



**MATEMATICKO-FYZIKÁLNÍ
FAKULTA**
Univerzita Karlova

BAKALÁŘSKÁ PRÁCE

Zuzana Šimečková

Varianty Eberhardovy věty

Informatický ústav Univerzity Karlovy

Vedoucí bakalářské práce: doc. Mgr. Robert Šámal, Ph.D.

Studijní program: Informatika

Studijní obor: IOI

Praha 2018

Prohlašuji, že jsem tuto bakalářskou práci vypracoval(a) samostatně a výhradně s použitím citovaných pramenů, literatury a dalších odborných zdrojů.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona v platném znění, zejména skutečnost, že Univerzita Karlova má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle §60 odst. 1 autorského zákona.

V dne

Podpis autora

Poděkování.

Název práce: Varianty Eberhardovy věty

Autor: Zuzana Šimečková

Ústav: Informatický ústav Univerzity Karlovy

Vedoucí bakalářské práce: doc. Mgr. Robert Šámal, Ph.D., Informatický ústav Univerzity Karlovy

Abstrakt: Abstrakt.

Klíčová slova: klíčová slova

Title: Eberhard-Like Theorems

Author: Zuzana Šimečková

Institute: Computer Science Institute of Charles University

Supervisor: doc. Mgr. Robert Šámal, Ph.D., Computer Science Institute of Charles University

Abstract: Abstract.

Keywords: key words

Obsah

Úvod	2
1 Pojmy a definice	3
2 Strategie důkazu	5
2.1 Triarky a operace s nimi	5
2.2 Důkaz za předpokladu existence pomocných grafů	6
3 Řešítko	8
3.1 Algoritmus	8
3.2 Uživatelská dokumentace/formáty výstupu	10
4 Výsledky	12
4.1 Kreslení	12
Závěr	15
Seznam použité literatury	16
Seznam obrázků	17
Seznam použitých zkratk	18
A Přílohy	19
A.1 První příloha	19

Úvod

Zkusíme-li upravovat Eulerův vzorec pro rovinné grafy $|V| - |E| + |S| = 2$, můžeme pro kubické grafy dospět do tvaru

$$\sum_{k \geq 3} (6 - k)p_k = 12, \quad (1)$$

kde p_k značí počet k -hranných stěn grafu. Výraz (1) je nutnou podmínkou pro existenci rovinného kubického grafu s daným počtem stěn. Pozoruhodně, počet šestihranných stěn v této podmínce nehraje žádnou roli. Toho si všiml Eberhard [1], který v roce 1891 formuloval a dokázal, že pokud můžeme volit p_6 , dokážeme najít kubický graf zadaný počtem ostatních stěn, který je rovinný.

Věta 1 (Eberhardova věta). *Pro každou posloupnost $(p_k \mid 3 \leq k \neq 6)$ kladných celých čísel, splňující (1) existuje taková hodnota p_6 , že existuje rovinný 3-regulární 3-souvislý graf, který má právě p_k k -hranných stěn pro každé $k \in \mathbb{N}, k \geq 3$.*

Na větu navázali další a dnes je známá celá řada jejích obměn. Fisher [2] dokázal silnější verzi Eberhardovy věty, kde počet potřebných stěn velikosti 6 shora omezil počtem ostatních stěn chtěného grafu. Grünbaum [3] představuje vlastní stručnější důkaz Eberhardovy věty a shrnuje výsledky podobného typu.

TODO protáhnout jestli bude třeba

DeVos a kol. [4] představuje obdobu Eberhardovy věty, kde místo p_6 můžeme volit p_5 a p_7 . Článek také ukazuje, že konstrukce důkazu, kterou autoři použili, by mohla jít uplatnit i pro další velikosti stěn, které mohou splnit (1).

V tomto textu nejprve zavedeme potřebnou terminologii a představíme zmiňovanou strategii důkazu, která pro dokončení potřebuje najít pomocné grafy. Cílem práce bylo tyto grafy získat. V závěrečných kapitolách navrhne program, který bude potřebné grafy hledat, a nakonec představíme, pro které velikosti stěn jsme grafy získali a tedy dokončili důkaz dalších variant Eberhardovy věty.

1. Pojmy a definice

Pokud čtenář není seznámen se základy teorie grafů, doporučujeme začít třeba knihou Kapitoly z diskretní matematiky [5]. Před zadefinováním základní terminologie pro tuto práci dokažme nejprve, jak z Eulerova vzorce získáme (1).

Důkaz. Pro každý kubický graf $G = (V, E)$ platí $3|V| = 2|E|$. Obecně platí, že počet stěn s můžeme přepsat následovně $s = \sum_{k \geq 3} p_k$ a pro rovinné grafy platí i $\sum_{k \geq 3} k \cdot p_k = 2|E|$, protože pokud pro každou stěnu započteme každou její hranu, získáme právě $2|E|$. Úpravou a dosazením do vzorce získáme požadovaný výraz:

$$\begin{aligned} |V| - |E| + |S| &= 2 \\ -\frac{1}{3}|E| + s &= 2 \\ -\frac{1}{6} \sum_{k \geq 3} k \cdot p_k + \sum_{k \geq 3} p_k &= 2 \\ \sum_{k \geq 3} (6 - k)p_k &= 12 \end{aligned}$$

□

Zavedme nyní základní pojmy. Stěně rovinného grafu, která se skládá z k hran, budeme říkat jednoduše **k -úhelník**. Pokud máme posloupnost $(p_k) = (p_3, p_4, \dots)$, která bude představovat počty k -úhelníků v grafu, definujeme $\mathbf{P} = \{k \mid p_k \neq 0\}$ jako její **množinu stěn**.

Klasifikujme posloupnosti $(p_k) = (p_3, p_4, \dots)$ podle jejich vlastností.

Definice 1 (Neutrální posloupnost). *Posloupnost $(p_k) = (p_3, p_4, \dots)$ nezáporných celých čísel je neutrální, pokud $\sum_{k \geq 3} (6 - k)p_k = 0$.*

Definice 2 (Přípustná posloupnost). *Posloupnost $(p_k) = (p_3, p_4, \dots)$ nezáporných celých čísel je přípustná, pokud $\sum_{k \geq 3} (6 - k)p_k = 6$.*

Definice 3 (Realizovatelná posloupnost). *Posloupnost $(p_k) = (p_3, p_4, \dots)$ nezáporných celých čísel je realizovatelná, pokud existuje konečný rovinný kubický graf, který má právě p_k k -úhelníků.*

Všimněme si, že původní Eberhardova věta, a i mnoho jejích variant, byla formulována pro 3-souvislé grafy, tedy přesněji pro jednoduché konvexní 3-polytopy. Bijekci mezi těmito dvěma strukturami dokázal až o několik desítek let později Steinitz [6].

Pro ujištění pojmů uvedme následující jednoduché vlastnosti neutrálních, přípustných a realizovatelných posloupností, které využijeme později.

Pozorování 2. *Pro každou neutrální (či přípustnou nebo realizovatelnou) posloupnost $p = (p_3, p_4, \dots)$ existuje $h \in \mathbb{N}$, že $\forall k \in \mathbb{N}, k > h$ platí $p_k = 0$.*

Důkaz. Neutrální posloupnost p splňuje $\sum_{k \geq 3} (6 - k)p_k = 0$ neboli $\sum_{3 \leq k \leq 5} (6 - k)p_k = -\sum_{k \geq 5} (k - 6)p_k$. Všechny sčítance na obou stranách výrazu jsou kladné. Hodnota

levého součtu tří sčítanců je konečná. Pravý součet má tedy také konečnou hodnotu a navíc mají všechny jeho nenulové sčítance hodnotu alespoň 1. Nenulových hodnot v posloupnosti je tedy pouze konečně.

□

Pozorování 3. *Pro každou q neutrální posloupnost $\exists a, b \in Q : a < 6 \wedge b > 6$.*

Součet dvou neutrálních posloupností je neutrální posloupnost. Součet neutrální a přípustné posloupnosti je přípustná posloupnost.

Pro každou přípustnou posloupnost $(p_k) = (p_3, p_4, \dots)$ existuje neutrální posloupnost q , že $p + qn$ je realizovatelná pro nějaké $n \in \mathbb{N}$.

Pro ukázání první vlastnosti si stačí uvědomit, že chceme $\sum_{k \geq 3} (6 - k)p_k = 0$, kde pro $k < 6$ je sčítaný člen kladný, zato pro $k > 6$ je sčítaný člen záporný. Druhá vlastnost plyne z definice a distributivity násobení. Třetí tvrzení pro je slabší verze Věty 1.

K poslední vlastnosti se nabízí otázka, jak lze q omezit, aby pořad $p + qn$ byla realizovatelná. Eberhard ukázal, že stačí $q = (0, 0, 0, q_6 = 1, 0, \dots)$, tedy přidat šestiúhelníky. DeVos a kol. ukázal že i omezení na $q = (0, 0, 1, 0, 1, 0, \dots)$ funguje. V této práci ukážeme, že posloupností s pouze dvěma nenulovými hodnotami, které v tomto směru vyhovují, existuje výrazně víc.

2. Strategie důkazu

Představme hypotézu, kterou se snažíme ověřit.

Hypotéza 4. *Mějme přípustnou posloupnost $p = (p_k | 3 \leq k \neq 6)$ a neutrální posloupnost $q = (q_k | 3 \leq k \neq 6)$, pak existuje takové přirozené n , že $p + nq$ je realizovatelná.*

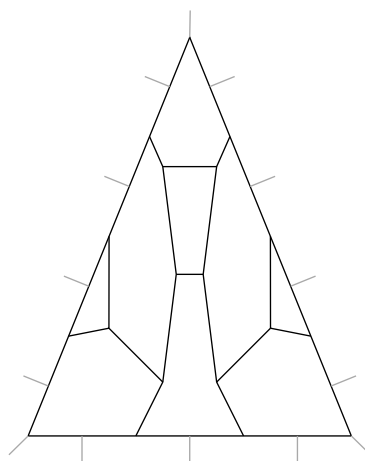
V článku [4] autoři naznačují konstrukci důkazu, za předpokladu, že existují nějaké pomocné grafy. V další kapitole představíme způsob, jak takové grafy hledat. Teď se zaměříme na důkaz samotný, respektive nejprve představíme graf, který v důkazu pomáhal již Eberhardovi.

2.1 Triarky a operace s nimi

Definice 4 (Triark). *Triark je takový rovinný graf T , že vrcholy jeho vnější stěny tvoří cyklus C , každý vnitřní vrchol (tj. vrcholy $T - C$) má v T stupeň právě 3 a v C jsou tři navzájem různé vrcholy x, y, z stupně 2 – **rohy**, že vrcholy každé ze tří cest v C , které vzniknou odstraněním rohů z C , mají střídavě stupeň 2 a 3, počínaje i konče stupněm 2.*

Strana triarku je každá z výše zmíněných cest v C , ke které na oba konce připojíme i příslušný roh. **Délka strany** triarku odpovídá počtu jejích vnitřních vrcholů stupně 2 v T . O triarku se stranami délky a, b, c mluvíme jako o (a, b, c) -**triarku**. Poznamenejme, že na pořadí stran v názvu nezáleží (odpovídají rotacím). Později využijeme ještě dalšího značení. **M -triark** má vnitřní strany pouze velikostí z M . U triarku přejímáme některé termíny používané pro trojúhelníky a jejich význam bude vždy intuitivní.

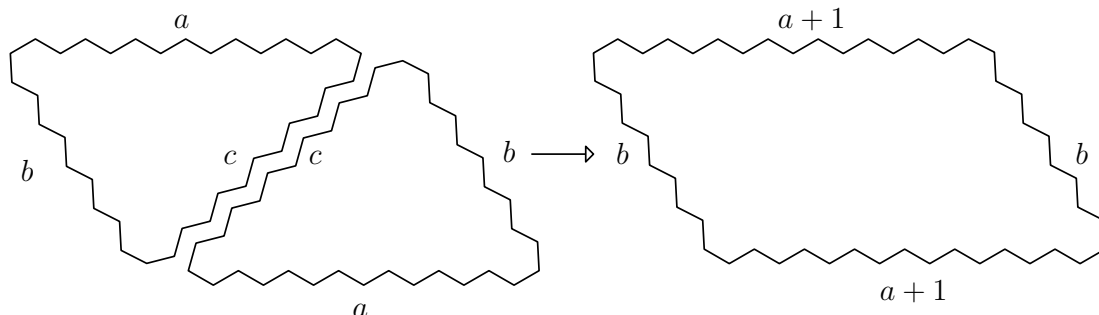
TODO definovat i $(abv)Q$ triatk



Obrázek 2.1: $(4,4,3)\{4,7\}$ -triark.

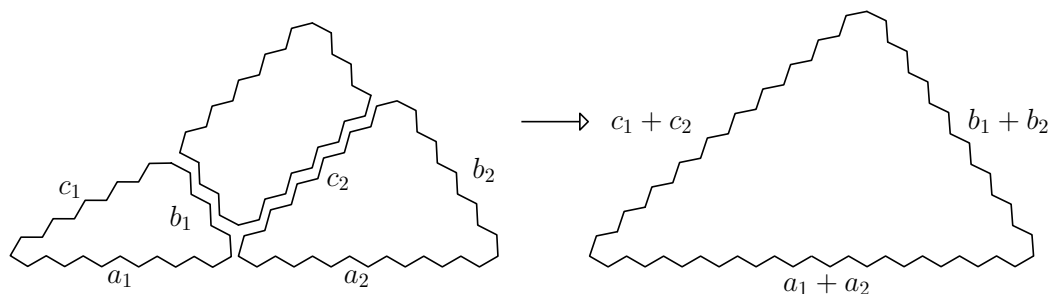
Zmiňme velmi užitečnou vlastnost triarků: pokud máme dva triarky, oba mající stranu stejné délky, můžeme je za tuto stranu slepit jako na obrázku 2.2 a získáme opět graf, jehož vnitřní vrcholy mají stupeň 3. (Při slepování se ztotožní

vždy vrchol stupně 2 s vrcholem stupně 3.) Pokud slepíme triarky tak, že protější strany výsledného grafu budou stejně dlouhé (tedy speciálně při slepení dvou stejných triarků), budeme podle jeho tvaru mluvit o **rovnoběžníku**. Rovnoběžníky jdou navíc slepovat podobně jako triarky a můžeme tím každý již existující rovnoběžník zvětšit na libovolný rozměr, který je násobkem jeho původních rozměrů.



Obrázek 2.2: Spojení dvou triarků za vzniku rovnoběžníku.

Mohli bychom ale chtít spojovat triarky tak, aby výsledkem byl opět triark. Mějme (a_1, b_1, c_1) -triark a (a_2, b_2, c_2) -triark a vhodný rovnoběžník. Slepáním, jako na obrázku 2.3, vznikne $(a_1 + a_2, b_1 + b_2, c_1 + c_2)$ -triark.



Obrázek 2.3: Spojení triarků spolu s rovnoběžníkem za vzniku triarku.

A závěrem budeme chtít spojit dva triarky tak, aby výsledkem byl kubický graf (tedy aby nevznikly žádné nové stěny s vrcholy stupně dva). Graf, který má tuto funkci označíme za **prstenec**. Pro lepší představu si prstenec představujeme jako plášť trojbokého hranolu, triarky jako dolní a boční podstavu. Na všech podstavních hranách dojde ke splynutí vrcholů triarků s vrcholy prstence a vznikne graf nakreslitelný na hranol (tedy tedy i na sféru), takže výsledkem je rovinný graf. Zdefinujme prstenec. Prstenec je souvislý graf, ve kterém můžeme vyznačit dva disjunktní cykly tak, že stupně vrcholů každého cyklu jsou opačné než stupně příslušných triarků, které chceme spojit. Kde opačně znamená záměna dvoj- a tří-vaznosti vrcholů. Pro případné dokreslení můžeme použít obrázek 3.3, na kterém je speciální druh prstence.

TODO provazba na definici prstence, chybí kubičnost a rovinnost

2.2 Důkaz za předpokladu existence pomocných grafů

Převeďme hypotézu ve větu.

Věta 5. *Mějme přípustnou posloupnost $p = (p_k | 3 \leq k \neq 6)$ a neutrální posloupnost $q = (q_k | 3 \leq k \neq 6)$ a následující grafy pro nějaké přirozené k :*

- (i) $(k, k, k)Q$ -triark;
- (ii) $(k, k, k-1)Q$ -triark;
- (iii) *rovnoramenný $Q \cup \{l\}$ -triark, délky jehož stejných stran jsou dělitelné k a který obsahuje právě jednu stěnu velikosti l pro každé nenulové p_l v p , této stěně říkáme **jádro** triarku;*
- (iv) *prstenec, který dokáže spojit dva stejně velké, rovnostranné triarky.*

TODO pro každé t a (t, t, t) triarky

Pak existuje takové přirozené n , že $p + nq$ je realizovatelná.

Myšlenka důkazu pak není příliš složitá: každou stěnu ze zadané posloupnosti p zabalíme do triarku (iii), připravíme si pomocné lepicí a zkrášlující prvky (ii), díky kterým získáme jediný velký rovnostranný triark. K němu zkonstruuujeme ještě jeden se stejně dlouhými (i) a pomocným prstencem je spojíme v kýžený graf (iv). Všechny tyto pomocné objekty jsou totiž (kromě jader) jen ze stěn, které jsou v zadané neutrální posloupnosti, lepicí operace zachovávají kubičnost uvnitř grafu a prstenec pak spojí dva triarky v hledaný kubický graf.

Důkaz. Nejprve slepíme dva $(k, k, k-1)$ -triarky za stěnu délky k a získáme rovnoběžník se všemi stranami délky k . A díky slepování můžeme získat i libovolný rovnoběžník o rozměrech mk, lk pro m, l přirozená. Poté postupně spojujeme jednotlivé jádrové triarky za pomoci příslušného lichoběžníku, dokud nezískáme jediný triark, který obsahuje všechny stěny z P .

Zkusme vzniklý triark upravit na rovnostranný. Při slepování triarků se velikosti výsledných stran rovnají součtům původním. Pokud tedy slepujeme s $(k, k, k-1)$ -triarkem, zmenšíme vždy tu stěnu, na kterou připadne rozměr $k-1$, vůči ostatním. Takže pokud budeme opakovat lepení výsledného triarku, v každém kroku se součet rozdílů mezi stranami zmenší a tedy nutně získáme rovnostranný triark. Modifikujme ho stejnou operací, aby zůstal rovnostranný, ale navíc délka jeho stran byla násobkem k a označme výsledný triark T_1 .

Slepováním (k, k, k) -triarků z (i) spolu s k, k rovnoběžníky získáme druhý triark T_2 o stejném rozměru jako T_1 . Kdybychom celý problém řešili na tóru, stačilo by T_1 a T_2 spojit do rovnoběžníku a sjednotit odpovídající strany. Na kouli místo toho použijeme prstenec z (iv). Tím získáme graf, který je kubický, rovinný a obsahuje požadované stěny. \square

TODO jak se píše tórus

Všimněme si, že větu jde jednoduše zesílit: nejen že za splnění předpokladů existuje nějaké $n \in \mathbb{N}$, pro které je $p + nq$ realizovatelná; existuje jich libovolně mnoho. V důkazu stačí před spojením T_1 a T_2 prstencem nejprve oba triarky slepit s dalšími instancemi T_2 tak, aby vznikly nové, stejně velké rovnostranné triarky T_1^* a T_2^* . Tímto způsobem lze libovolně zvětšit n . T_1^* a T_2^* pak opět spojíme prstencem.

3. Řešítko

Abychom mohli dokončit důkaz některých instancí hypotézy 4, potřebujeme získat požadované stavební bloky. Nabízí se naprogramovat řešítko, které bude umět alespoň některé typy hledaných grafů najít. Hlavním cílem této práce bylo takový program připravit a pomocí něj získat lepší představu o potenciálu uvedené konstrukce důkazu.

3.1 Algoritmus

Program na vstupu očekává zadání vnější stěny: každý vrchol je zastoupen jedním bitem, který určuje, zda má být ve výsledném grafu stupně 2 nebo 3. Navíc očekává seznam velikostí stěn, které má využít. Na výstupu informuje, zda se mu daný graf podařilo najít (říkejme **vyplnit**), a umožní jej exportovat.

Postup hledání původně imitoval lidské pokusy o řešení problému: nakreslit si vnější stěnu, zkusit spojit nějaké dva vrcholy řetízkem vhodné délky (aby nově uzavřená stěna byla z neutrální posloupnosti) a dokud je místo na papíře, spojovat. Pak si překreslit nejvnitřnější, zatím neuzavřenou stěnu (budeme mluvit o **hranici**), ta se stane „vnější stěnou“ na novém papíře a pokračovat. V situaci, kdy nelze dál nic spojit, nebo je jasné, že graf nemůže vyhovovat parametrům, vrátit se podle uvážení zpět.

TODO vizuálně na body?

TOD prolink dopředu na obrázek uфона

Kdybychom chtěli znát jen „ano/ne“ odpověď, jestli graf existuje, nebylo by vůbec třeba si pamatovat celý rozpracovaný graf, stačilo by pracovat s hranicemi, které navíc stačí reprezentovat jako binární číslo. Výsledkem by pak mohla být jen posloupnost hranic, kterými se prošlo před uzavřením grafu, nebo samotné „ano/ne“. Překvapivě obtížné je pak z této posloupnosti nestrojově získat skutečný graf, proto program nabízí i možnost graf dodatečně rekonstruovat podle prošlých stavů.

V tento okamžik je jasné, že problém je vlastně prohledávání v binárních řetězcích (které reprezentují hranice). Je proto vhodné zmínit, podle jakého kritéria se program rozhoduje, kterým směrem hledat dále. Implementace vždy upřednostňuje ke zpracování již nalezený řetězec nejmenší délky, a pro něj najde všechny další sousedy.

Aby toto prohledávání fungovalo dobře, je třeba trochu zkomplikovat reprezentaci hranice. Hlavním požadavkem bude identita mezi reprezentací (jediným binárním řetězcem) a všemi hranicemi (tedy cykly, na kterých vyznačené vrcholy ještě vyžadují dalšího souseda), které jsou pro algoritmus izomorfní – tedy všechny rotace a převrácení hranice.

TODO pseudokód

Definice 5 (Hranice a její reprezentace). *Dvojici cyklus C a množina $I \subseteq V(C)$ nazveme hranicí H . Množina I jsou právě ty vrcholy, které ve výsledném vyplnění musí mít dalšího souseda. Pokud $I = \emptyset$, mluvíme o hranici přímo jako o stěně.*

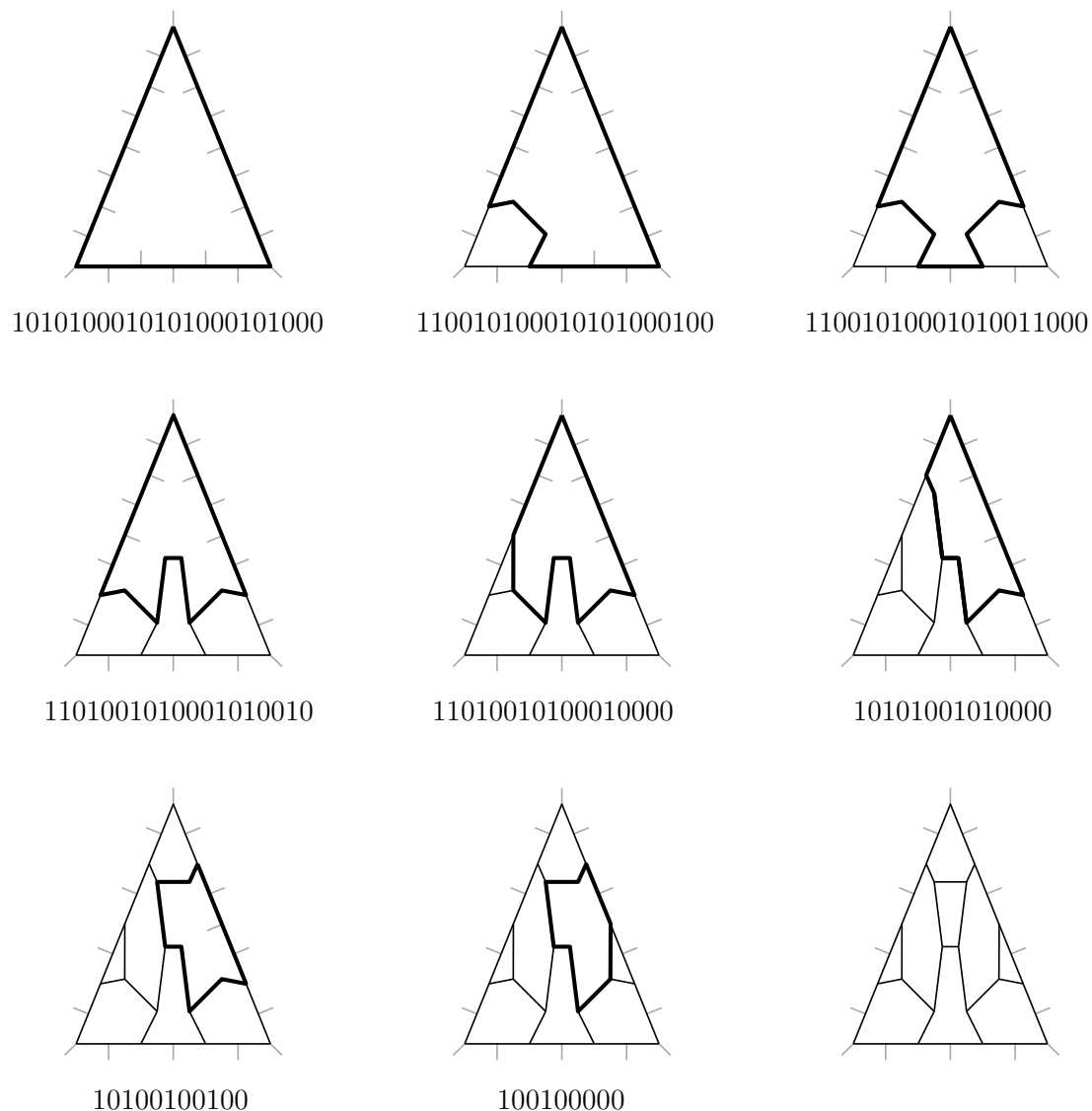
Definujme funkci $f(H, (u, v))$ jako charakteristický vektor množiny I následně: každý vrchol H popíšme buď znakem 1 (jako I v „in“ podle orientace pomyslené

hrany) nebo 0 (jako 0 v „out“). Hodnota $f(H, (u, v))$, kde $H = (C, I)$, $u, v \in V(C)$ a $\{u, v\} \in E(C)$, se pak získá zaznamenáním popisků vrcholů H , začínaje vrcholem u , pokračující vrcholem v a dále po hranách C .

Reprezentací hranice je hodnota $\max_{\{u,v\} \in E(C)} f(H, (u, v))$. Pokud $I = \emptyset$, pak přidejme speciální znak a zapamatujme počet vrcholů.

TODO vyřešit $I = \text{prázdná}$ předem TODO pozorování o začínání jedničkou + počtu řetězců v maximu

Pro lepší představu přikládáme posloupnost hranic s vizualizací na obrázku 3.1, která řeší $(4,4,3)\{4,7\}$ -triark.



Obrázek 3.1: Možný postup vyplnění $(4,4,3)\{4,7\}$ -triarku. Hodnota pod grafem vždy odpovídá reprezentaci aktuální, tučně zvýrazněné, hranice.

Pro jistotu poznamenejme, že pokud program hledaný graf nenašel, může, ale nemusí to znamenat, že neexistuje.

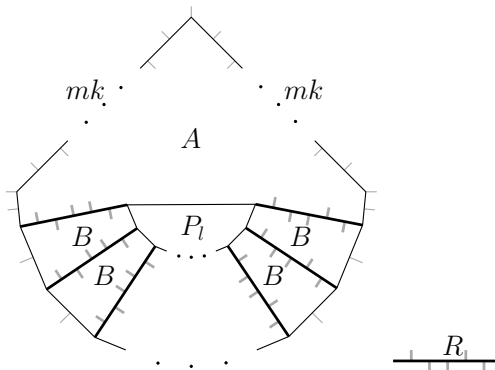
Podle předchozí kapitoly pro dokončení důkazu pro konkrétní dvojici p a q a nějaké přirozené k potřebujeme tyto čtyři typy grafů:

- (i) $(k, k, k), Q$ -triark;

- (ii) $(k, k, k - 1), Q$ -triark;
- (iii) $(mk, mk, x), Q \cup \{l\}$ -triark, kde je právě jedna stěna velikosti l ;
- (iv) prstenec, který dokáže spojit dva stejně velké, rovnostranné triarky.

Pro dané k získáme grafy (i) a (ii) z řešítka hned. Pro zbylé je nutné pomoci si konstrukcí, která se ukázala jako úspěšná pro některé neutrální sekvence. O výsledcích získaných z programu píšeme v kapitole 4.

Na graf (iii) se neumíme zeptat přímo, protože potřebujeme v grafu mít právě jednu stěnu délky p_l . Spojme proto stěnu s vnější stěnou triarku ručně a ptejme se na výplň vzniklých oblastí A a B . Ke spojení použijeme l kopií řetízku R , každý napojíme na jeden z vrcholů jádra a druhé konce spojíme s „in“ vrcholy základny triarku. V řetízku navíc fixujeme, ve které straně od něj budou mít které jeho vrcholy třetího souseda (znázorněno šedě v obrázku 3.2). Konstrukce je obecná pro všechny přípustné hodnoty l , tedy není závislá na volbě p . Navíc není (při dobré volbě R) vynucená ani příliš velká stěna, takže konstrukce může být obecná i pro všechny q (protože q je neutrální, tedy v ní musí být nenulová hodnota na pozici reprezentující stěnu velikosti alespoň 6, tvrzení 3).



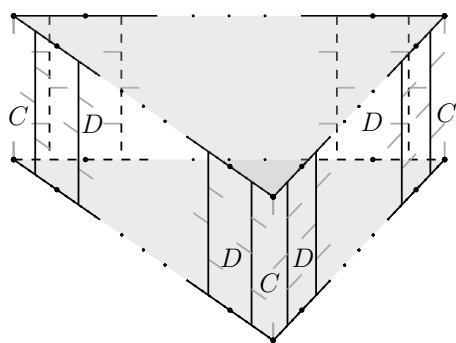
Obrázek 3.2: Konstrukce grafu typu (iii) pomocí řetízků R , konkrétně $(mk, mk, l + 1)$ -triarku s l -úhelníkem P_l jako jádro.

Podobnou konstrukci tvoříme i pro graf typu (iv). V tomto případě za pomoci řetízků spojujeme odpovídající vrcholy rovnostranných, stejně velkých triarků T_1 a T_2 . Vzniknou dva typy oblastí - C při rozích triarku a D mezi odpovídajícími kusy stran triarků. Přesný náčrtek je na obrázku 3.3. Poznamenejme, že pokud jde graf nakreslit na povrch hranolu, je rovinný.

Tyto konstrukce jsou v řešítce implementovány. Výsledný program tedy na vstupu očekává seznam velikostí stěn, které mohou tvořit neutrální posloupnost. Pokud pro daný seznam stěn existuje více neutrálních posloupností, využije program pro každý pomocný graf libovolnou z nich.

3.2 Uživatelská dokumentace/formáty výstupu

TODO?



Obrázek 3.3: Konstrukce grafu typu (iv), slepované triarky jsou vyznačené šedě, prstenec odpovídá plášti.

4. Výsledky

Díky programu popsanému v předchozí kapitole bylo možné zkusit dokončit důkaz věty (4) pro některé neutrální posloupnosti q (na volbě posloupnosti p nezáleží, protože konstrukce pro jádrový triark nezávisí na velikosti jádra).

Vzhledem k výpočetním omezením programu jsme zkusili dokončit důkaz věty pouze pro takové neutrální posloupnosti q , že $Q = \{r, s\}$ a navíc $r < s < 18$. Seznam posloupností, pro které program našel potřebné grafy je v tabulce 4.1. Pokud bychom se omezili na rozsah $r < s < 14$, pak program grafy našel právě tehdy, když r a s jsou nesoudělná čísla.

-	7	8	9	10	11	12	13	14	15	16	17
3	•	•		•	•		•	•		•	•
4	•		•		•		•		•		
5	•	•	•		•	•	•				

Obrázek 4.1: Úplný výčet dvojic stran, pro které se podařilo dokončit důkaz věty (4).

TODO fakt větu jako výsledek!!

Všechny potřebné grafy pro doložení důkazu jsou k práci přiloženy. TODO přiložit

Přirozenou snahou při zkoumání nalezených grafů je grafy zobrazit, aby byly pro člověka dobře čitelné. V další sekci se tomuto tématu krátce věnujeme.

4.1 Kreslení

Přirozenou snahou pro studování nalezených grafů, a pochopení, proč právě soudělnost velikosti stěn zabraňuje v použití navrženého algoritmu k dokončení důkazu, je jejich zobrazení, které je pro člověka dostatečně čitelné. Překvapivě, i přes rovinnost grafů a celkem dobré znalosti jejich struktury není jednoduché hezké rovinné nakreslení najít.

Zamysleme se, jak graf bude vypadat. Vrcholy vnější stěny rozmístíme po kružnici a pak - podle jednotlivých hranic, které graf řeší - vždy vrcholy nově uzavřené stěny nakresleme na soustřednou, menší kružnici tak, aby spojnice žádného bodu hranice se středem kružnic neprotínala jiný bod hranice. Tímto způsobem určitě získáme rovinné nakreslení (ale hrany mohou být libovolné křivky), protože v každém kroku je možné spojit libovolné dva vrcholy, které mohou být v řešení zrovna spojovány, tak, abychom zachovali požadovanou vlastnost tvaru hranice.

Problémem takového nakreslení je počet soustředných kružnic, které bychom potřebovali, který odpovídá počtu stěn grafu. Množství potřebných kružnic by šlo celkem jednoduše snížit. Nově přidávané vrcholy nakreslíme vždy na největší kružnici, na které v příslušné výseči ještě žádný vrchol neleží. Dalším problémem by bylo rozložení vrcholů do výsečí. Představme si třeba zadání, ve kterém značný podíl tvoří souvislá posloupnost „out“ vrcholů. Pokud ve výpočtu dojde k uzavření stěny, která tyto vrcholy obsahuje, až na závěr, bude výseč, ve které leží, jinak zcela prázdná.

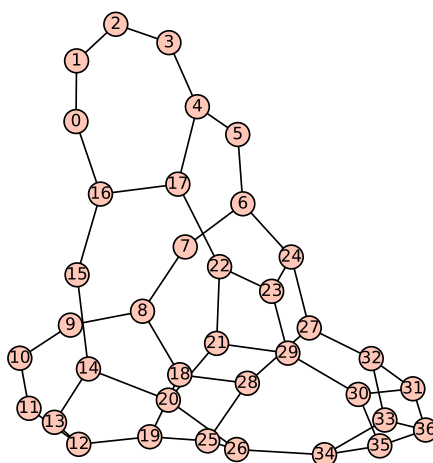
Další možností jsou běžně dostupné programy či funkce na kreslení grafů. Posouzení jejich kvality na náhodném z malých nalezených grafů necháváme na čtenáři.

Nejuspokojivější nalezenou možností je Tuttův (barycentrický) algoritmus. Jak název napovídá, jde o umísťování vrcholů do „těžišť“. Nejprve je třeba rozdělit vrcholy do dvou skupin: pevné a volné. Pevné vrcholy jsou rozestaveny, aby tvořily konvexní n -úhelník. Pozice volných vrcholů se pak dopočítá jako vážený průměr sousedních vrcholů, tedy stačí řešit soustavu lineárních rovnic.

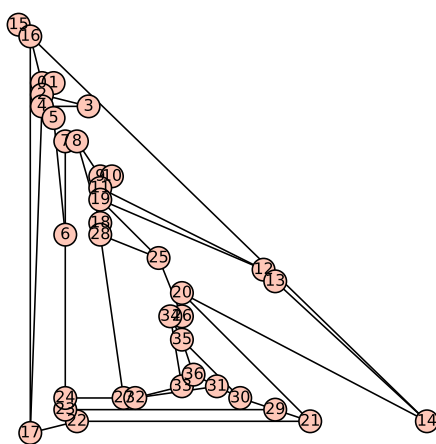
Aby mohl algoritmus dobře fungovat, je nutné, aby graf byl 3-souvislý (že jde i o postačující podmínku ukazuje článek [7]). Pokud by nebyl 3-souvislý, pak vrcholy komponenty, která by po odebrání dvou vrcholů byla oddělena od zbytku grafu a neobsahovala by pevné vrcholy, budou ležet v jedné přímce.

Pro převedení grafu na 3-souvislý stačí do každé vnitřní stěny vložit nový vrchol a spojit ho se všemi vrcholy dané stěny. Ze způsobu, kterým graf vzniká, víme, že je 2-souvislý. Uvažujme nyní situaci po odebrání dvou vrcholů, které způsobí rozpadnutí grafu na více komponent. V každé nově vzniklé komponentě je vrchol, který je nově ve „vnější stěně“ a před odebráním byl ve stěně, ve které je i jiný vrchol než u a v . To znamená, že po přidání vrcholů pro stěny bude jedním z nich spojen s další komponentou a tedy bude 3-souvislý. TODO!! není pravda

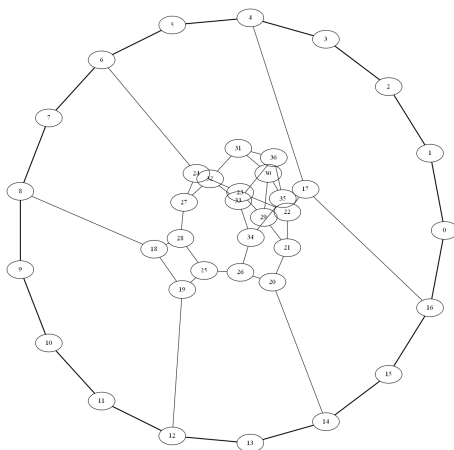
V našem případě za pevné vrcholy volíme vrcholy vnější stěny, které jsou rozmístěné na kružnici a podle konkrétního grafu je pak možné nastavit váhy jednotlivých vrcholů. Obecně se pro dostatečně malé grafy (do 40-ti vrcholů) osvědčila lineární závislost váhy na pořadí přidání vrcholu do grafu, nejdříve přidáný je nejtěžší.



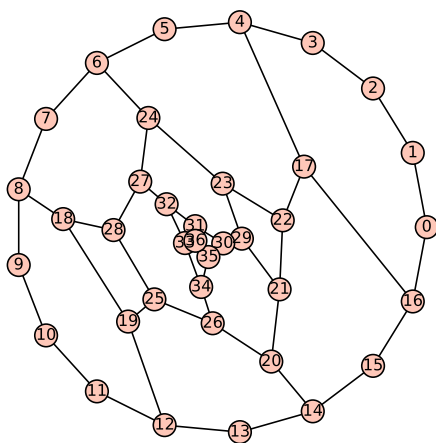
Obrázek 4.2: Automatické nakreslení SageMath.



Obrázek 4.3: Rovinné nakreslení SageMath.



Obrázek 4.4: Neato s předdefinovanými pozicemi vnější stěny.



Obrázek 4.5: Tuttovo nakreslení.

Závěr

V práci se podařilo dokončit důkaz několika instancí hypotézy 4, konkrétně pro neutrální posloupnosti, které mají jen dvě nenulové hodnoty, velikosti jejích stěn jsou nesoudělné a dostatečně malé. Výčet všech dvojic těchto hodnot je v tabulce 4.1.

Kromě tohoto teoretického výsledku poskytujeme implementaci algoritmů z kapitoly 3, který může sloužit při hledání 3-regulárních grafů, speciálně i s konkrétně zadanými velikostmi stěn.

Na základě získaných výsledků bychom v budoucnu rádi zjistili, jestli hypotéza 4 platí i pro nějaké posloupnosti, jejichž množina stěn má soudělné prvky. Také bychom rádi ověřili, že pokud jsou nesoudělné, platí hypotéza vždy.

Vedle hlavního tématu práce bychom rádi získali vhodnější algoritmy pro kreslení rovinných grafů do lidsky dobře čitelné podoby.

Seznam použité literatury

- [1] V. Eberhard. Zur morphologie der polyeder. *Leipzig: B. G. Teubner, 1891*, 1891.
- [2] J. C. Fisher. An existence theorem for simple convex polyhedra. *Discrete Math.* 7, pages 75–97, 1974.
- [3] B. Grünbaum. *Convex Polytopes*. Second Edition. Springer-Verlag, New York, 2003.
- [4] M. DeVos, A. Georgakopoulos, B. Mohar, and R. Šámal. An eberhard-like theorem for pentagons and heptagons. *TODO*, 2009.
- [5] J. Matoušek and J. Nešetřil. *Kapitoly z diskrétní matematiky*. Karolinum, Praha, 2009.
- [6] E. Steinitz. Polyeder und raumeinteilungen. *Encyclopädie der mathematischen Wissenschaften, Band 3??, ??:1–139*, 1922.
- [7] W. T. Tutte. How to draw a graph. *Proceedings of the London Mathematical Society*, s3-13:743–767, 1963.

Seznam obrázků

2.1	$(4,4,3)\{4,7\}$ -triark.	5
2.2	Spojení dvou triarků za vzniku rovnoběžníku.	6
2.3	Spojení triarků spolu s rovnoběžníkem za vzniku triarku.	6
3.1	Možný postup vyplnění $(4,4,3)\{4,7\}$ -triarku. Hodnota pod grafem vždy odpovídá reprezentaci aktuální, tučně zvýrazněné, hranice. .	9
3.2	Konstrukce grafu typu (iii) pomocí řetízků R , konkrétně $(mk, mk, l+1)$ -triarku s l -úhelníkem P_l jako jádro.	10
3.3	Konstrukce grafu typu (iv), slepované triarky jsou vyznačené šedě, prstenec odpovídá pláští.	11
4.1	Úplný výčet dvojic stran, pro které se podařilo dokončit důkaz věty (4).	12
4.2	Automatické nakreslení SageMath.	13
4.3	Rovinné nakreslení SageMath.	14
4.4	Neato s předdefinovanými pozicemi vnější stěny.	14
4.5	Tuttovo nakreslení.	14

Seznam použitých zkratek

A. Přílohy

A.1 První příloha