22:  $S \leftarrow \emptyset, \zeta \leftarrow ()$ 
23: SEARCH( $G, \pi_0, \pi_0, ()$ )

```

Konačno, kako bismo iskoristili automorfizme u odsecanju prilikom određivanja kanonske forme, potrebno je da istovremeno vršimo prethodna dva algoritma. Primetimo da je otkrivanje automorfizama sada moguće vršiti i u odnosu na list ρ .

Algoritam 6 Određivanje kanonske forme i grupe automorfizama

Ulaz Obojen graf (G, π_0)

Izlaz Permutacija π_ρ i skup generatora S

```

1: procedure SEARCH( $G, \pi_0, \pi, \nu$ )
2:   global  $S, \zeta, \rho$ 
3:    $\pi \leftarrow \text{Refine}(G, \pi, \nu)$ 
4:    $\text{max\_path} \leftarrow |\rho| < |\nu| \vee \phi(G, \pi_0, \nu) \geq \phi(G, \pi_0, [\rho]_{|\nu|})$ 
5:    $\text{aut\_path} \leftarrow |\zeta| \geq |\nu| \wedge \phi(G, \pi_0, \nu) = \phi(G, \pi_0, [\zeta]_{|\nu|})$ 
6:   if  $\neg \text{max\_path} \wedge \neg \text{aut\_path}$  then
7:     return  $|\nu|$ 
8:    $\text{cell} \leftarrow \text{Target\_cell}(\pi)$ 
9:    $\text{mcr} \leftarrow \text{cell}$ 
10:  for  $v \in \text{cell}$  do
11:    if  $v \in \text{mcr}$  then
12:       $b \leftarrow \text{Search}(G, \pi_0, I(\pi, v), \nu||v)$ 
13:      if  $|\nu| > b$  then
14:        return  $b$ 
15:       $\text{mcr} \leftarrow \text{mcr} \cap \{v \in V \mid v = \text{mcr}_{S_\nu}(v)\}$ 
16:  if  $\text{cell} = \emptyset$  then
17:    if  $\rho = () \vee \phi(G, \pi_0, \nu) > \phi(G, \pi_0, \rho)$  then
18:       $\rho \leftarrow \nu$ 
19:    if  $\zeta = ()$  then
20:       $\zeta \leftarrow \nu$ 
21:     $g = \pi_\zeta^{-1} \pi_\nu$ 
22:    if  $G^g \neq G$  then
23:       $g = \pi_\rho^{-1} \pi_\nu$ 
24:    if  $G^g = G$  then
25:       $\text{insert}(S, g)$ 
26:      if  $\text{mcr}_{S_{\text{lca}(\nu, \zeta)}}(v) \neq v$  where  $\nu = \nu' || v$  then
27:        return  $|\text{lca}(\nu, \zeta)|$ 
28:      return  $|\text{lca}(\nu, \rho)|$ 
29:  return  $|\nu|$ 
30:  $S \leftarrow \emptyset, \zeta \leftarrow (), \rho \leftarrow ()$ 
31: SEARCH( $G, \pi_0, \pi_0, ()$ )

```

Za otkrivanje automorfizama moguće je čuvati i više od pomenuta dva lista ρ i ζ . U tom slučaju je moguće ranije u pretrazi otkriti veći deo grupe automorfizama, ali tada sama provera ekvivalentnosti listova prilikom otkrivanja automorfizma postaje sporija.

Moguće su i drugačije strategije obilaska stabla. Program *Traces* vrši pretragu

stabla u širinu što omogućava odsecanje svih čvorova stabla nekog nivoa čija invarijanta nije maksimalna. Pošto ovakvim obilaskom nije moguće direktno otkrivati automorfizme kao u slučaju pretrage u dubinu, puštaju se pomoćni slučajni putevi do listova stabla prilikom pretrage sa svrhom otkrivanja automorfizama na način sličan prethodno opisanom.

3.7 Invarijanta grafa

U definiciji pojma ekvitabilnog bojenja uvedeno je preslikavanje $\tilde{\psi}(G, u, W) = \sum_{v \in W} \psi(G, u, v)$ koja predstavlja broj grana koje povezuju čvor u sa čvorovima iz skupa W . Možemo primetiti da se stavovi i dokazi koji koriste koncept ekvitabilnosti ne pozivaju na konkretno značenje funkcije ψ , već samo na određena svojstva te funkcije koja nam omogućavaju da uopštimo značenje ekvitabilnog bojenja.

Definicija. Preslikavanje $\psi : \mathcal{G} \times V \times V \rightarrow \mathbb{N}$ je *invarijanta grafa* ukoliko za svaki graf G i par čvorova u i v zadovoljava sledeće uslove:

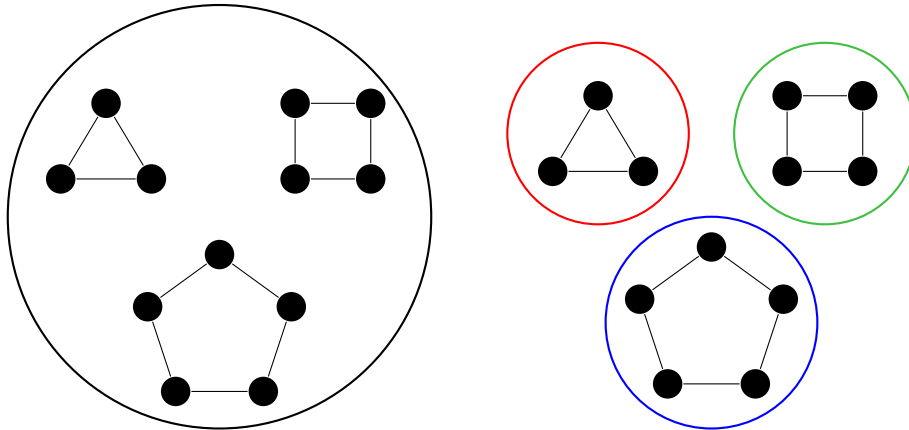
$$(\psi 1) \quad \psi(G, u, v) = \psi(G, v, u)$$

$$(\psi 2) \quad \psi(G^g, u^g, v^g) = \psi(G, u, v) \text{ za } g \in S_n$$

Onda možemo definisati $\tilde{\psi}(G, u, W) = \sum_{v \in W} \{\psi(G, u, v)\}$ pri čemu sada sumu interpretiramo kao sabiranje multiskupova.

Značaj uvedenog pojma dolazi iz činjenice da je dovoljno jednom, pre izvršavanja algoritma, za uneti graf izračunati odabranu invarijantu. To znači da je jedino povećanje u ukupnoj složenosti celog postupka cena tog pretprocesiranja. Nakon toga, ukoliko je odabrana pogodna invarijanta, moguća je velika ušteda usled moćnije procedure profinjavanja i manjeg stepena grananja stabla pretrage.

Primer 22. *Posmatrajmo (nepovezan) graf sa slike 3.6. Standardna procedura profinjavanja određuje najgrublje bojenje finije od početnog i ekvitabilno u odnosu matricu povezanosti kao invarijantu grafa. Ona će kao rezultat vratiti bojenje grafa jednako početnom zato što je početno bojenje grafa ekvitabilno u odnosu na tu invarijantu - svaki čvor grafa je povezan sa tačno dva čvora iste boje. Procedura profinjavanja koja koristi invarijantu grafa $d(G, u, v)$ koja predstavlja dužinu najkraćeg puta između čvorova u i v (ili ∞ ako put ne postoji) će odrediti bojenje čije su ćelije tačno orbite čvorova u grupi automorfizama.*



Slika 3.6: Levo je prikazana particija usled standardne procedure profinjavanja. Količnički graf sadrži jedan čvor c i važi $\psi_Q(c, c) = 2$. Desno je prikazana particija usled profinjavanja sa invarijantom najkraćeg puta. Količnički graf sadrži tri čvora a , b i c i važi $\psi_Q(a, a) = \{0, 1, 1\}$, $\psi_Q(b, b) = \{0, 1, 1, 2\}$ i $\psi_Q(c, c) = \{0, 1, 1, 2, 2\}$, dok su $\psi_Q(x, y)$ za različite x i y multiskupovi sa različitim brojem beskonačnosti.

Napomenimo da u radu [7] i detaljnije u [8] postoji uveden sličan pojam invarijante grafa koji svakom čvoru (nasuprot parovima čvorova u našoj definiciji) obojenog grafa dodeljuje neku vrednost. Ovake invarijante se mogu koristiti za dodatno profinjavanje bojenja, najčešće u prvih nekoliko nivoa stabla pretrage. Ističemo činjenicu da njima nije moguće efektivno pojačati samu proceduru profinjavanja.

Glava 4

Rezultati

U ovoj glavi prikazani su rezultati vremena izvršavanja implementacije algoritma na raznim klasama grafova. Algoritam je implementiran u programskom jeziku C++ u okviru programa *morphi*. Izvorni kod je dostupan na adresi <https://github.com/idrecun/morphi>. Testiranje je vršeno na grafovima iz kolekcije programa *bliss* i na nekoliko klasa nasumično generisanih grafova. Svi grafovi preuzeti su sa <https://pallini.di.uniroma1.it/Graphs.html>.

4.1 Detalji testiranja

U rezultatima prikazano je vreme izvršavanja programa mereno komandom *time* ugrađenom u *Bash*, kao zbir vremena koje je program proveo u korisničkom i sistemskom režimu. Granica za vreme izvršavanja je u svim primerima postavljena na 100 sekundi. Testiranje je vršeno na HP Spectre x360 laptop računaru sa Intel i5 procesorom. U trenutku testiranja implementacija nije podržavala izvršavanje na više procesorskih jezgara.

Testirane su dve varijante algoritma. U prvoj je za invarijantu grafa uzeta matrica povezanosti, dok je u drugoj korišćena invarijanta najkraćeg puta iz primera 22. Poređenja radi su prikazani rezultati vremena izvršavanja programa *nauty* (na mestima gde je značajno i sa upotrebom invarijante najkraćeg puta opisane u [8]) i *Traces* na istim primerima. Za *nauty* su za svaku klasu grafova prikazani rezultati za onu reprezentaciju grafa (normalna ili proređena) koja je dala bolje rezultate. Horizontalna osa grafika predstavlja broj čvorova grafa, a vertikalna osa vreme izvršavanja u sekundama (na logaritamskoj skali).

4.2 Nasumično generisani grafovi

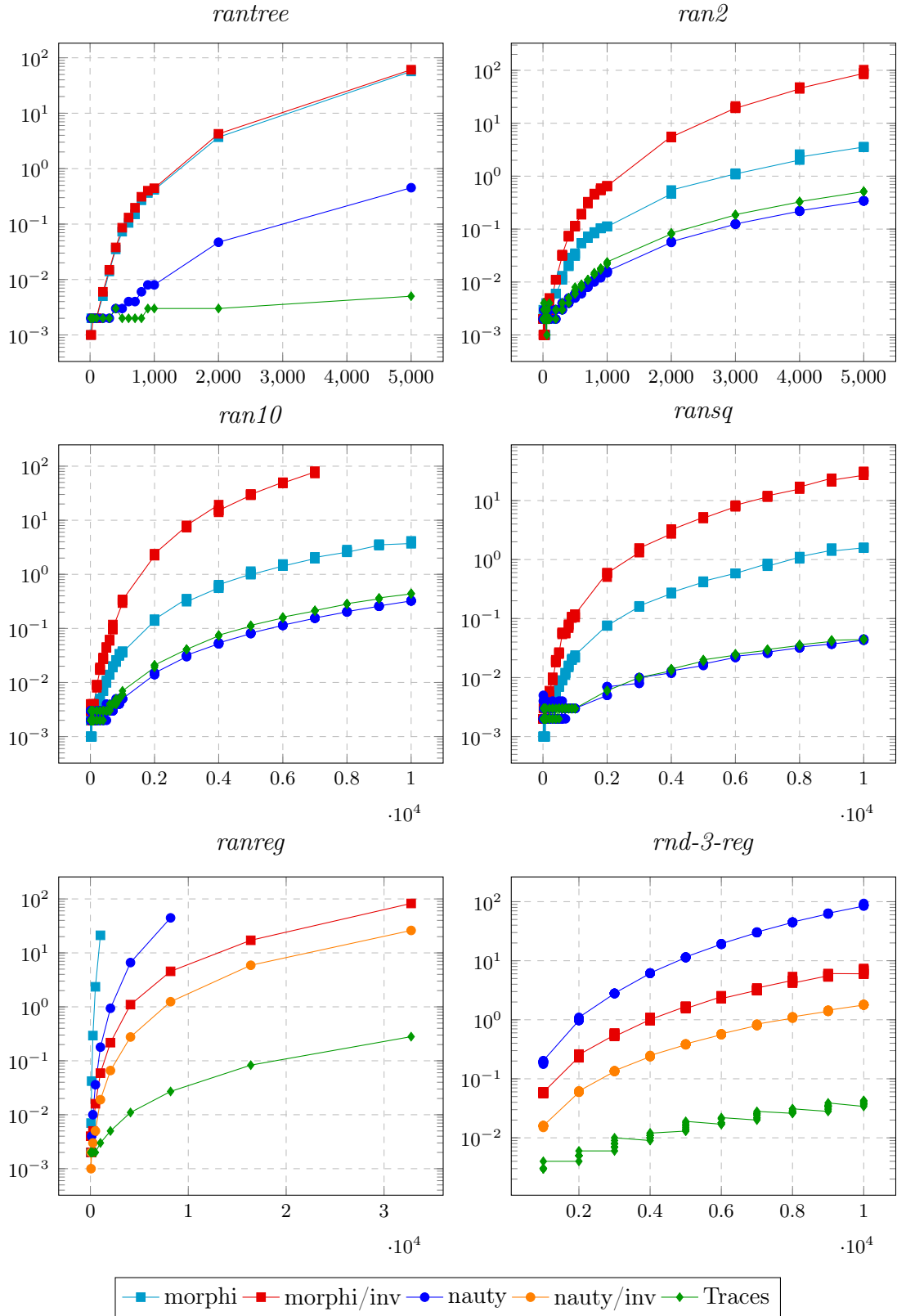
U ovom odeljku su prikazani rezultati testiranja na klasama nasumično generisanih grafova navedenih u nastavku. Rezultati za klasu *rnd-3-reg* bez upotrebe invarijante razdaljine nisu prikazani zato što za većinu primera program nije prekinuo izvršavanje pre granice od 100 sekundi.

- *rantree* - Stabla
- *ran2* - Grafovi sa verovatnoćom grane $\frac{1}{2}$
- *ran10* - Grafovi sa verovatnoćom grane $\frac{1}{10}$
- *ransq* - Grafovi sa verovatnoćom grane $\frac{1}{\sqrt{n}}$ za n čvorova
- *ranreg* - Regularni grafovi
- *rnd-3-reg* - 3-regularni grafovi (iz *bliss* kolekcije)

U klasama slučajnih grafova bez strukture *ran2*, *ran10* i *ransq* složenost preprocesiranja invarijante grafa najkraćeg puta dominira. Pojačanje procedure profinjavanja je za ove primere zanemarljivo - na skoro svim primerima se i bez upotrebe dodatne invarijante grafa generiše samo jedan čvor stabla pretrage. Upotreba invarijante je nepotrebna i u slučaju nasumično generisanih stabala *rantree*. Složenost algoritma na primerima ove klase proizilazi pretežno iz pretrage.

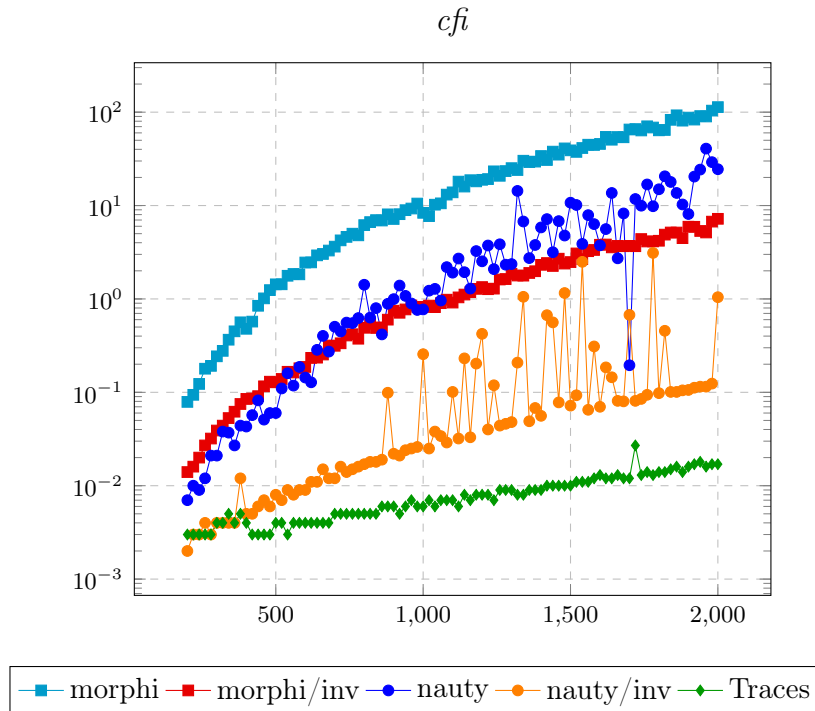
Slučaj klasa regularnih grafova *ranreg* i *rnd-3-reg* je drugačiji. Osnovni algoritam se na njima pokazao izuzetno neefikasnim. Upotrebom invarijante najkraćeg puta se ova neefikasnost zaobilazi zbog toga što pojačanje procedure profinjavanja dovodi do generisanja stabla pretrage koje sadrži jedan čvor. Rezultat ovoga je da vreme izvršavanja nije značajno sporije od vremena izvršavanja programa *nauty*.

GLAVA 4. REZULTATI



4.3 Grafovi donje granice nekih algoritama

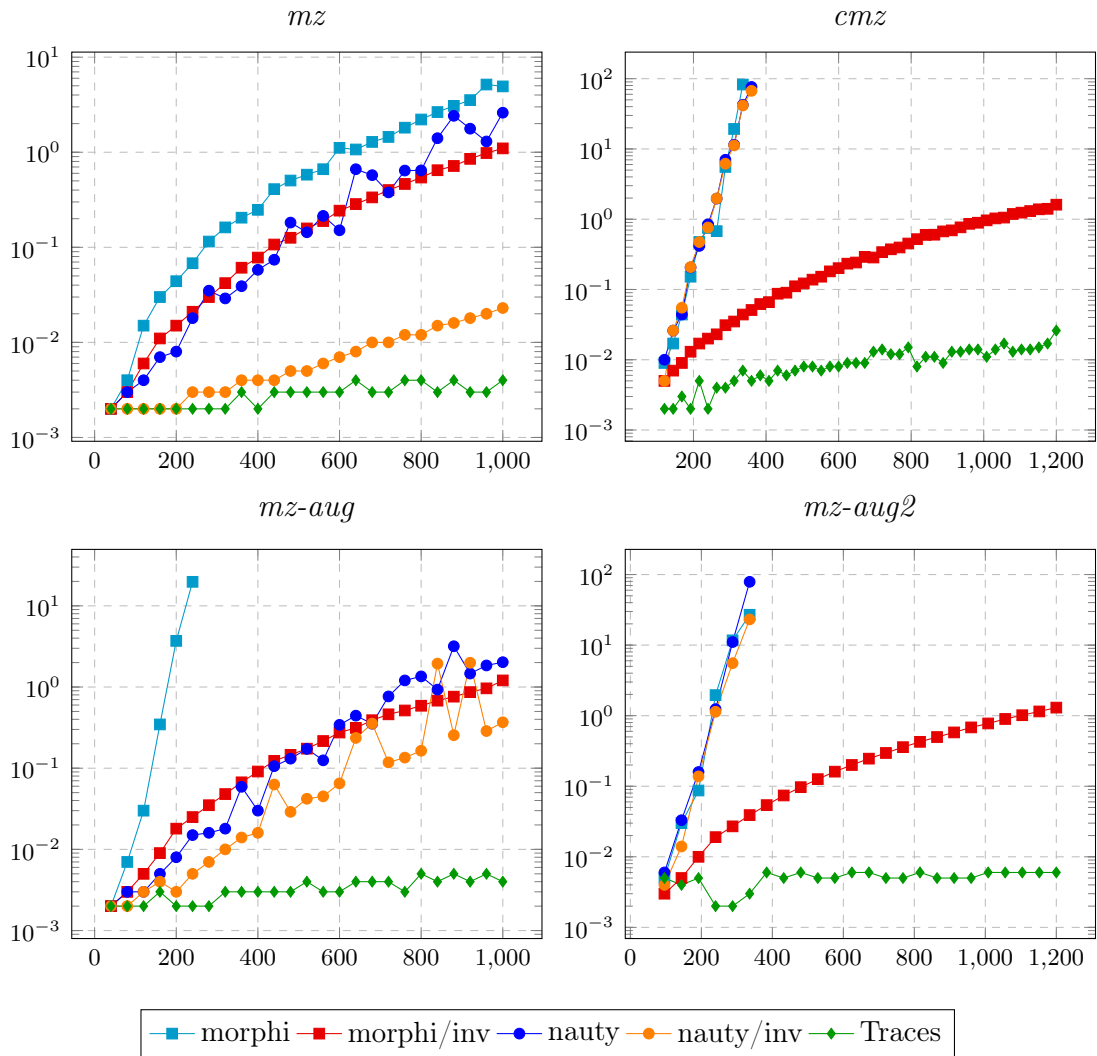
Jedno uopštenje klasifikacije čvorova procedurom profinjavanja je *k-dimenzioni WL-metod* koji služi za klasifikaciju uređenih k -torki čvorova grafa i koji se može koristiti za ispitivanje izomorfizma dva grafa. Klasa *cfi* iz *bliss* kolekcije je klasa grafova opisanih kao težak primer za k -dimenzioni WL-metod po tome da je neophodno odabrati $k = \Omega(n)$ kako bi korektnost ispitivanja izomorfizma para grafova sa n čvorova bila zagarantovana [2]. Rezultati pokazuju da je upotrebom invarijante najkraćeg puta moguće ostvariti vreme izvršavanja koje je u većini slučajeva za nijansu brže od *nauty*-a bez upotrebe invarijante, ali idalje sporije od drugih varijanti programa *nauty* i *Traces*.



Klase *mz*, *cmz*, *mz-aug*, *mz-aug2* su zasnovane na klasi *Mijazakijevih grafova* opisanih kao teški primeri za Mekejev algoritam [9] koji je centralna tema ovog rada. Osnovni algoritam implementiran u okviru *morphi*-a se na primerima ovih klasa ponaša slično *nauty*-u, izuzev u klasi *mz-aug* gde je složenost algoritma znatno veća. Na ovim primerima do izražaja dolazi pojačanje procedure profinjavanja usled upotrebe invarijante najkraćeg puta, posebno u klasama *cmz* i *mz-aug2* gde obe varijante *nauty*-a pokazuju eksponencijalno ponašanje koje *morphi* izbegava. Ovo nije neočekivano pošto su *Mijazakijevi* grafovi težak primer za upravo onaj algoritam

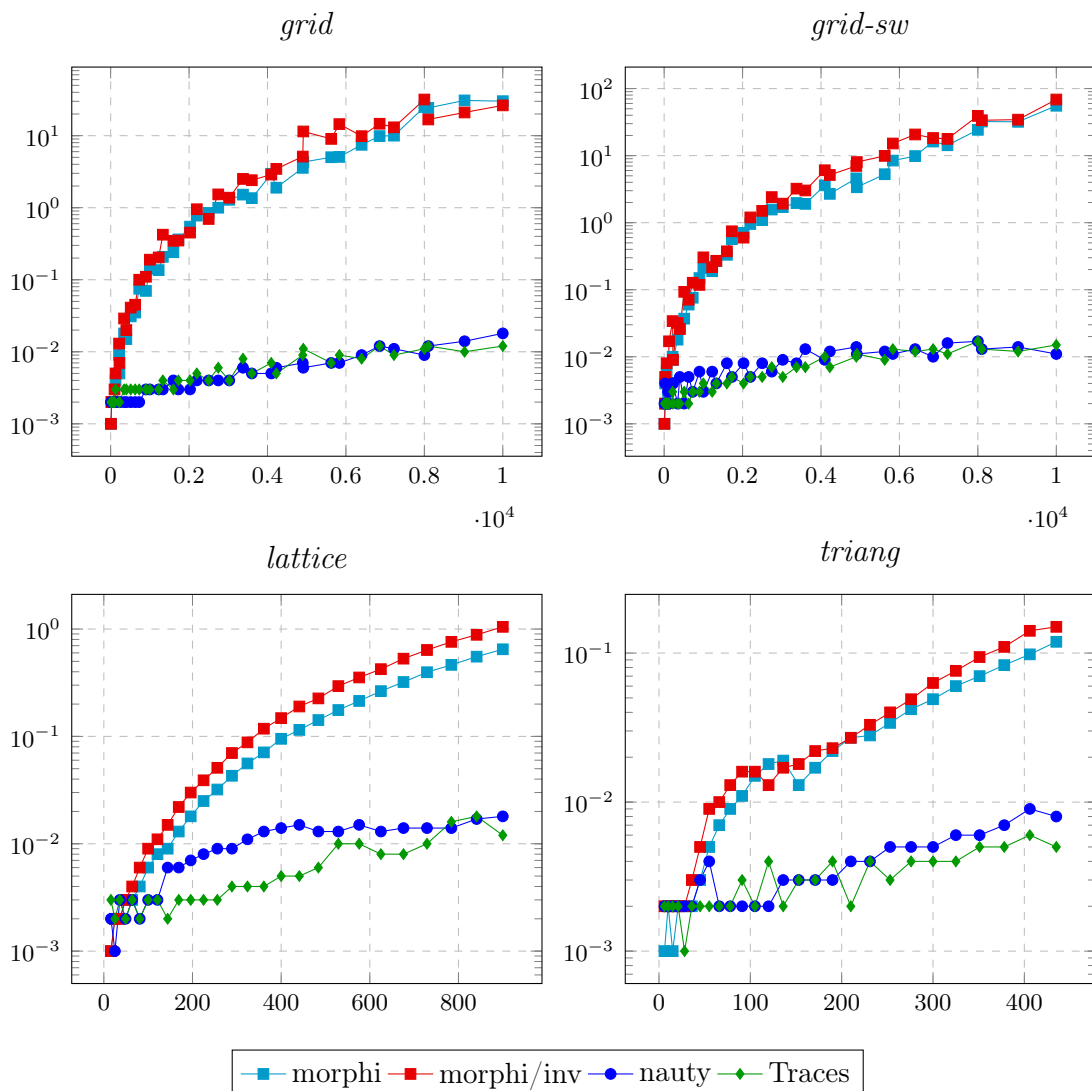
GLAVA 4. REZULTATI

koji *nauty* i osnovna verzija *morphi*-a implementiraju i pošto dodatna invarijanta menja taj algoritam na ključnom mestu.



4.4 Mreže

Grafovi klase *grid* su grafovi koordinatnih mreža u dve i tri dimenzije. Grafovi klase *grid-sw* su isti ti grafovi sa razmenjenim parovima nekih grana. *triang* predstavlja klasu trougaonih grafova. Još mreža nalazi se u klasi *lattice*. Rezultati pokazuju da nema značajne razlike između dve varijante *morphi*-a. Programi *nauty* i *Traces* su na primerima ove klase znatno brži. Jedan od mogućih razloga za to je što *morphi* u trenutku testiranja nije podržavao odvojenu strukturu podataka za rad sa proređenim grafovima, koja je za ove klase korišćena u *nauty*-u.



4.5 Grafovi nekih kombinatornih objekata

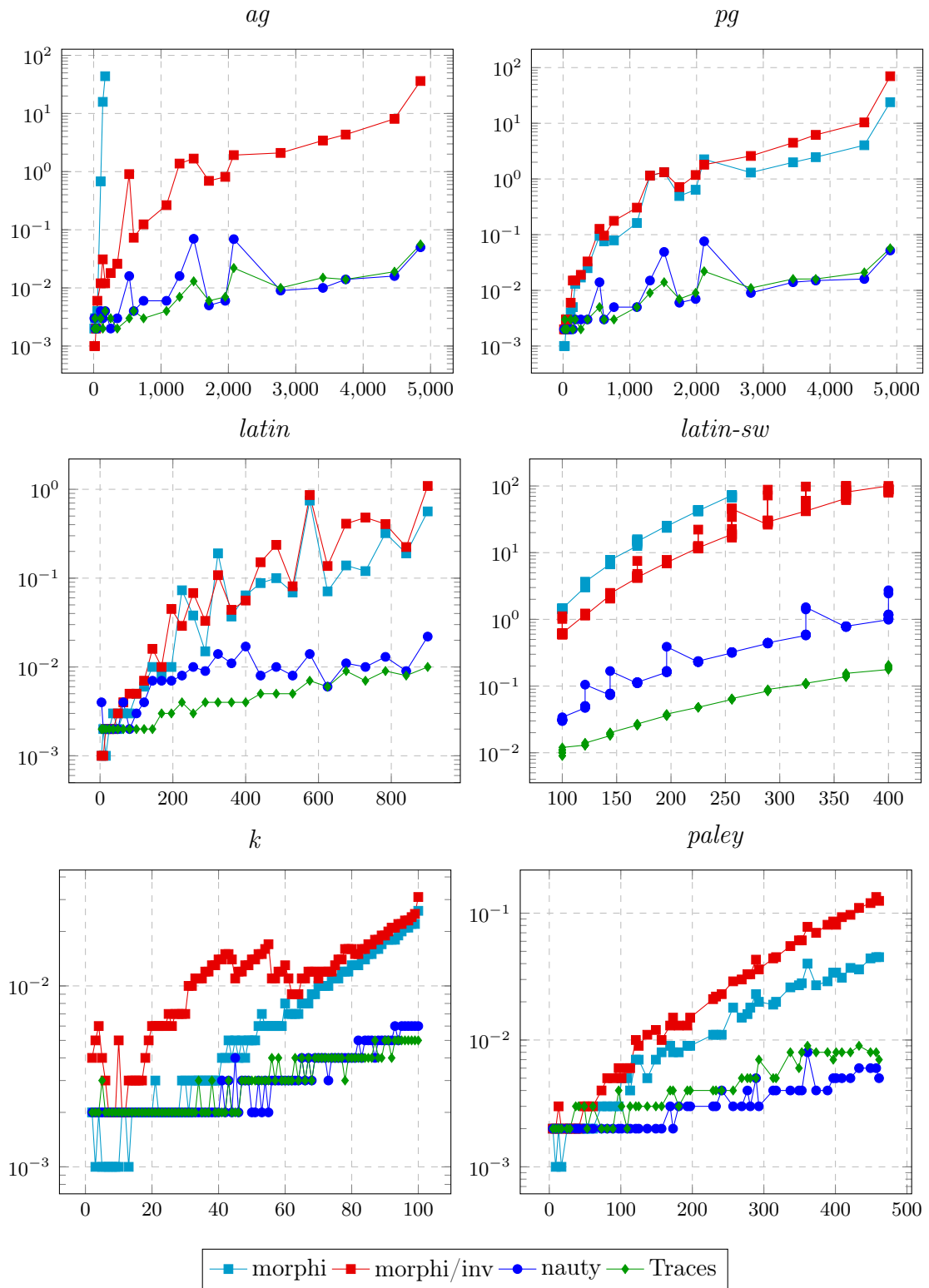
U ovom odeljku su prikazani rezultati testiranja na klasama grafova iz *bliss* kolekcije navedenih u nastavku.

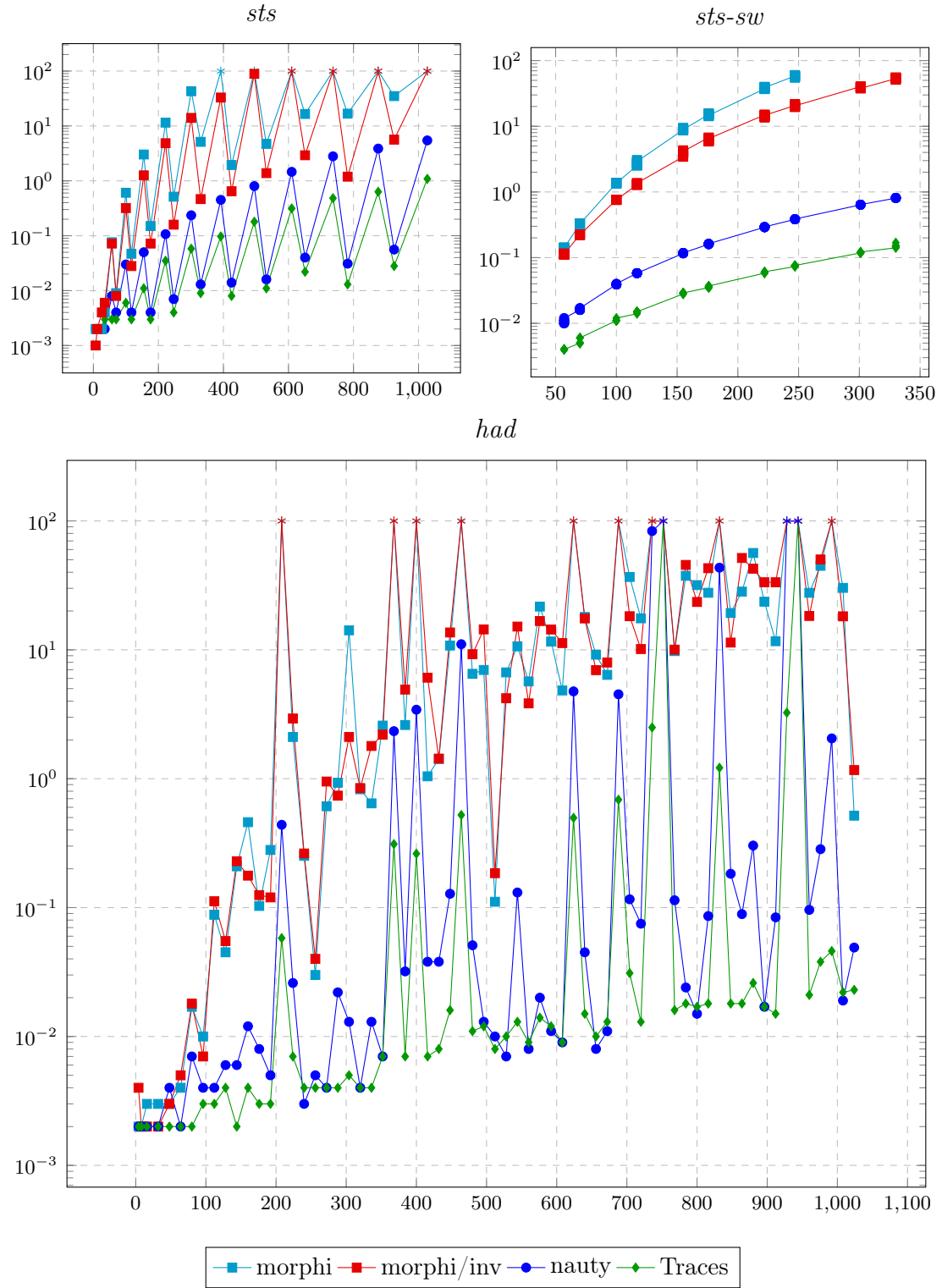
- *ag* - Grafovi afine geometrije
- *had* - Grafovi Adamarovih matrica
- *latin* - Grafovi latinskih kvadrata
- *latin-sw* - Grafovi latinskih kvadrata sa razmenjenim parovima grana
- *k* - Kompletni grafovi
- *paley* - Pejljevi grafovi
- *pg* - Grafovi Dezargovih ravni
- *sts* - Grafovi Štajnerovih sistema trojki
- *sts-sw* - Grafovi Štajnerovih sistema trojki sa razmenjenim parovima grana

Pored navedenih klasa, u *bliss* kolekciji postoje i klase *had-sw*, *kef* i *pp*. Rezultati za klasu *had-sw* nisu prikazani zato što za većinu primera program nije prekinuo izvršavanje pre granice od 100 sekundi. Rezultati za klase obojenih grafova *kef* i *pp* nisu prikazani zato što prilikom testiranja implementacija nije podržavala unošenje obojenih grafova.

Upotreba invarijante najkraćeg puta ima različite posledice na vreme izvršavanja. Na primerima klase *pg* je usporenje najosetnije, dok se najznačajnije ubrzanje postiže za primere klase *ag*. Programi *nauty* i *Traces* na velikom broju klasa postižu očekivano bolje rezultate i ostavljaju prostor za poboljšanje implementacije *morphi-a*.

GLAVA 4. REZULTATI





Glava 5

Zaključak i dalji rad

Cilj ovog rada bio je izučavanje i opis savremenog okvira za konstrukciju algoritma za ispitivanje izomorfizma grafova. Konstruisan je i implementiran algoritam zasnovan na tom okviru i prikazani su rezultati vremena izvršavanja implementiranog algoritma na reprezentativnom skupu ulaza. Uvedeno je uopštenje algoritma variranjem korišćene invarijante grafa. Rezultati testiranja pokazali su da algoritam može biti značajno optimizovan na nekim klasama grafova korišćenjem tog uopštenja.

Postoji više mogućih pravaca za dalji rad. Jedan od njih podrazumeva analizu i implementaciju efikasnijih struktura podataka i dodatnih optimizacija sa svrhom poboljšanja vremena izvršavanja programa. Jedna optimizacija koja u radu nije obrađena je mogućnost otkrivanja automorfizama u unutrašnjim čvorovima stabla pretrage na osnovu specifičnih svojstava bojenja. Još jedna optimizacija vezana za automorfizme je određivanje orbita u datom čvoru stabla. Dok smo u radu razmatrali samo podskup automorfizama koji stabilizuju niz individualizovanih čvorova, moguće je koristiti pametniju strukturu podataka poput Šrajer-Sims metoda ili njegove probabalističke varijante [12].

Još jedna oblast od interesa za dalje istraživanje je uvedeni pojam invarijante grafa. Pored toga što je moguće konstruisati invarijantu grafa koja daje jednaku ili bolju proceduru profinjavanja, postoji još pitanja kojima se u radu nismo bavili. Kako je cilj profinjavanja particionisanje čvorova u klase što približnije orbitama čvorova, nameće se pitanje u kojoj meri je ovo moguće izvesti u opštem slučaju, odnosno da li postoji invarijanta grafa koja proizvoljan graf particioniše tačno u orbite ili makar u particiju koja se u odnosu na particiju orbita razlikuje na neki odozgo ograničen način. Slično pitanje moguće je postaviti i za određene klase gra-

fova, kao što je na primeru prikazano da 2-regularne grafove u potpunosti rešava invarijanta razdaljine čvorova. Još jedno srodno, možda suprotno pitanje je da li postoji klasa grafova na kojoj proizvoljna invarijanta grafa nije značajno bolja od matrice povezanosti.

Iako je problem vremenske složenosti sa praktičnog stanovišta rešen, ono što ostaje kao otvoren problem je korektnost implementacije i verifikabilnost rezultata. Postoje dva moguća pristupa ovom problemu. Jedan podrazumeva implementaciju algoritma i formalan dokaz korektnosti te implementacije. Mana ovog pristupa je efikasnost ovakvih implementacija koje mogu biti znatno sporije u odnosu na implementacije razvijene u nekom nižem jeziku. Drugi pristup podrazumeva generisanje sertifikata uz rezultat izvršavanja efikasno implementiranog programa. Sertifikat bi po završetku izvršavanja programa mogao biti analiziran verifikatorom (programom čija je korektnost dokazana) koji bi sertifikat koristio kao svedoka da dokaže da je izlaz programa korektan. Cilj je koncipirati sertifikat na taj način da implementacija verifikatora, iako pisana u višem programskom jeziku koji omogućava formalan dokaz korektnosti, bude efikasnija od implementacije samog rešavača. Na primer, program za ispitivanje izomorfizma dva grafa kao sertifikat u slučaju da su grafovi izomorfni može koristiti pronađen izomorfizam, dok slučaj neizomorfni grafova nije na isti način trivijalan.

Bibliografija

- [1] László Babai. Graph Isomorphism in Quasipolynomial Time. *CoRR*, abs/1512.03547, 2015.
- [2] Jin-Yi Cai, Martin Fürer, and Neil Immerman. An optimal lower bound on the number of variables for graph identification. *Combinatorica*, 12:389–410, 1992.
- [3] Paolo Codenotti, Hadi Katebi, Karem Sakallah, and Igor Markov. Conflict Analysis and Branching Heuristics in the Search for Graph Automorphisms. pages 907–914, 11 2013.
- [4] Tommi Junttila and Petteri Kaski. Engineering an efficient canonical labeling tool for large and sparse graphs. In David Applegate, Gerth Stølting Brodal, Daniel Panario, and Robert Sedgewick, editors, *Proceedings of the Ninth Workshop on Algorithm Engineering and Experiments and the Fourth Workshop on Analytic Algorithms and Combinatorics*, pages 135–149. SIAM, 2007.
- [5] José Luis López-Presa, Antonio Fernández Anta, and Luis Núñez Chiroque. Conauto-2.0: Fast Isomorphism Testing and Automorphism Group Computation. *CoRR*, abs/1108.1060, 2011.
- [6] Brendan D. McKay. Practical Graph Isomorphism. *Congressus Numerantium*, 30:45–87, 1981.
- [7] Brendan D. McKay and Adolfo Piperno. Practical Graph Isomorphism, II. *Journal of Symbolic Computation*, 60:94–112, 2014.
- [8] Brendan D. McKay and Adolfo Piperno. nauty and Traces User’s Guide (Version 2.7), 2019.
- [9] Takunari Miyazaki. The complexity of McKay’s canonical labeling algorithm. In *Groups and Computation*, 1995.

BIBLIOGRAFIJA

- [10] Daniel Neuen and Pascal Schweitzer. Benchmark Graphs for Practical Graph Isomorphism. *CoRR*, abs/1705.03686, 2017.
- [11] Adolfo Piperno. Search Space Contraction in Canonical Labeling of Graphs. *CoRR*, abs/0804.4881, 2008.
- [12] Ákos Seress. *Permutation Group Algorithms*. Cambridge Tracts in Mathematics. Cambridge University Press, 2003.