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

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

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

kde p_k značí počet k-hranných stěn grafu. Výraz (1) je nutnou podmínkou pro rovinnost kubického grafu. Pozoruhodně, počet šestihranných stěn v této podmínce nehraje žádnou roli. Toho si všiml Victor Eberhard a v roce 1981 formuloval a dokázal, že pokud můžeme volit p_6, dokážeme najít graf zadaný počtem ostatních stěn, který je rovinný.

Abychom mohli větu formulovat formálně, je nutné definovat přijatelné posloupnosti: posloupnost $(p_k) = (p_3, p_4, ...)$ kladných celých čísel je přijatelná, pokud existuje rovinný kubický 3-spojitý graf, který má právě p_k k-hranných stěn.

Přidáním podmínky na 3-spojitost získáváme jen grafy, které (podle Steinitzovy věty) odpovídají konvexním 3-polytopům. Ekvivalentně lze tedy přijatelnou posloupnost definovat následovně:

Definice 1 (Přijatelná posloupnost)**.** *Posloupnost* $(p_k) = (p_3, p_4, ...)$ *kladných celých čísel je přijatelná, pokud existuje jednoduchý (vrcholy mají stupeň 3) 3-polytop, který má právě* p_k k*-hranných stěn.*

Formulujme nyní Eberhardovu větu //TODO odkaz:

Věta 1 (Eberhardova věta)**.** *Pro každou posloupnost* $(p_k | 3 \leq k \neq 6)$ *kladných celých čísel, splňující* (1) *existuje taková hodnota* p_6*, že* $(p_k | k \geq 3)$ *je přijatelná.*

1. Pojmy a definice

1.1 sekce 1

2. Konstrukce důkazu

Představme hypotézu, kterou se snažíme ověřit.

Věta 2. *Pro každou přípustnou posloupnost* $p = (p_k|3 \leq k \neq 6)$ *a neutrální posloupnost* $q = (q_k|3 \leq k \neq 6)$*, existuje takové přirozené* n*, že* $p+nq$ *je přijatelná.*

TODO zmínit výjimku pro torus.

V článku TODO odkaz 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.

Definujme nejprve graf, který v důkazu pomáhal již Eberhardovi.

Definice 2 (Triarc)**.** ***Triarc*** *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 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 *triarcu je každá z výše zmíněných cest v C, ke které na oba konce připojíme i příslušný roh.*

Délka strany *triarcu odpovídá počtu jejích vnitřních vrcholů stupně 2 v T.*

O triarcu se stranami délky a, b, c mluvíme jako o ***(a, b, c,)-triarcu****. Poznamenejme, že na pořadí stran v názvu nezáleží (odpovídají rotacím).*

Později využijeme ještě dalšího značení. M***-triarc*** *má vnitřní strany pouze velikostí z* M*.*

Zmiňme velmi užitečnou vlastnost triarců: pokud máme dva triarcy, oba mající stranu stejné délky, můžeme je za tuto stranu slepit a získáme opět graf, jehož vnitřní vrcholy mají stupeň 3. (Při slepovaní se ztotožní vždy vrchol stupně 2 s vrcholem stupně 3.) Podle jeho tvaru budeme mluvit o **rovnoběžníku**.

Mohli bychom ale chtít spojovat (alespoň nějaké) triarcy tak, aby výsledkem byl opět triarc. Mějme (a_1, b_1, c_1)-triarc a (a_2, b_2, c_2)-triarc, kde b_1 a c_2 jsou sudé, a vhodný rovnoběžník(třeba vzniklý spojením dvou vhodných triarců). Slepením vznikne $(a_1 + a_2, b_1 + b_2, c_1 + c_2)$triarc.

TODO prodloužit popis podle výmluvnosti obrázku.

TODO připravit obrázky.

Myšlenka důkazu pak není příliš složitá: každou stěnu ze zadané posloupnosti p zabalíme do triarcu, připravíme si pomocné lepící a zkrášlující prvky, díky kterým získáme jediný velký rovnostranný triarc. K němu zkonstruujeme ještě jeden se stejně dlouhými stranami a pomocným prstencem je spojíme v kýžený graf. Všechny tyto pomocné objekty jsou totiž (kromě středu zabalovacích triarců) jen ze stěn, které jsou v zadané neutrální posloupnosti.

3. Řešítko

Abychom mohli dokončit důkaz věty /TODO odkaz/ nebo alespoň některých jejích instancí, 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.

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. K tomu očekává seznam velikostí stěn, které má využít. Na výstupu informuje, zda se mu daný graf podařilo najít, a je schopný jej exportovat.

Postup hledání původně imitoval lidské pokusy o řešení problému: nakreslí si vnější stěnu, zkusí spojit nějaké dva vrcholy řetízkem vhodné délky (aby nově uzavřená stěna byla z neutrální posloupnosti) a dokud má místo na papíře, spojuje. Pak si překreslí nejvnitřnější, zatím neuzavřenou stěnu (budeme mluvit o **hranici**), ta se stane "vnější stěnou"na novém papíře a pokračuje. Pokud dojde do situace, kdy neumí dál nic spojit, nebo je jasné, že graf nemůže vyhovovat parametrům, vrátí se podle uvážení zpět. Během hledání tedy vůbec není třeba si pamatovat celý rozpracovaný graf, stačí pracovat s hranicemi, které navíc stačí reprezentovat jako binární číslo. Výsledkem by pak mohla být jen posloupnost hranic, kterými se prošlo před uzavřením grafu. 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 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.

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

4. Formát PDF/A

Opatření rektora č. 13/2017 určuje, že elektronická podoba závěrečných prací musí být odevzdávána ve formátu PDF/A úrovně 1a nebo 2u. To jsou profily formátu PDF určující, jaké vlastnosti PDF je povoleno používat, aby byly dokumenty vhodné k dlouhodobé archivaci a dalšímu automatickému zpracování. Dále se budeme zabývat úrovní 2u, kterou sázíme TEXem.

Mezi nejdůležitější požadavky PDF/A-2u patří:

- Všechny fonty musí být zabudovány uvnitř dokumentu. Nejsou přípustné odkazy na externí fonty (ani na „systémové", jako je Helvetica nebo Times).
- Fonty musí obsahovat tabulku ToUnicode, která definuje převod z kódování znaků použitého uvnitř fontu to Unicode. Díky tomu je možné z dokumentu spolehlivě extrahovat text.
- Dokument musí obsahovat metadata ve formátu XMP a je-li barevný, pak také formální specifikaci barevného prostoru.

Tato šablona používá balíček `pdfx`, který umí LATEX nastavit tak, aby požadavky PDF/A splňoval. Metadata v XMP se generují automaticky podle informací v souboru `prace.xmpdata` (na vygenerovaný soubor se můžete podívat v `pdfa.xmpi`).

Validitu PDF/A můžete zkontrolovat pomocí nástroje VeraPDF, který je k dispozici na `http://verapdf.org/`.

Pokud soubor nebude validní, mezi obvyklé příčiny patří používání méně obvyklých fontů (které se vkládají pouze v bitmapové podobě a/nebo bez unicodových tabulek) a vkládání obrázků v PDF, které samy o sobě standard PDF/A nesplňují.

Další postřehy o práci s PDF/A najdete na `http://mj.ucw.cz/vyuka/bc/pdfaq.html`.

Závěr

Seznam obrázků

Seznam tabulek

Seznam použitých zkratek

A. Přílohy

A.1 První příloha