Sveučilište Jurja Dobrile u Puli

Tehnički fakultet u Puli

EMILI PUH

ALGORITMI ZA ODRŽIVE TOPOLOGIJE SUSTAVA

Završni rad

Pula, rujan, 2024. godine

Sveučilište Jurja Dobrile u Puli

Tehnički fakultet u Puli

EMILI PUH

ALGORITMI ZA ODRŽIVE TOPOLOGIJE SUSTAVA

Završni rad

JMBAG: 0303098021, redoviti student/ica

Studijski smjer: Računarstvo

Predmet: Strukture podataka i algoritmi, Matematika 1, Matematika 2

Znanstveno područje: 1. Prirodne znanosti, 2. Tehničke znanosti

Znanstveno polje: 1.01. Matematika, 2.09. Računarstvo

Znanstvena grana: 1.01.02. geometrija i topologija, 2.09.03. obrada informacija, 2.09.06. programsko inženjerstvo

Mentori: Tihana Galinac Grbac, Neven Grbac

Pula, rujan, 2024. godine

Tehnički fakultet u Puli

Ime i prezime studenta/ice Emili Puh

JMBAG 0303098021

Status: ☒ redoviti ☐ izvanredni

PRIJAVA TEME ZAVRŠNOG RADA

Tihana Galinac Grbac, Neven Grbac

Ime i prezime mentora

Računarstvo

Studij

Strukture podataka i algoritmi, Matematika 1, Matematika 2

Kolegij

Potvrđujem da sam prihvatio/la temu završnog/diplomskog rada pod naslovom:

Algoritmi za održive topologije sustava

(na hrvatskom jeziku)

Algorithms for Sustainable System Topologies

(na engleskom jeziku)

Datum: 21.3.2023.

Sadržaj

1. Uvod

Tema ovog završnog rada je algoritmi za održive topologije sustava. Općenito, glavni cilj ovog rada je dati uvid u topološku analizu podataka i kako se ona može primjenjivati u strukturama softvera. Detaljnije, ciljevi ovog rada su razumijevanje specifičnih izazova modeliranja i upravljanja softverskim sustavima, razumijevanje potreba i koristi generaliziranih pristupa za modeliranje softvera i upravljanje istim, razumijevanje načina na koji se topološke strukture podataka mogu primjenjivati te razumijevanje mogućnosti izgradnje održive softverske strukture.

Rad je podijeljen u nekoliko poglavlja. Uvodni dio označuje poglavlje pod brojem 2 u kojem se raspravlja o definicijama dizajna i upravljanja složenim sustavima. U poglavlju pod brojem 3 motivira se i uvodi u topološku analizu podataka. Poglavlje pod brojem četiri uvodi u same topološke algoritme te se na kraju, u poglavlju 5 zaključuje rad.

Tijekom rada na projektu, moj je zadatak bio provesti analize nad određenim grafovima predstavljenim kao setovima podataka koristeći algoritam za topološku analizu podataka i taj algoritam implementirati u Matlabu te sistematizirati dobivene rezultate. Grafovi, odnosno podaci nad kojima je provedena analiza su JDT(Java Development Tools), PDE(Plug-in Development Environment) i BIRT(Business Inteligence and Reporting Tools), međutim moj posao je među ostalim bio analiza BIRT skupa podataka. Projekt JDT u sebi sadrži 14 setova podataka, PDE 13 te BIRT 9 setova podataka. Analiza je provedena na način da se u Matlab učitalo po jedan set podataka od svakog projekta te se nad tim određenim setom podataka pomoću definirane formule provela analiza i dobili su se traženi rezultati. Glavni cilj ove analize bilo je zapravo dobiti uvid u razne defekte podataka, a među ostalim bilo je potrebno dobiti uvid koliko se u nekom grafu nalazi čvorova, veza, trokuta i tetraedara. Svaki graf je imao svoju funkciju po kojoj se dobivao broj navedenih geometrijskih podataka.

Rezultati ovog završnog rada dobiveni su i objavljeni u znanstvenom radu [37], a postupak analize je opisan u materijalima s ljetne škole [40], stoga ovaj završni rad ukratko opisuje što su glavni rezultati tih znanstvenih radova, kao i postupke i tehnike kojima su dobiveni.

2. Održive strukture softvera

Postoji nekoliko specifičnih izazova prilikom modeliranja i upravljanja održivim softverskim sustavima. Disciplina softverskog inženjerstva rješava probleme ponovne upotrebe sustava, neovisne evolucije dijelova sustava i razvoja generičkih softverskih funkcija koje postaju sve teže i skuplje kako se složenost softvera povećava. Posljedice loše dizajniranih softverskih rješenja mogu biti ozbiljne i utjecati na operativno ponašanje softvera.

Izazovi koji se nameću prilikom dizajniranja složenih sustava je teško razumijevanje, upravljanje, održavanje i razvijanje. Princip dizajniranja koji se koristi u disciplini softverskog inženjerstva jest modularni dizajn sustava koji cilja na razvoj sustava kao skupa labavo povezanih komponenti, s nekoliko razina apstrakcije, slojevitim dizajnom i jasnom hijerarhijom. Sukladno tome, svaki softverski sustav ima modularne strukture i sastoji se od modula, odnosno komponenti, i njihovih interakcija. Pojam modul odnosi se na komponentu sa standardnim i labavo povezanim sučeljima koji koriste drugi moduli unutar njezina okruženja. Moduli mogu predstavljati potprograme, funkcije, klase, objekte itd.

Sljedeći izazov koji se nameće u modernim softverskim sustavima je taj što takvi sustavi obično rade u globalno međusobno povezanim internetskim okruženjima. Struktura sustava se tada sastoji ne samo od mnogih komponenti koje se izvode unutar izoliranih računalnih čvorova, već je softverska struktura skup softverskih komponenti koje su međusobno povezane na geografski distribuiranoj internetskoj mreži, čime se u pitanje dovodi održivost mrežnih operacija i količina potrošnje enegije. Ovo su neki od izazova modeliranja i upravljanja održivim softverskim sustavima.

Da bi se odgovorilo na gore navedene izazove, disciplina softverskog inženjerstva razvila je razne generalizirane pristupe modeliranju i upravljanju softverom. Jedan široko korišten apstraktni artefakt je softverska struktura. Softverska struktura definira se kao skup modula i njihovih interakcija koje se koriste za postizanje globalne funkcionalnosti sustava. Jedan od najčešćih prikaza softverske strukture je korištenjem pozivnih grafova, tzv. *call graphs.* U praksi se prilikom proučavanja takvih grafova koristi i topološka analiza podataka, koja daje topološke informacije koje su u današnje vrijeme primjenjive u raznim domenama analize podataka, kao npr. u zdravstvu, neuroznanosti, epidemiologiji, klasifikaciji proteina i sl.

3. Topološka analiza podataka (TDA)

Topologija je grana matematike koja se bavi kvalitativnim geometrijskim informacijama. Ova disciplina, ukratko, proučava oblik i formu geometrijskog objekta i zanemaruje veličinu i položaj. Ova metrička neosjetljivost može biti korisna u situacijama gdje se metrika shvaća samo na „grub“ način.
Topološka analiza podataka odnosi se na proučavanje podataka korištenjem topoloških metoda. Osobito je korisna za opisivanje i usporedbu oblika, otkrivanje trendova i traženje skrivenih obrazaca u podacima.

3.1. Topologija kao kvalitativni model realnosti

Ponekad u proučavanju nije važno imati precizne kvantitativne modele. U takvim slučajevima je interes u obliku, trendovima ili formi, što znači da su tada modeli stvarnosti kvalitativni. Kvalitativni modeli dolaze u dvije vrste: diskretni i kontinuirani. Uobičajeni primjer diskretnih modela su grafovi i oni se mogu koristiti za modeliranje veza između objekata.
Kroz prije definiranu definiciju topologije, može se reći da su dva objekta topološki ekvivalentna ako se mogu transformirati jedan u drugi rastezanjem i modeliranjem, ali bez rezanja ili lijepljenja. Primjer topološke ekvivalencije bili bi krafna i šalica zbog toga jer se, ako su napravljeni od gline, mogu modelirati jedan u drugi bez rezanja ili lijepljenja.
Topološki prostor je skup točaka za koje je definirana određena ideja „bliskosti“. To se postiže definiranjem skupa otvorenih susjedstva za svaku točku. Na primjer, realna linija je topološki prostor u kojem su susjedstva svake točke otvoreni intervali koji sadrže tu točku. Dva topološka prostora su topološki ekvivalentna ako postoji homeomorfizam između njih, a homeomorfizam je kontinuirana bijekcija između dva topološka prostora tako da je njezina inverzna funkcija također kontinuirana. Razmatranje i uspoređivanje topoloških prostora je važno u topološkoj analizi podataka jer sam topološki prostor sadrži informacije o osnovnim podacima, njihovoj strukturi i svojstvima.

3.2. Topološke invarijante kao aproksimacije

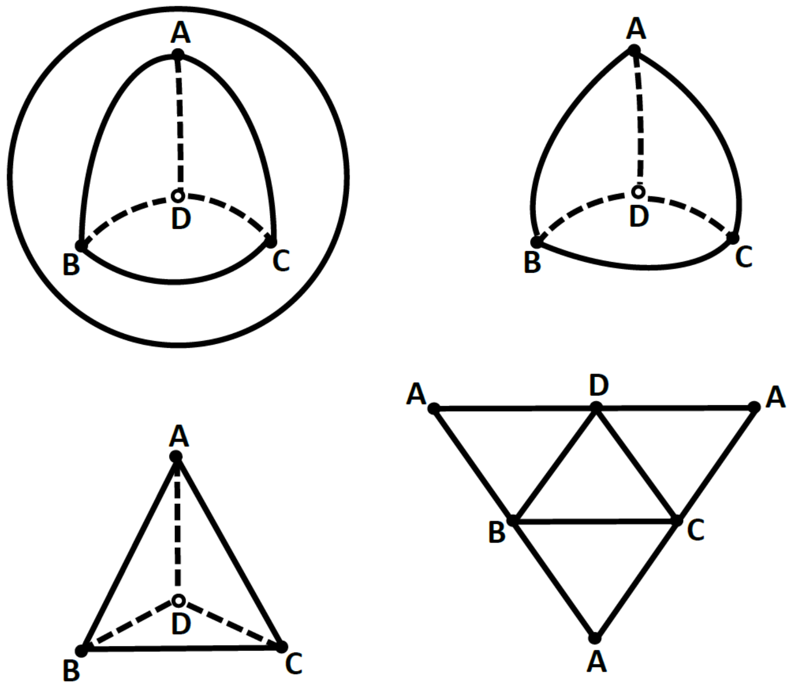
Topološke invarijante su određeni objekti pridruženi topološkim prostorima koji ostaju nepromijenjeni pod homeomorfizmom. Drugim riječima, topološki ekvivalentni prostori imaju iste topološke invarijante. Postoje mnoge topološke invarijante i većina njih je algebarske prirode. Iz tog se razloga proučavanje topoloških invarijanata često naziva algebarska topologija. Najčešće topološke invarijante su fundamentalna ili temeljna grupa, homotopija, homologija i kohomologija.

Za razliku od matematičkih analiza, topološki prostori nisu kvantitativni. Aproksimacije topoloških prostora trebale bi na neki način aproksimirati oblik i predstavljati istu ulogu kao što, npr. polinomi predstavljaju u matematičkim analizama. Algebarske topološke invarijante mogu se promatrati kao algebarski objekti koji aproksimiraju topološki prostor.

3.3. Od topoloških prostora do njegovih invarijanti

Prijelaz s topološkog prostora na njegove algebarske topološke invarijante obično zahtijeva neku vrstu diskretnog modela kontinuiranog objekta poput topološkog prostora. To se postiže korištenjem tzv. triangulacije. Topološki prostor se opisuje korištenjem „kostura", koji je sličan grafu koji opisuje prostor, ali također sadrži i dijelove viših dimenzija (ne samo vrhove i bridove). Takav kostur naziva se simplicijalni kompleks i on je generalizacija grafa. Dakle, trijangulacija simplicijalnim kompleksom je generalizacija grafova u kojima su prisutni i dijelovi viših dimenzija, a ti dijelovi se nazivaju simpleksi. U duhu teorije grafova, d-dimenzionalni simpleks jednostavno je skup od d+1 vrhova (točaka), zajedno sa svim nižedimenzionalnim podsimpleksima, tj. nepraznim podskupovima. Brid grafa može se promatrati kao skup od dvije točke, zajedno s njegovim krajnjim točkama promatranim kao skupovima s jednom točkom, tako da je to 1-dimenzionalni simpleks. Vrhovi grafa su 0-dimenzionalni simpleksi. Trokut, zajedno sa svojim stranicama i vrhovima, može se promatrati kao 2-dimenzionalni simpleks. Slično, tetraedar je 3-dimenzionalni simpleks. Bilo koji (konačni) skup simpleksa, promatran kao jedan objekt, je simplicijalni kompleks. Bit je u tome da se bilo koji topološki prostor može triangulirati simplicijalnim kompleksom, što se može promatrati kao diskretizacija topološkog prostora.

Primjer trijangulacije vidljiv je na slici 1. Predstavljeni primjer prikazuje trijangulaciju 2-sfere. Trijagulacija 2-sfere je prikazana na samoj sferi, izdvojena iz sfere i sa ravnim bridovima, ali isto tako i na ravninskom dijagramu u kojem su identificirane točke i segmenti sa istim nazivima.



Slika I Trijangulacija 2-sfere

3.4. Topološka analiza softvera

Proučavanje softvera koristeći grafove je stara i široko rasprostranjena ideja. Pokazalo se vrlo korisnim u svim fazama životnog ciklusa softvera, od dizajna, testiranja i verifikacije do održavanja. Svaki (ne-trivijalni) softverski sustav sastoji se od komponenti koje, ovisno o kontekstu, mogu biti moduli, klase, objekti i sl. Komponente se mogu promatrati kao vrhovi grafa, a veze između komponenti kao bridovi grafa. Zajedno formiraju graf koji predstavlja softverski sustav. Međutim, grafički prikaz softvera razumije samo jednodimenzionalne odnose između softverskih komponenti, a da bi se mogao dobiti kvalitetniji opis softverske strukture potrebno je razmotriti višedimenzionalne odnose.

Dakle, pri proučavanju softverske strukture sustava i njegovih komponenti potrebno je pronaći grupe komponenti povezane zajedno. Takve grupe mogu se promatrati kao simpleksi, što daje simplicijalni kompleks softverskog sustava. Simplicijalni kompleks je, kao što je prije definirano, diskretizacija nekog topološkog prostora. Topološke invarijante tog prostora mogu se promatrati kao topološke invarijante softvera. Invarijante se mogu promatrati kao aproksimacije softverske strukture. Ključna točka ovog pristupa strukturi softverskog sustava je da se struktura aproksimira topološkim invarijantama dobivenim iz simplicijalnog kompleksa softverskog sustava. Topološki pristup zanemaruje veličinu sustava i koncentrira se samo na njegov oblik i samim time topološka analiza može otkriti skrivene sličnosti i razlike u strukturama softverskih sustava.

4. Topološki algoritmi

Algoritam koji je prestavljen u ovom poglavlju je najosnovniji algoritam računalne topologije. Cilj ovog poglavlja je uvođenje u samo područje topološke analize podataka. Ovo je osnovni korak u proučavanju računalne algebarske topologije, posebno homologije i kohomologije među ostalima.

Glavni pojmovi uvedeni u ovom poglavlju su osnovne ideje homološke algebre. Krajnji cilj je dati uvod u Bettijeve brojeve, tj. rangove homoloških grupa povezanih s simplicijalnim kompleksom. Radi jednostavnosti izlaganja, osnovno polje je $\mathbb{Z}_2 = \{0, 1\}$ od dva elementa s operacijama zbrajanja i množenja modulo dva. Pojmovi koji se koriste u daljnjem tekstu poglavlja sažeti su u tablici 1.

Važnost homoloških grupa i pridruženih Bettijevih brojeva leži u činjenici da oni kodiraju određene topološke informacije o prostoru, ili u ovom slučaju, o softverskom grafu. Te informacije su uglavnom povezane s povezanošću topološkog prostora čiji je „kostur" softverski graf o kojem je riječ. Iako homološke grupe nad $\mathbb{Z}_2$ razmatrane u ovim poglavljima obuhvaćaju samo ograničene topološke informacije, one pružaju dobar uvod u temu topološke analize podataka.

Pojam	Simbol	Definicija
Simplicijalni kompleks	Kaligrafsko slovo K	Familija nepraznih skupova koja sadrži sve neprazne podskupove svojih članova
Simpleks	$\sigma, \tau, \ldots$	Članovi simplicijalnog kompleksa
p-simpleks		Simpleks dimenzije *p*, tj. sadrži *p*+1 elemenata
p-lanac	$c, d, \ldots$	Formalne sume simpleksa iste dimenzije *p*
Granični operator	∂	Linearni operator definiran na lancima kao lanac niže dimenzije dan kao formalni zbroj graničnih površina ili lica
p-granični operator	∂_p	Granični operator koji djeluje na *p*-lance
Ciklus	$z, x, y, \ldots$	Lanci za nultom granicom
Granice	$b, a, \ldots$	Lanci dobiveni kao granice lanaca više dimenzije
Rang *p*-ciklusa	z_p	Algoritam po bazi dva broja *p*-ciklusa
Rang *p*-granice	b_p	Algoritam po bazi dva broja *p*-granica
Betti broj	β_p	Nenegativni cijeli broj dobiven kao $\beta_p = z_p - b_p$

Tablica 1 Tablica osnovnih pojmova

4.1. Simplicijalni kompleks

Definicija simplicijalnog kompleksa je u suštini vrlo jednostavna. Simplicijalni kompleks je samo familija skupova, ali kad god je neki skup unutar tog skupa, tada su svi njegovi neprazni podskupovi također u istom skupu. Formalno, konačna familija *K* nepraznih konačnih skupova naziva se simplicijalnim kompleksom ako je svaki neprazni podskup *τ* bilo kojeg skupa *σ* u familiji *K* također član familije *K* (naziva se stranom *σ*), tj.

$$\text{ako je } \sigma \in K \text{ i } \emptyset \neq \tau \subseteq \sigma, \text{ onda } \tau \in K.$$

Skupovi u simplicijalnom kompleksu nazivaju se simpleksi. Elementi skupova u simplicijalnom kompleksu obično se nazivaju točkama. Drugim riječima, svaki simpleks sastoji se od točaka. Broj točaka u simpleksu određuje njegovu dimenziju koja se ponekad naziva i stupnjem simpleksa. Preciznije, simpleks s *p* + 1 točaka dimenzije *p* i obično se naziva *p*-simpleks. Dakle, *p*-simpleks je skup u *K* koji sadrži točno *p* + 1 točaka, tako da su

0-simpleksi samo točke,

1-simpleksi sadrže dvije točke (ponekad nazvane segmentima),

2-simpleksi sadrže tri točke (ponekad nazvane trokuti),

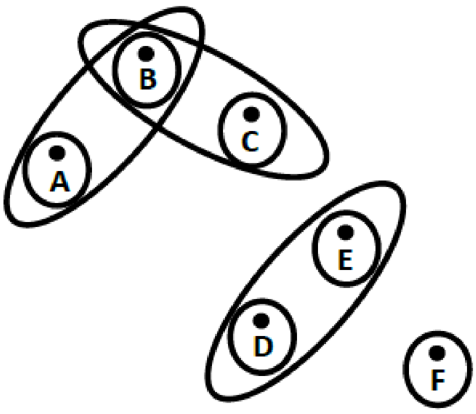
...

p-simpleksi sadrže *p* + 1 točaka.

Primjer jednog simplicijalnog kompleksa prikazan je na slici. Simplicijalni kompleks prikazan na slici sadrži 1-simplekse. Kao što je prikazano na slici, to je familija skupova

$$K = \{\{A, B\}, \{B, C\}, \{D, E\},$$
$$\{A\}, \{B\}, \{C\}, \{D\}, \{E\}, \{F\}\},$$

gdje su 1-simpleksi navedeni u prvom retku, a 0-simpleksi u drugom retku.

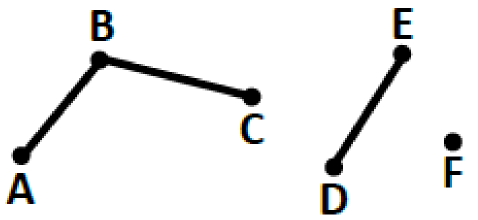


Slika II Simplicijalni kompleks

4.2. Geometrijska perspektiva simplicijalnog kompleksa

Simplicijalni kompleksi potječu iz geometrije, ili preciznije topologije, i mogu se interpretirati kao familija poliedara u dovoljno visokodimenzionalnom prostoru. Geometrijska perspektiva je način vizualizacije definicije simplicijalnog kompleksa temeljenog na skupovima, ali nije prikladna za računalne svrhe.

Simplicijalni kompleks sa slike 2 je prikazan kroz geometrijsku perspektivu na slici 3. Ovaj simplicijalni kompleks sadrži skupove od dvije točke koji su predstavljeni stvarnim geometrijskim segmentima. Njegovi 1-simpleksi predstavljeni su kao pravci, dok su 0-simpleksi točke.



Slika III Geometrijska perspektiva simplicijalnog kompleksa

4.3. Simplicijalni kompleks grafa softvera

Postoji nekoliko načina za dodijeljivanje simplicijalnog kompleksa grafu. U ovom radu objašnjen je najjednostavniji način, iako je ovaj pristup previše jednostavan za ozbiljne primjene. Međutim, cilj je objasniti topološki algoritam pa je ovaj primjer prikladan.

Neka je $G = \{V, E\}$ graf sa setom vrhova *V* i setom grana ili rubova *E*. Ovdje se razmatraju samo jednostavni grafovi, tj. *G* graf je neusmjeren, neponderiran ili bez težine, nema petlje i nema višestrukih grana. Petlja u grafu je grana koja ima iste krajnje točke, a višestruka grana označava postojanje više grana s istim krajnjim točkama. *P*-simpleks u grafu *G* definiran je kao bilo koji podgraf koji sadrži *p* + 1 vrhova, pri čemu je svaki par vrhova vezan rubom. Takav podgraf naziva se potpun graf s *p* + 1 vrhova. U primjena se takvi podgrafovi često nazivaju klikama (eng. *cliques*) u grafu. Preciznije, simpleksi simplicijalnog kompleksa povezanog sa danim grafom su sljedeći:

0-simpleksi su samo vrhovi (potpuni podgrafovi s jednim vrhom)

1-simpleksi su samo rubovi (potpuni podgrafovi sa dva vrha)

2-simpleksi su trokuti (potpuni podgrafovi sa tri vrha)

3-simpleksi su tetraedri (potpuni podgrafovi sa četiri vrha)

...

p-simpleksi su potpuni podgrafovi s *p* + 1 vrhova.

Potrebno je uočiti da su svi podgrafovi potpunog grafa također potpuni. Stoga se može zaključiti da se zaista dobio simplicijalni kompleks, jer su strane simpleksa zaista simpleksi.

4.4. Lanci

Motivacija za uvođenje lanaca i njihove adicije je omogućiti linearno-algebarski formalizam koji omogućuje razmatranje više simpleksa iste dimenzije kao jednog objekta. To će omogućiti proučavanje višedimenzionalnih topoloških struktura, a posebno Bettijevih brojeva, koristeći linearnu algebru.

Neka S_p označava familija svih *p*-simpleksa u simplicijalnom kompleksu *K*. Neka n_p bude broj *p*-simpleksa, tj. $n_p = |S_p|$ je kardinalnost S_p. Označuje se

$$S_p = \{\sigma_1, \sigma_2, \dots, \sigma_{n_p}\}$$

za *p*-simplekse u familiji *K*.

P-lanac u *K* je bilo koja podfamilija S_p. Drugim riječima, bilo koji odabir bilo kojeg broja *p*-simpleksa stvara *p*-lanac. Dopušten je čak i prazan odabir, u kojem nijedan od *p*-simpleksa nije odabran.

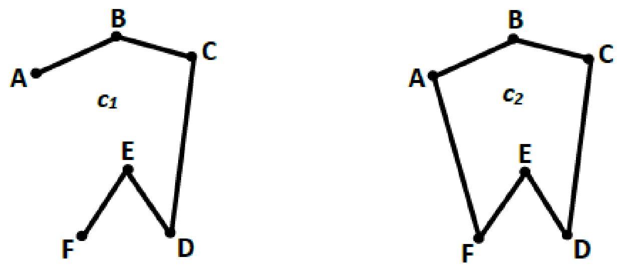
Primjer *p*-lanca prikazan je na slici 4. 1-lanac c_1 s lijeve stranice na slici sastoji se od 1-simpleksa $\overline{AB}$, $\overline{BC}, \overline{CD}, \overline{DE}$ i $\overline{EF}$ dok se 1-lanac c_2 s desne strane na slici sastoji od istih 1-simpleksa, uz dodatni 1-simpleks $\overline{FA}$. Postoji praktičan način za zapisivanje *p*-lanaca koristeći formalne sume *p*-simpleksa s koeficijentima u $\mathbb{Z}_2 = \{0, 1\}$. Koeficijenti kodiraju izbor *p*-simpleksa u *p*-lancu. Ako je koeficijent *p*-simpleksa nula, onda taj simpleks nije odabran u *p*-lancu, dok ako je koeficijent jedan, onda je odabran da bude u *p*-lancu. Preciznije, *p*-lanac c u K može se formalno pisati kao suma

$$c = \sum_{i=1}^{n_p} a_i \sigma_i,$$

gdje je $a_i \epsilon \mathbb{Z}_2 = \{0, 1\}$. Ako je koeficijent a_i = 0, onda σ_i nije dio *p*-lanca *c*, dok ako je a_i = 1, onda je σ_i unutar *p*-lanca *c*. 1-lanci na slici 4 mogu se prikazati kao

$$c_1 = \overline{AB} + \overline{BC} + \overline{CD} + \overline{DE} + \overline{EF}$$
$$c_2 = \overline{AB} + \overline{BC} + \overline{CD} + \overline{DE} + \overline{EF} + \overline{FA}$$

u formalnoj notaciji sume.



Slika IV Lanac

4.5. Operator ruba

Operator ruba je „ljepilo“ koje povezuje sve simplekse različitih dimenzija u simpleksnom kompleksu. Operator ruba *p*-tog reda dodjeljuje svakom *p*-simpleksu njegovu granicu, tj. skup njegovih *p* – 1 simpleksnih strana. Takav skup je *p* – 1 lanac. Za dani *p*-simpleks, djelovanje operatora ruba *p*-tog reda može se izraziti kao

$$\partial_p \sigma_i = \sum_{\substack{\tau \epsilon S_{p-1} \\ \tau \subseteq \sigma_i}} \tau \epsilon C_{p-1},$$

što je suma *p* – 1 simpleksa τ, i stoga *p* – 1 lanac u C_{p-1}.

Generalno, danim *p*-lancom

$$c = \sum_{i=1}^{n_p} a_i \sigma_i \epsilon C_p,$$

p-operator ruba ∂_p djeluje se kao

$$\partial_p c = \partial_p \Big(\sum_{i=1}^{n_p} a_i \sigma_i \Big)$$

$$= \sum_{i=1}^{n_p} a_i \partial_p \, \sigma_i \epsilon C_{p-1},$$

gdje

$$\partial_p \sigma_i = \sum_{\substack{\tau \epsilon S_{p-1} \\ \tau \subseteq \sigma_i}} \tau \epsilon C_{p-1},$$

kao gore navedeno. Dakle, ∂_p je preslikavanje iz C_p u C_{p-1}. Sada je jasnije iz definicije kako operator ruba povezuje informacije o simpleksima i lancima različitih dimenzija. Na primjer, 1-granični operator ∂_1 primijenjen na 1-lanac C_1 i C_2 na slici 4 računa se kao

$$\partial_1 c_1 = \partial_1 (\overline{AB} + \overline{BC} + \overline{CD} + \overline{DE} + \overline{EF})$$

$$= \partial_1 \overline{AB} + \partial_1 \overline{BC} + \partial_1 \overline{CD} + \partial_1 \overline{DE} + \partial_1 \overline{EF}$$

$$= (A + B) + (B + C) + (C + D) + (D + E) + (E + F)$$

$$= A + F$$

koji je 0-lanac dan kao suma 0-simpleksa A i F, dok je

$$\partial_1 c_2 = \partial_1 (\overline{AB} + \overline{BC} + \overline{CD} + \overline{DE} + \overline{EF} + \overline{FA})$$

$$= \partial_1 \overline{AB} + \partial_1 \overline{BC} + \partial_1 \overline{CD} + \partial_1 \overline{DE} + \partial_1 \overline{EF} + \partial_1 \overline{FA}$$

$$= (A + B) + (B + C) + (C + D) + (D + E) + (E + F) + (F + A)$$

$$= 0$$

što je prazni 0-simpleks označen s nulom. Operator ruba računa granicu, odnosno rub simpleksa u lancu izraženom kao suma preko $\mathbb{Z}_2$, tj. s dodatkom modulo 2. Operator ruba daje nulu ako simpleksi tvore ciklus, kao što je slučaj sa c_2.

4.6. Matrica operatora ruba

Pogodan način za promatranje operatora ruba je kao linearni operator na vektorskim prostorima lanca. U tom kontekstu može se uvesti matrica operatora ruba kao linearni operator.

P-lanci u C_p mogu se promatrati kao linearne kombinacije s koeficijentima u $\mathbb{Z}_2$= {0, 1} simpleksa u S_p. Stoga oni čine vektorski prostor nad poljem $\mathbb{Z}_2$ s dva elementa. Baza C_p kao vektorskog prostora nad $\mathbb{Z}_2$ je skup S_p svih *p*-simpleksa. Dakle, dimenzija C_p nad $\mathbb{Z}_2$ je broj $n_p = |S_p|$ *p*-simpleksa. Po samoj definiciji *p*-operatora ruba, on je linearni operator i C_p u C_{p-1} kao vektorske prostore nad $\mathbb{Z}_2$ jer poštuje linearne kombinacije

$$\partial_p\left(\sum_{i=1}^{n_p} a_i\sigma_i\right) = \sum_{i=1}^{n_p} a_i\partial_p\sigma_i$$

za bilo koji $a_i \epsilon\ \mathbb{Z}_2$.

Kao bilo koji linearni operator, *p*-operator ruba ∂_p može se prikazati pomoću matrice. Matrica *p*-operatora ruba ∂_p ima n_{p-1} redaka, jer je to dimenzija C_{p-1}, i n_p stupaca, jer je to dimenzija C_p. Ova matrica sadrži samo nule i jedinice, budući da su vektorski prostori nad $\mathbb{Z}_2$ = {0, 1}. Retci predstavljaju *p*-1 simplekse u S_{p-1}, a stupci predstavljaju *p*-simplekse u fiksom redoslijedu. *J*-ti stupac predstavlja *p*-simpleks $\sigma_j \epsilon S_p$. Sadrži jedinice u retcima koji predstavljaju njegove *p*-1 simplekse strane u S_{p-1}, a nule u retcima koji predstavljaju *p*-1 simplekse u S_{p-1} koji nisu njegove strane. Svaki *p*-simpleks ima točno *p*+1 strana, tako da svaki stupac sadrži točno *p*+1 jedinica. Preciznije, element na presjeku *i*-tog retka i *j*-tog stupca jednak je 1 ako je *i*-ti simpleks u S_{p-1} i *p*-1 simpleksna strana *j*-tog simpleksa u S_p, a jednak je 0 u suprotnom. Dakle,

$$\partial_p = \begin{bmatrix} d_{1,1} & \cdots & d_{1,n_p} \\ \vdots & d_{i,j} & \vdots \\ d_{n_{p-1},1} & \cdots & d_{n_{p-1},n_p} \end{bmatrix},$$

gdje

$$d_{i,j} = \begin{cases} 1, ako\ je\ i-ti\ p-1\ simpleks\ \tau_i \epsilon S_{p-1}\ lice\ j-tog\ p-simpleksa\ \sigma_j \epsilon S_{p,} \\ 0, inače \end{cases}$$

4.7. Operatori ruba kao diferencijali

U ovom poglavlju objašnjeno je najosnovnije svojstvo operatora ruba, a to svojstvo jest da je granica operatora ruba jednaka nuli. Operatori ruba u različitim dimenzijama uklapaju se u sljedeću sliku:

$$\dots \xrightarrow{\partial_{p+2}} C_{p+1} \xrightarrow{\partial_{p+1}} C_p \xrightarrow{\partial_p} C_{p-1} \xrightarrow{\partial_{p-1}} \dots \xrightarrow{\partial_2} C_1 \xrightarrow{\partial_1} C_{0.}$$

Osnovno svojstvo operatora ruba je da je sastav uzastopnih operatora ruba nulti operator, tj.

$$\partial_p \circ \partial_{p+1} = 0$$

za sve *p*-eve. Drugim riječima, granica operatora ruba bilo kojeg lanca je nula.

Ovo osnovno svojstvo operatora ruba je definirajuće svojstvo diferencijala simplicijalnog kompleksa. Dakle, prema definiciji diferencijala, ovo svojstvo znači da su oni diferencijali simplicijalnog kompleksa. Dokaz ovog svojstva proizlazi iz činjenice da je svaka *p*-1 simpleksna strana *p*+1 simpleksa ujedno strana točno dvaju njegovih *p*-simpleksnih strana, a budući da je 2 = 0 mod 2, rezultirajući lanac je nulti lanac.

4.8. Ciklusi i granice

Ciklusi i granice u simplicijalnom kompleksu posebne su vrste lanaca. Oni su definirani u odnosu na operatore ruba. *P*-ciklus *z* je *p*-lanac s nultom granicom, tj.

$$\partial_p z = 0.$$

Skup svih *p*-ciklusa je označen kao Z_p. To je podskup skupa C_p svih *p*-lanaca.

Primjer 1-ciklusa je 1-lanac c_2 na slici 4 jer je, računajući njegovu 1-granicu u poglavlju 4.5 dobivena formula $\partial_1 c_2$ = 0, što je prazni 0-lanac. Drugi lanac c_1 koji je prikazan na slici 4 nije 1-ciklus iz razloga jer granica, odnosno rub $\partial_1 c_2 = A + F$ nije prazni 0-lanac. Slika također objašnjava odakle dolazi sam naziv ciklusa. 1-ciklus c_2 je prikazan zatvorenom petljom, dok 1-lanac c_1, koji nije 1-ciklus, nije prikazan kao zatvorenja petlja. Broj $|Z_p|$ *p*-lanaca je potencija broja dva

$$|Z_p| = 2^{z_p},$$

gdje se eksponent z_p naziva rangom Z_p. U posebnom slučaju kada je *p* = 0, ne postoji operator ruba, ali se uvodi konvencija da su svi 0-lanci 0-ciklusi, tj. $Z_0 = C_0$, a rang z_0 je $z_0 = n_0$.

P-granica ili *p*-rub *b* je *p*-lanac koji je rub nekog *p*+1 lanca, odnosno

$$b = \partial_{p+1} c$$

za neke $c \epsilon C_{p+1}$. Skup svih *p*-rubova ili granica označen je kao B_p, koji označava podskup skupa C_p svih *p*-lanaca.

Broj $|B_p|$ *p*-rubova je također potencija broja dva

$$|B_p| = 2^{b_p},$$

gdje eksponent b_p predstavlja rang B_p.

Ako *t* označuje najveću dimenziju simpleksa u simplicijalnom kompleksu, tj. ne postoje *p*-simpleksi dimenzije *p* > *t*, tada je postoje *t*-rubovi jer ne postoji *t*+1 simpleks u simplicijalnom kompleksu.

Osnovno svojstvo operatora ruba implicira da je svaki *p*-rub ujedno *p*-ciklus jer je granica, odnosno rub od ruba uvijek jednak nuli. Preciznije, ako je $b = \partial_{p+1}c$ bilo koji *p*-rub, onda

$$\partial_p b = \partial_p(\partial_{p+1}c) = (\partial_p \circ \partial_{p+1})c = 0,$$

što znači da je *b* *p*-ciklus. Stoga vrijedi

$$B_p \subseteq Z_p$$

za sve *p*.

4.9. Homologija

P-ta homologijska grupa H_p simplicijalnog kompleksa broji *p*-cikluse koji nisu dobiveni kao *p*-rubovi, odnosno broji „prave“ cikluse među *p*-lancima. Za potrebe ovog rada, dovoljno je znati da je broj $|H_p|$ elemenata u *p*-toj homologijskoj grupi također potencija broja dva te da je njezin rang razlika između rangova z_p i b_p.

Ako se piše

$$|H_p| = 2^{\beta_p},$$

onda se rang β_p od H_p naziva *p*-tim Bettijevim brojem. Rang je izračunat iz formule

$$\beta_0 = z_0 - b_0 = n_0 - b_0,$$

$$\beta_p = z_p - b_p, \qquad 1 < p < t\text{-}1$$

$$\beta_t = z_t - b_t = z_t,$$

$$\beta_q = 0, \qquad q < t,$$

gdje je *t* najveća dimenzija simpleksa u simplicijalnom kompleksu.

5. Zaključak

Softver je glavna tehnologija koja pokreće digitalizaciju u mnogim područjima kao podrška u procesima donošenja odluka. On apstrahira fizički svijet i upravlja takvim apstraktnim resursima te implementira algoritme strojnog učenja i umjetne inteligencije za analizu velikih podataka. Način na koji se projektiraju softverska rješenja ima značajan utjecaj na postizanje ciljeva održivosti u području primjene gdje je softver implementiran, ali i unutar industrije softverskog inženjerstva. Inženjerstvo održivog softvera postaje ključan za buduće tehnološke napretke. Glavna prepreka u inženjerstvu održivog softvera je ljudska sposobnost projektiranja složenih sustava.

U ovom radu definirani su složeni sustavi i specifični izazovi modeliranja modernih softverskih sustava te njihove posljedice na ponašanje održivog softvera. Objašnjen je problem modeliranja složenog softverskog ponašanja s perspektive modeliranja lokalnih svojstava sustava koja se mjere na dijelovima softvera i globalnih svojstava sustava koja se mjere u operaciji sustava. Nadalje, predstavljena je struktura softvera kao jedan od ključnih instrumenata pomoću kojeg se modeliraju odnosi između lokalnih i globalnih svojstava sustava i objašnjena je njezina grafička reprezentacija.

Na kraju, uvodi se u topološku analizu podataka (TDA) kao alat za analizu koja može biti vrlo korisna u modeliranju složenih sustava kao komplementarni alat raznim postojećim grafičkim algoritmima. TDA je sposobna opisivati topološki prostor strukture koji može uvesti dodatnu korisnu dimenziju za objašnjavanje algoritamskih odluka u različitim primjenama. Između ostalog, uvodi se i u ključne pojmove topološke analize podataka koji su ključni za razumijevanje iste.

6. Literatura

[1] Aliance Next Generation Mobile Networks (NGMN): Green future networks: Network energy efficiency. (2021), https://www.ngmn.org/publications/green-futurenetworks-network-energy-efficiency.html

[2] Allili, M., Kaczynski, T., Landi, C., Masoni, F.: Acyclic partial matchings for multidimensional persistence: Algorithm and combinatorial interpretation. J. Math.Imaging Vis. 61(2), 174–192 (2019)

[3] Bendich, P., Edelsbrunner, H., Kerber, M.: Computing robustness and persistence for images. IEEE Trans. Vis. Comput. Graph. 16(6), 1251–1260 (2010)

[4] Callahan, D., Carle, A., Hall, M.W., Kennedy, K.: Constructing the procedure call multigraph. IEEE Transactions on Software Engineering 16(4), 483–487 (1990).

[5] Cang, Z., Mu, L., Wu, K., Opron, K., Xia, K., Wei, G.W.: A topological approach for protein classification. Computational and Mathematical Biophysics 3(1) (2015)

[6] Carlsson, G.: Topology and data. Bull. Amer. Math. Soc. (N.S.) 46(2), 255–308 (2009)

[7] Carlsson, G., Vejdemo-Johansson, M.: Topological data analysis with applications. Cambridge University Press, Cambridge (2022)

[8] Choudhary, A., Kerber, M., Raghvendra, S.: Improved approximate rips filtrations with shifted integer lattices and cubical complexes. J. Appl. Comput. Topol. 5(3), 425–458 (2021)

[9] Dey, T.K., Mandal, S., Mukherjee, S.: Gene expression data classification using topology and machine learning models. BMC Bioinform. 22-S (Suppl. 10), Article no. 627 (2021)

[10] Edelsbrunner, H., Harer, J.: Computational Topology – an Introduction. American Mathematical Society (2010)

[11] Edelsbrunner, H., Morozov, D.: Persistent homology: theory and practice. In: Proceedings of the European Congress of Mathematics 2012, pp. 31–50. EMS Press, Berlin (2014)

[12] Fremerey, C., Mueller, M., Clausen, M.: Towards bridging the gap between sheet music and audio. In: Knowledge representation for intelligent music

processing. Dagstuhl Seminar Proceedings (DagSemProc), vol. 9051, pp. 1–11. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl, Germany (2009)

[13] Galinac Grbac, T.: The role of functional programming in management and orchestration of virtualized network resources. Part I. System structure for complex systems and design principles. In: Composability, Comprehensibility and Correctness of Working Software, 7th Winter School. Revised Selected Papers, Lecture Notes in Computer Science vol. 11916, Springer, Cham (2023), to appear. Available at https://arxiv.org/abs/2107.12136

[14] Galinac Grbac, T., Domazet, N.: The role of functional programming in management and orchestration of virtualized network resources. Part II. Network evolution and design principles. In: Composability, Comprehensibility and Correctness of Working Software, 8th Summer School. Revised Selected Papers, Lecture Notes in Computer Science vol. 11950, Springer, Cham (2023), to appear. Available at https://arxiv.org/abs/2107.12227

[15] Gasarch, W., Fasy, B.T., Wang, B.: Open problems in computational topology. SIGACT News 48(3), 32–36 (2017)

[16] Giray, G., Bennin, K.E., Köksal, Ö, Babur, Ö, Tekinerdogan, B.: On the use of deep learning in software defect prediction. Journal of Systems and Software 195, Article no. 111537 (2023)

[17] Hatcher, A.: Algebraic topology. Cambridge University Press, Cambridge (2002)

[18] Hilty, L.M., Aebischer, B.: ICT for sustainability: An emerging research field. In: Hilty, L.M., Aebischer, B. (eds.) ICT Innovations for Sustainability. pp. 3–36. Springer International Publishing, Cham (2015)

[19] Hilty, L.M., Arnfalk, P., Erdmann, L., Goodman, J., Lehmann, M., Wäger, P.A.: The relevance of information and communication technologies for environmental sustainability – a prospective simulation study. Environmental Modelling & Software 21(11), 1618–1629 (2006)

[20] International Organization for Standardization (ISO), International Electrotechnical Commission (IEC): ISO/IEC 25002:2024 Systems and software engineering – Systems and software Quality Requirements and

Evaluation (SQuaRE) – Quality model overview and usage. International standard, 1st edition (2024).

[21] Kang, L., Xu, B., Morozov, D.: Evaluating state space discovery by persistent cohomology in the spatial representation system. Frontiers Comput. Neurosci. 15, Article no. 616748 (2021)

[22] Kim, H.S., Yi, C., Kim, Y., Park, U., Kook, W., Oh, B., Kim, H., Park, T.: Topological data analysis can extract sub-groups with high incidence rates of type 2 diabetes. Int. J. Data Min. Bioinform. 22(1), 44–60 (2019)

[23] Lago, P., Koçak, S.A., Crnković, I., Penzenstadler, B.: Framing sustainability as a property of software quality. Commun. ACM 58(10), 70–78 (2015)

[24] Lago, P., Meyer, N., Morisio, M., Müller, H.A., Scanniello, G.: Leveraging "energy efficiency to software users": Summary of the second greens workshop, at ICSE 2013. SIGSOFT Softw. Eng. Notes 39(1), 36–38 (2014)

[25] Leykam, D., Angelakis, D.G.: Topological data analysis and machine learning. Advances in Physics: X 8(1), Art. no. 2202331, 24 pages (2023)

[26] Malott, N.O., Chen, S., Wilsey, P.A.: A survey on the high-performance computation of persistent homology. IEEE Trans. Knowl. Data Eng. 35(5), 4466–4484 (2023)

[27] Mauša, G., Galinac Grbac, T.: Co-evolutionary multi-population genetic programming for classification in software defect prediction: An empirical case study. Appl.Soft Comput. 55, 331–351 (2017)

[28] Munch, E.: A user's guide to topological data analysis. Journal of Learning Analytics 4(2), 47–61 (2017)

[29] Munkres, J.R.: Elements of algebraic topology. Addison-Wesley Publishing Company, Menlo Park, CA (1984)

[30] Munkres, J.R.: Topology. Prentice Hall, Inc., Upper Saddle River, NJ (2000)

[31] Pandey, S.K., Mishra, R.B., Tripathi, A.K.: Machine learning based methods for software fault prediction: A survey. Expert Systems with Applications 172, Article no. 114595 (2021)

[32] Penzenstadler, B., Raturi, A., Richardson, D., Tomlinson, B.: Safety, security, now sustainability: The nonfunctional requirement for the 21st century. IEEE Software 31(3), 40–47 (2014)

[33] Petrić, J., Galinac Grbac, T.: Software structure evolution and relation to system defectiveness. In: Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering, EASE ’14. pp. 34:1–34:10. ACM (2014)

[34] Petrić, J., Galinac Grbac, T., Dubravac, M.: Processing and data collection of program structures in open source repositories. In: Proceedings of the 3rdWorkshop on Software Quality Analysis, Monitoring, Improvement and Applications (SQAMIA 2014). CEUR Workshop Proceedings, vol. 1266, pp. 57–66. CEUR (2014)

[35] Pita Costa, J., Škraba, P.: A topological data analysis approach to the epidemiology of influenza. In: SIKDD15 Conference Proceedings (2015)

[36] Pita Costa, J., Galinac Grbac, T.: The topological data analysis of time series failure data in software evolution. In: Proceedings of the 8th ACM/SPEC on International Conference on Performance Engineering Companion. pp. 25–30. ICPE ’17 Companion, ACM, New York, NY, USA (2017).

[37] Puh E., Galinac Grbac, T., Grbac, N.: Preliminary study of higher dimensional software structures. In: Proceedings of the 10th Workshop on Software Quality Analysis, Monitoring, Improvement and Applications (SQAMIA 2023). CEUR Workshop Proceedings, vol. 3588, pp. 13–25. CEUR (2023)

[38] Valverde, S., Solé, R.V.: Network motifs in computational graphs: a case study in software architecture. Physical review. E, Statistical, nonlinear, and soft matter physics 72 2 Pt 2, 026107–1, 026107–8 (2005)

[39] Vranković, A., Galinac Grbac, T., Car, Ž.: Software structure evolution and relation to subgraph defectiveness. IET Softw. 13(5), 355-367 (2019)

[40] Galinac Grbac, T., Puh, E., Grbac, N.: Algorithms for Sustainable System Topologies, lecture notes, SusTrainable Summer School 2022, University of Rijeka, July 2022

7. Prilozi

7.1. Popis slika

7.2. Popis tablica

8. Sažetak

Glavni cilj ovog rada je pružiti blagi uvod u topološku analizu podataka zbog njenih mogućih primjena u održivom softverskom inženjerstvu. Kada se topološka analiza podataka primjenjuje na strukture softverskih sustava, mogu se uočiti dodatne strukturne metrike iz topološkog prostora koje mogu biti korisne kao dopunska tehnika analize za razumijevanje složenog ponašanja softverskih sustava, dok se teži postizanju potrebnih atributa kvalitete i održivih ciljeva. Razmatraju se specifični izazovi u vezi s održivošću kao atributom kvalitete softvera, ali i kao aspektom softverskog inženjerstva. U radu su objašnjeni topološki pojmovi i ideje te su oni objašnjeni na zasebnim primjerima u svakom od poglavlja. Topološki pojmovi i ideje predstavljeni su i objašnjeni na zasebnim kratkim primjerima, u kojima su eksplicitno izračunate osnovne topološke invarijante. Rezultati ovog rada objavljeni su u obliku znanstvenog rada na konferenciji.

Ključne riječi: Softverski sistemi, softverska struktura, topološka analiza podataka, računska topologija, algebarska topologija

9. Abstract

The main goal of this paper is to provide a gentle introduction to topological data analysis because of its possible applications to sustainable software engineering. When applying topological data analysis to software system structures, one can observe additional structural metrics from the topological space which may be useful as a complement analysis technique in understanding complex software system behaviour while aiming to address required quality attributes and sustainable goals. Specific challenges on addressing sustainability as a software quality attribute, but also as an aspect of engineering software, are discussed. The topological notions and ideas are introduced and explained on various examples, in which the basic topological invariants are calculated explicitly. The results of this final paper were published in the form of a scientific paper at the conference.

Keywords: Software systems, software structure, topological data analysis, computational topology, algebraic topology