

UNIVERZA V LJUBLJANI
FAKULTETA ZA MATEMATIKO IN FIZIKO

Matematika – 1. stopnja

Gašper Golob

Preverjanje ravninskosti grafov v Lean

Delo diplomskega seminarja

mentor: prof. dr. Andrej Bauer
somentorica: assist. dr. Katja Berčič

Ljubljana, 2021/22

1 Uvod

Pri dokazovanju izrekov se pogosto pojavi veliko obravnavanje primerov, ali računanja. V teh primerih, bi bilo pogosto preveč zamudno, to delati ročno, in posledično, se za specifične korake dokaza uporabi računalnik. Problemi, pa se lahko pojavijo, če koda vsebuje napake, saj nam program lahko v teh primerih vrne rezultate, ki so videti pravilni, vendar zaradi njih dokažemo neresnične trditve. Da rešimo ta problem, je smiselno uporabljati programe, ki nam poleg izvajanja algoritmov omogočajo tudi dokazovanje lastnosti o njih.

Če lahko govorimo o lastnostih algoritmov, pa se pravzaprav izkaže, da lahko v računalniku formuliramo definicije in izreke, kar nam omogoča, da s pomočjo računalnika, formaliziramo matematiko. Ena glavnih prednosti tega pristopa je, da računalnik ne more delov dokazov zaključiti na podlagi tega, da so očitni, kar pomeni, da mora dokaz res slediti vsem aksiomom, ki se jih zastavi.

Programi, ki so narejeni posebej za namen dokazovanja izrekov so pomočniki za dokazovanje. Te pogosto opravljajo poleg formalizacije izrekov še določene dodatne naloge, kot recimo samodejno dokazovanje. Uporablja se jih lahko tudi kot bazo podatkov, za razna dejstva. Primer uporabe, bi bil recimo, da za poljuben izrek iz baze dobimo možne leme, ki jih lahko uporabimo, za nadaljevanje dokaza. Še ena uporaba je shranjevanje primerov, kar bi lahko matematikom pomagalo z na primer iskanjem protiprimerov za dano izjavo.

Preden je možno iskati protiprimer, pa je potrebno v dokazovalnik vnesti bazo primerov, kar je pravzaprav motivacija projekta, del katerega je ta diplomska naloga. Bolj specifično je namen projekta, da se v dokazovalnik, vnese primere grafov iz zbirke grafov House of Graphs in samodejno izpelje nekatere od njihovih lastnosti. Primeri lastnosti, ki so bodo vnesle so na primer povezanost grafa, regularnost grafa, kromatično število in če je graf ravninski, kar pa je tudi cilj te diplomske naloge. Bolj specifično je cilj te diplomske naloge razviti nek mehanizem, ki bo v dokazovalniku čim bolj samostojno preverjal, če je nek graf ravninski.

Projekt se bo delal v pomočniku za dokazovanje Lean, saj je eden izmed redkih dokazovalnikov, ki omogoča dokazovanje izrekov in hkrati deluje tudi, kot programski jezik. Celoten dokazovalnik, je zgrajen na podlagi nekega manjšega jedra, za katerega je na roko preverjeno, da je formalno pravilno in za tem se preostanek funkcionalnosti gradi na podlagi tega. Ena izmed funkcionalnosti, ki je zgrajen na jedru programa, so taktike, ki omogočajo, da se napiše postopek za dokazovanje trditev, ki potem lahko reši trditve, od katerih dokazi imajo podobno strukturo. Primer, bi bila recimo taktika, ki rešuje enačbe, s tem, da poskuša različne izreke o seštevanju in množenju. Kot tudi mnogi drugi pomočniki za dokazovanje, Lean temelji na teoriji tipov, ki igra vlogo osnove matematike, podobno, kot ponavadi teorija množic. Kljub temu pa, se pri delu z Lean-om pojavijo določene težave.

1.1 Dokaz s pričo

Čeprav je teoretično možno vse narediti v Leanu, se pojavijo problemi pri hitrosti. Pogosto se pri strukturah, da se dobi lepše dokaze, žrtvuje učinkovitost, kar je, za dokazovanje lastnosti dobro, za samo računanje pa hitro preseže časovne in prostorske omejitve.

Primer tega, bi bila že naravna števila, ki so predstavljena po pristopu Peanovih aksiomov, kar pomeni, da imajo nek ničeln element in generator, ki pravi, da je naslednik vsakega naravnega števila tudi naravno število. V Lean 3 manjkajo tudi učinkovite podatkovne strukture z naključnim dostopom, zaradi česar je pogosto težko implementirati učinkovite funkcije.

Posledično je, v primeru, ko imamo veliko število grafov in želimo ugotoviti, če so ravninski, dobro čim več dela narediti z zunanjimi programi, ki so hitrejši. Hkrati seveda želimo, da je vse, kar dobimo, še vedno formalno, torej, da se ne rabimo zanašati na pravilnost zunanjih programov. To dosežemo s pomočjo dokazov s pričami.

Enostaven primer prič bi se pojavil v dokazu, da neko število ni praštevilo. Tu lahko za pričo vzamemo poljubnega delitelja tega števila in formalizacija potem zahteva samo, da preverimo, če je to število res delitelj začetnega števila. V tem primeru, se s pričo rešimo preverjanja velikega števila potencialnih deliteljev znotraj dokazovalnika.

Ponavadi je cilj, da je priča dovolj majhna in, da iz nje čim preprosteje sledi dokaz. Majhnost je pomembna, saj pogosto nimamo neomejeno spomina in zato, ne bi bilo dobro, če bi velikost priče naraščala eksponentno. Podobno je pomembna tudi enostavnost dokaza iz priče, saj tudi časa nimamo neomejeno.

S tem v mislih lahko sedaj izberemo primerno predstavitev za grafe in primerno karakterizacijo za ravninske grafe.

2 Predstavitev grafa s puščicami

Za razvoj teorije ravninskih grafov, bomo najprej definirali predstavitev grafov, ki bo imela čim enostavnejšo karakterizacijo ravninskih grafov.

Razlog za to je, da je definicijo, z vložitvijo grafa v ravnino, problematično formalizirati v dokazovalniku, ker se pogosto zanašamo na to, da za ravninske grafe opazimo, da njihova slika nima povezav, ki bi se sekale. To je naloga, ki je relativno enostavna za ljudi, za računalnik, pa bilo to težko formulirati.

Grafe zato, raje predstavljamo kot kombinatorične preslikave.

Definicija 2.1. Kombinatorična preslikava je trojica $m = (D, \alpha, \sigma)$, kjer končni množici D pravimo množica puščic in za katero velja:

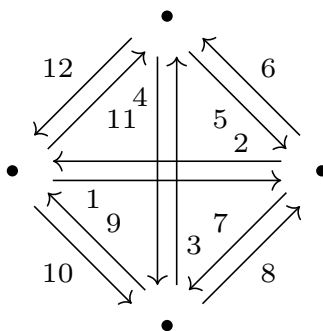
- $\alpha, \sigma : D \rightarrow D$ sta bijektivni preslikavi,
- $\alpha^2 = id$,
- za vse $e \in D$ je $\alpha(e) \neq e$.

Po geometrijski interpretaciji, so puščice daljice, ki imajo neko izhodišče in se končajo v izhodišču neke druge puščice. Pri tem je α preslikava, ki puščici priredi obratno obrnjeno puščico, σ pa je preslikava, ki ciklično, v nasprotni smeri urinega kazalca, izmenjuje med puščicami, ki imajo isto izhodišče.

Dodatno opazimo še, da lahko preslikavi α in σ razbijemo na disjunktne cikle. Cikle α lahko potem interpretiramo kot povezave grafa, ki ga predstavlja kombinatorična preslikava, cikle σ pa kot vozlišča tega grafa. Poleg tega imajo posebno interpretacijo tudi cikli preslikave $\alpha \circ \sigma$, ki predstavljajo lica.

Opomba 2.2. V kontekstu kombinatoričnih preslikav bomo cikle α , kar imenovali za povezave, cikle σ za vozlišča in cikle $\alpha \circ \sigma$ za lica.

Opomba 2.3. Če je v vozlišče, bomo z $A(v)$ označili množico vseh puščic, ki so v ciklu v .



Slika 1: Kombinatorična predstavitev K_4

Ta intuicija bo morda bolj razvidna iz naslednjega primera.

Primer 2.4. Naj bo $m = (D, \alpha, \sigma)$ kombinatorična preslikava, za katero velja $E = \{1, 2, \dots, 12\}$,

$$\alpha = (1, 2)(3, 4)(5, 6)(7, 8)(9, 10)(11, 12),$$

$$\sigma = (12, 4, 5)(6, 2, 7)(10, 1, 11)(8, 3, 9),$$

$$\alpha \circ \sigma = (1, 7, 3, 5, 2, 11, 4, 9)(12, 10, 8, 6).$$

Ta preslikava ima glede na prej omenjeno interpretacijo 6 povezav, 4 vozlišča in 2 lici. Glede na to interpretacijo, pa lahko dobimo tudi sliko 1, iz katere je razvidno, da ta kombinatorična preslikava predstavlja graf K_4 . Vidimo, da imamo res šest povezav in štiri vozlišča, lica pa je, zaradi prekrivanja povezav, težko interpretirati. $\diamond$

V tem primeru, smo videli nek graf, ki je predstavljen s kombinatorično preslikavo. Podobno se da predstaviti tudi večino ostalih grafov, pri čemer so izjeme grafi z izoliranimi vozlišči. V tem primeru seveda nimamo nobene povezave, ki bi imela to vozlišče za izhodišče in ker so osnovni objekti kombinatorične preslikave puščice, s katerimi lahko predstavljamo samo povezave, izolirane vozlišča ne moremo predstaviti.

Še ena opazka je, da na ta način nismo dobili ravninskega grafa, čeprav vemo, da graf K_4 je ravninski.

Pojavi se vprašanje, če obstaja pogoj, ki bi nam dal samo tiste kombinatorične preslikave, ki jih lahko v nekem smislu vložimo v ravnino.

3 Genus kombinatorične preslikave

Preden definiramo genus kombinatorične preslikave, začnimo z definicijo vložitve kombinatorične preslikave v ravnino.

Definicija 3.1. Jordanov lok v ravnini je zaloga vrednosti zvezne injektivne preslikave $\varphi : [0, 1] \rightarrow \mathbb{R}^2$. Točki $\varphi(0)$ in $\varphi(1)$ sta krajišči, slika $\varphi((0, 1))$ pa je notranjost loka.

Definicija 3.2. Naj bo $m = (D, \alpha, \sigma)$ kombinatorična preslikava. Preslikava I iz množice puščic v Jordanove loke je vložitev m v ravnino, če za vse $e_1, e_2 \in D$, kjer je $e_1 \neq e_2$, velja naslednje:

- če je $\alpha(e_1) \neq e_2$ sta notranjosti Jordanovih lokov $I(e_1)$, $I(e_2)$ disjunktni,
- če je $\alpha(e_1) = e_2$ sta Jordanova loka $I(e_1)$, $I(e_2)$ enaka,

Veljati mora tudi, da je za vsa vozlišča v krajišče loka $I(e)(0)$ enako za vse $e \in A(v)$.

Zadnji pogoj je, da sta orientaciji I in m usklajeni.

Opomba 3.3. Orientacija I je usklajena z orientacijo m , če za vsako vozlišče v vrstni red vložitev puščic $e \in D$, v ravnino kot $I(e)$, v pozitivni smeri okoli izhodišča, porodi enako urejenost kot cikel v .

Sedaj lahko definiramo lastnost, ki bo pomagala s karakterizacijo ravninskih grafov.

Definicija 3.4. Naj bo $m = (D, \alpha, \sigma)$ kombinatorična preslikava. Definirajmo še grupo H , ki je generirana s preslikavama α, σ in naj bo n število orbit delovanja grupe H na puščice D . Potem je genus te kombinatorične preslikave:

$$g(m) = n + \frac{1}{2} (Z(\alpha) - Z(\sigma) - Z(\alpha \circ \sigma)), \quad (1)$$

kjer z Z označimo funkcijo, ki preslikavi priredi število ciklov v zapisu z disjunktnimi cikli.

Opomba 3.5. Delovanje $h \in H$ na $e \in D$ je $h(e)$.

Sedaj lahko definiramo, kdaj je kombinatorična preslikava ravninska.

Definicija 3.6. Kombinatorična preslikava m je ravninska, če je $g(m) = 0$.

Izkaže se, da je to res natanko tedaj, ko imamo vložitev v ravnino v bolj klasičnem smislu, kar tudi formalizira naslednja trditev.

Trditev 3.7. *Kombinatorična preslikava ima vložitev v ravnino, natanko tedaj, ko je ravninska.*

Opomba 3.8. Z genusom je možno preverjati tudi obstoj vložitve v poljubne druge ploskve, vendar moramo v tem primeru teorijo izpeljati s pomočjo klasifikacije ploskev, kar presega obseg te diplomske naloge.

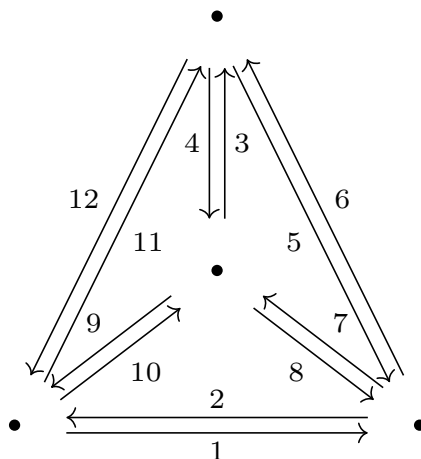
Delovanje te karakterizacije, pa se vidi v naslednjih dveh primerih.

Primer 3.9. Spomnimo se na primer 2.4. V njem smo imeli kombinatorično preslikavo, za katero smo opazili, da predstavlja graf K_4 , pri kateri slika ni bila ravninski graf. Opazimo, da je veljalo: $Z(\alpha) = 6$, $Z(\sigma) = 4$ in $Z(\alpha \circ \sigma) = 2$. Velja tudi, da ima delovanje grupe generirane z α, σ samo eno orbito, kar je najlažje razvidno iz slike 1. Da je orbita samo ena, se vidi tako, da lahko od poljubne puščice pridemo do poljubne druge z rotiranjem okoli vozlišč in obračanjem puščic. Genus te preslikave je bil torej

$$g(m) = 1 + \frac{1}{2} (6 - 4 - 2) = 1$$

in zato kombinatorična preslikava tudi po naši definiciji ni ravninska. ◇

Opomba 3.10. Da je orbita samo ena, je še lažje opaziti iz tega, da je graf, ki ga predstavlja kombinatorična preslikava, povezan.



Slika 2: Ravninska kombinatorična predstavitev K_4

Kot bomo videli v naslednjem primeru, pa lahko kombinatorično preslikavo za graf K_4 definiramo tudi nekoliko drugače.

Primer 3.11. Sedaj vzamemo za kombinatorično preslikavo $m = (D, \alpha, \sigma)$, kjer so puščice $D = \{1, 2, \dots, 12\}$ in

$$\begin{aligned}\alpha &= (1, 2)(3, 4)(5, 6)(7, 8)(9, 10)(11, 12), \\ \sigma &= (12, 4, 5)(6, 7, 2)(11, 1, 10)(8, 3, 9), \\ \alpha \circ \sigma &= (1, 6, 12)(2, 10, 8)(3, 5, 7)(4, 9, 11).\end{aligned}$$

V primerjavi s prejšnjo kombinatorično preslikavo je α enaka, σ pa ima samo spremenjen vrstni red puščic znotraj ciklov. Ti dve dejstvi pravzaprav pomenita, da kombinatorična preslikava še vedno predstavlja enak graf. Razlika pa se opazi, že v sliki, kot vidimo, če pogledamo sliko 2. Na njej opazimo, da nam ta kombinatorična preslikava na nek način poda ravninski graf.

Opazimo tudi, da je kombinatorična preslikava ravninska tudi po naši definiciji, saj je

$$g(m) = 1 + \frac{1}{2} (6 - 4 - 4) = 0,$$

kjer smo spet samo prešteli cikle in opazili, da imamo samo eno orbito delovanja. $\diamond$

4 Formalizacija za ravninske grafe

Zdaj, ko smo našli primerno karakterizacijo ravninskih grafov, lahko okvirno opišemo potek formalizacije.

Najprej iz zunanjega programa dobimo pričo, ki je sestavljena iz množice puščic in dveh preslikav. Za preslikavi najprej preverimo, da sta bijektivni, nato pa za α preverimo, da je involucija in da nobene puščice ne preslika same vase. Pri tem je lahko preveriti, če je α involucija, saj je dovolj, če preverimo, da je $\alpha^2 = id$ za vse puščice, ki pa jih je končno mnogo. Iz dejstva, da je

involucija sledi tudi, da je α bijekcija. Podobno lahko preverimo tudi lastnost, da nobene puščice, ne preslika same vase.

Podobnega trika za σ ne moremo uporabiti in ker je iskanje inverzne preslikave ali dokazovanje injektivnosti in surjektivnosti počasno, se tudi tukaj za dokaz uporabi dokazovanje s pomočjo prič. Za pričo tu vzamemo inverzno preslikavo, kjer dokaz s pričo naredimo tako, da spet izračunamo, če je kompozitum σ in predlaganega inverza enak identiteti za vse puščice.

S tem smo preverili, da imamo res kombinatorično preslikavo $m = (D, \alpha, \sigma)$ in posledično nam manjka samo še izračun genusa. Za izračun tega moramo za poljubne bijektivne preslikave prešteti število ciklov v faktorizaciji na disjunktne cikle in število orbit delovanja grupe, generirane z α, σ , na puščicah D .

Za faktorizacijo na cikle se ponovno lahko uporabi priče, kjer je priča seznam vseh ciklov faktorizacije. V tem primeru, je treba preveriti samo, da ti cikli res podajo začetno preslikavo in da so disjunktne.

Sedaj nam manjka samo še število orbit delovanja grupe. Trenutno natančen pristop še ni določen, vendar je primer naivnega pristopa, da se zanašamo na končnost množice puščic in elementov iz grupe, kar omogoča, da preprosto generiramo vse orbite.

Kot našo končno strukturo potem za ravninske grafe shranimo kombinatorično preslikavo in dokaz, da je njen genus enak 0.

Ceprav ta karakterizacija dela dobro za ravninske grafe, pa bi bilo pri grafih, ki niso ravninski, treba pokazati, da primerna kombinatorična preslikava ne obstaja. Dokaz neobstoja ni učinkovit, zato za grafe, ki niso ravninski, želimo najti drugačno karakterizacijo.

5 Subdivizija grafa Kuratowskega

Najbolj znan primer take karakterizacije je preverjanje obstoja oziroma neobstoja podgrafa, ki je izomorfen subdiviziji enega izmed grafov K_5 ali $K_{3,3}$.

Trditev 5.1. *Graf je ravninski, če in samo če, ne vsebuje subdivizije izomorfne K_5 ali $K_{3,3}$.*

Opazimo, da nam ta karakterizacija ponudi dobro pričo za dokaz, da graf ni ravninski, saj moramo za pričo podati le podgraf izomorfen tovrstni subdiviziji.

Po drugi strani ta karakterizacija ni dobra za dokaz, da graf je ravninski, saj moramo potem pokazati neobstoje primerne podgrafa.

Literatura

- [1] Gertrud Josefine Bauer. Formalizing plane graph theory: towards a formalized proof of the kepler conjecture. 2006.
- [2] Gunnar Brinkmann, Kris Coolsaet, Jan Goedgebeur, and Hadrien Mélot. House of graphs: A database of interesting graphs. *Discrete Applied Mathematics*, 161(1–2):311–314, Jan 2013.
- [3] Kurt Mehlhorn and Stefan Näher. *LEDA: A Platform for Combinatorial and Geometric Computing*. Cambridge University Press, January 1999.