



**MATEMATICKO-FYZIKÁLNÍ
FAKULTA**
Univerzita Karlova

BAKALÁŘSKÁ PRÁCE

Anna Gajdová

Jonesův polynom

Katedra algebry

Vedoucí bakalářské práce: doc. RNDr. Stanovský David, Ph.D.

Studijní program: Matematika

Studijní obor: obecná matematika

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: Jonesův polynom

Autor: Anna Gajdová

Katedra: Katedra algebry

Vedoucí bakalářské práce: doc. RNDr. Stanovský David, Ph.D., Katedra algebry

Abstrakt: Abstrakt.

Klíčová slova: klíčová slova

Title: Jones polynomial

Author: Anna Gajdová

Department: Department of Algebra

Supervisor: doc. RNDr. Stanovský David, Ph.D., Department of Algebra

Abstract: Abstract.

Keywords: key words

Obsah

Úvod	2
1 Definice a vlastnosti Jonesova polynomu	3
1.1 začátek	3
1.2 Definice	3
1.3 Závorkový polynom	4
2 Výpočet Jonesova polynomu	6
2.1 Výpočetní složitost problému	6
2.2 Algoritmus	6
2.2.1 PD notace	6
2.2.2 Výpočet Jonesova polynomu ze závorkového polynomu . .	7
2.2.3 Přímočarý výpočet závorkového polynomu	7
2.2.4 Průběžné rozmotávání	8
2.2.5 Vhodná volba křížení	8
2.2.6 Konečný algoritmus	9
2.2.7 Implementace	9
2.3 Analýza složitosti algoritmu	9
2.3.1 Horní odhad	9
2.3.2 Dolní odhad	10
3 Třetí	11
3.1 Co v ní	11
Závěr	12
Seznam použité literatury	13
Seznam obrázků	14
Seznam tabulek	15
Seznam použitých zkratk	16
A Přílohy	17
A.1 První příloha	17

Úvod

Studium uzlových invariantů a polynomů

1. Definice a vlastnosti Jonesova polynomu

Definice, důkaz ekvivalence definic Předpokládám reudenaistra a pod Studium invariantů, polynomů Skein relation Uzel Link Pro linky? Uzly? Diagramy - popsat?

Ukázat, že ampirichal knot má substituci (wiki říká, že přes kauffman)

Uzel, orientace, link. Vhodné věci tučně

Zmínit původní definici?

1.1 začátek

Při definování Jonesova polynomu je důležité rozlišovat mezi linkem a jeho diagramem, tedy vhodnému rovinnému nakreslení nějaké jeho projekce. Každý uzel má nekonečně mnoho diagramů, ovšem uzlový invariant každému přiřadí stejnou hodnotu (což je blbá věta)

V diagramu orientovaného linku rozlišujeme křížení s kladnou a zápornou orientací, viz obrázek.

Pro popis polynomů uzlech a lincích se používají skein (česky přadeno) vztahy. Skein vztah popisuje, jaká je spojitost mezi polynomy tří linků L_+ , L_- a L_0 , jejichž diagramy jsou identické až na oblast jednoho křížení. V linku L_+ má toto křížení kladnou orientaci, v L_- zápornou a v L_0 je křížení rozpojené, viz obrázek.

1.2 Definice

Definice 1. Jonesův polynom orientovaného linku L je laurentův polynom značený v proměnné \sqrt{t} (tj. polynom v $\mathbb{Z}[\sqrt{t}, \sqrt{t^{-1}}]$), značený $V_K(t)$, který

- je invariantní vůči *** transformacím,
- je normalizovaný, tedy polynom V_\circ triviálního uzlu má hodnotu 1,
- splňuje skein vztah

$$\frac{1}{t}V_{L_+} - tV_{L_-} = (\sqrt{t} + \frac{1}{\sqrt{t}})V_{L_0}$$

Lemma 1. Lemma o tom, jak se spočítá polynom sjednocení unknots. Vzorec

Důkaz. Pro dvě kružnice ze skein, pro dvě to plyne indukci.

□

Poznámka. Z každého diagramu uzlu lze změnou křížení z kladného na záporné či obráceně získat diagram triviálního uzlu. Z každého diagramu linku tedy můžeme změnou křížení získat diagram sjednocení triviálních uzlů, jejichž polynom známe podle předchozího lemmatu. Jonesův polynom každého linku lze tedy díky skein vztahu rekursivně spočítat z jeho libovolného diagramu. Definice je tím pádem korektní.

Definice Jonesova polynomu pomocí skein vztahů není příliš vhodná pro algoritmický výpočet. K němu použijeme ekvivalentní definici založenou na použití tzv. závorkového polynomu (bracket polynomial, Kauffman bracket).

1.3 Závorkový polynom

Závorkový polynom je definovaný pouze pro diagramy neorientovaných linků (tedy nikoli pro samotné linky). Je počítán z jednodušších uzlů.

Definice 2. Závorkový polynom neorientovaného diagramu D , značení $\langle D \rangle$, je Laurentův polynom v proměnné A , definovaný třemi odvozovacími pravidly:

- i. $\langle \bigcirc \rangle = 1$, kde \bigcirc značí diagram s jednou komponentou bez křížení
- ii. $\langle \text{křížení} \rangle = A \langle \text{vert} \rangle + A^{-1} \langle \text{hor} \rangle$, kde křížení značí diagram obsahující křížení, vert je diagram, který je shodný až na dané křížení, které je vertikální rozpojeno a hor je diagram, v němž je křížení rozpojeno horizontálně.
- iii. $\langle D \cup \bigcirc \rangle = (-A^2 - A^{-2}) \langle D \rangle$, kde $D \cup \bigcirc$ značí sjednocení diagramu D a diagramu s jednou komponentou bez křížení.

Důsledek. $\langle \text{křížení opacně} \rangle = A \langle \text{hor} \rangle + A^{-1} \langle \text{vert} \rangle$

Lemma 2. $\langle \text{smyčka} \rangle = A^{-3} \langle \text{odsmýčka} \rangle$ $\langle \text{smyčkanaopak} \rangle = A^3 \langle \text{odsmýčka} \rangle$

Důkaz. Par obrazku

□

Je známý výsledek, že dva diagramy znázorňují stejný lnk (jsou ekvivalentní), pokud mezi nimi existuje série Reidematreových pohybů. Předchozí lemma nám říká, že závorkový polynom není invariantní vůči prvnímu typu Reidemastra. Je ovšem invariantní vůči zbylým typům.

Tvrzení 3. Závorkový polynom je invariantní vůči druhému Reidemastrovi

Důkaz. Par obrazku

□

Důsledek. Závorkový polynom je invariantní vůči třetímu Reidemastrovi.

Důkaz. Par obrazku

□

Potřebujeme tedy nějakou úpravu, aby to fungovalo. Měřit míru zakroucení, writhe.

Definice 3. Zakroucení (writhe) orientovaného diagramu D je součet znamení všech křížení v D , značí se $w(D)$.

Definice 4. Normalizovaný závorkový polynom orientovaného linku L definujeme $X(L) = (-A^3)^{-w(D)} \langle D \rangle$, kde je libovolný diagram linku L .

Korektnost definice plyne z následujícího tvrzení.

Tvrzení 4. *Normalizovaný závorkový polynom je uzlový (linkový) invariant.*

Důkaz. Již víme, že je invariantní vůči dva a tři, podle lemmatu bla bla je i podle jedna a je to hotovo.

□

Věta 5. *Normalizovaný závorkový polynom se substituovanou proměnnou je roven Jonesovu polynomu.*

Důkaz. Jedna sedí podle tvrzení Dva sedí podle definice Tři se musí nějak dokázat

□

Vlastnosti: uzly mají jen celočíselné, amperichal knots,

Zmínit, jaké polynomy jsou zobecněním Jonesova?

Měla bych také říct, že je otevřená otázka, jestli má nějaký neunknot polynom jedna.

2. Výpočet Jonesova polynomu

Je to v tride number P Popis algoritmu, vypocet horniho odhadu, dolni odhad pro nějakou třídu uzlů, na které se to rozbije, skripta z počítačové algebry, důkaz správnosti algoritmu Odhad složitosti?

Bylo by zajímavé identifikovat, pro které typy uzlů je algoritmus efektivní a pro které naopak dosahuje nejhorších výsledků. Rychle na kanonických nakreslených torus uzlu a preclikových uzlu.

Jak se používá velké O ? Už jsem viděla, že můžu použít $O(0.83)$. Ale co s tím spodním? Co problém, co algoritmus.

2.1 Výpočetní složitost problému

Podle (On the computational complexity of the Jones and Tutte polynomials) patří problém určení Jonesova polynomu alternujícího uzlu do třídy složitosti $\#P$, dokonce je tento problém $\#P$ -těžký.

Třída $\#P$ obsahuje problémy, jejichž cílem je určit počet přijímacích cest nedeterministického Turingova stroje, jedná se tedy o rozšíření problémů třídy NP . Například problém $\#SAT$ znamená nejen určit, jestli existuje pravdivostní ohodnocení Boolovské formule, ale i spočítat, kolik takových ohodnocení existuje celkem.

Říct, co z toho plyne. Jako že nemůžu najít lineární algoritmus. Nebo bych jako byla fakt dobrá, kdyby ano.

2.2 Algoritmus

Nejdřív chcu popsat, že to tak jde. Pak dodat pseudokód.

Do jakých detailů? Jak popsat implementaci? Python? Důkaz správnosti - vždy se zastaví. Vstup s počtem křížení. Vstup je PD notace (to až nějak v implementaci). Podkapitola na implementační detaily a vychytávky?

Náš algoritmus dostane na vstupu diagram orientovaného linku s n kříženími zapsaný v PD notaci.

2.2.1 PD notace

PD notace je zápis sestávající ze čtveřice čísel pro každé křížení a jednoznačně popisuje daný diagram. Zápis diagramu v PD notaci se získá následovně: úseky mezi kříženími se očíslovají po směru orientace linku čísly od 1 do $2n$. Každé křížení se označí čtyřmi přilehlými úseky, přičemž se začne úsekem, který do křížení vstupuje spodem, a pokračuje se s úseky navazujícími proti směru hodinových ručiček. Viz obrázek.

Linky

2.2.2 Výpočet Jonesova polynomu ze závorkového polynomu

Jak již bylo řečeno, k výpočtu Jonesova polynomu používáme závorkový polynom. Podle věty se Jonesův polynom získá z normalizovaného závorkového polynomu substitucí proměnné.

Algoritmus 1: Jonesův polynom

Data: Diagram linku L s n křížení

Result: Jonesův polynom v proměnné t

```

závorkovýPolynom  $\leftarrow$  Bracket( $L$ )                                /* v proměnné  $A$  */
zamotání  $\leftarrow$  Writhe( $L$ )
normalizovanýPolynom  $\leftarrow$   $(-A^3)^{\text{zamotání}} \times \text{závorkovýPolynom}$ 
jonesůvPolynom  $\leftarrow$  Substitute(normalizovanýPolynom,  $A$ ,  $t^{1/2}$ )
return jonesůvPolynom

```

V PD notaci lze jednoduše určit, jestli je křížení kladné, či záporné orientace, tedy zamotání spočítáme v $\mathcal{O}(n)$ čase. spočítáme v lineárním čase vzhledem k počtu křížení.

Dále se budeme zabývat výpočtem závorkový polynom.

2.2.3 Přímočarý výpočet závorkového polynomu

Možná to celé nahradit diagramy.

Z definice závorkového polynomu plyne jednoduchý rekurzivní algoritmus.

Budu tomu říkat rozpojení křížení. Synové s $n - 1$.

Háčky jsou možné, ale nesmí být jednoslovné názvy. Polynomy rovnou psát v jaké jsou proměnné.

Algoritmus 2: Závorkový polynom

Function Bracket(L)

Data: Diagram linku s n kříženími

Result: Závorkový polynom v proměnné A

if link L je kruznička **then**

└ return 1

vyber křížení linku L

HL \leftarrow link L , kde křížení je rozpojeno horizontálně

VL \leftarrow link L , kde křížení je rozpojeno vertikálně

if v linku HL vznikla disjunktní kruznička **then**

└ Hk \leftarrow 1

else

└ Hk \leftarrow 0

if v linku VL vznikla disjunktní kruznička **then**

└ Vk \leftarrow 1

else

└ Vk \leftarrow 0

zavorkPoly $\leftarrow A(-A^2 - A^{-2})^{Hk}$ Bracket(HL) + $A^{-1}(-A^2 - A^{-2})^{Vk}$

Bracket(VL)

return zavorkPoly

Závorkový polynom linku s n kříženími se vypočte ze dvou závorkových polynomů linků s $n - 1$ kříženími. Algoritmus má tedy časovou složitost $\mathcal{O}(2^n)$.

Stejnou časovou složitost má i výpočet Jonesova polynomu používající tento postup. Zastaví + koretnost.

2.2.4 Průběžné rozmotávání

Algoritmus na výpočet závorkového polynomu zrychlíme, pokud se link pokusíme v každém kroku rozmotat, tedy pokud nalezneme diagram ekvivalentního uzlu s menším množstvím křížení.

V PD notaci jsou snadno naleznutelné případy, kdy lze link rozmotat použitím prvního či druhého Reidemastrova pohybu.

Při použití prvního Reidemastrova pohybu odmotáme jednu smyčku a zbavíme se jednoho křížení, ovšem výsledný závorkový polynom se změní o mocninu A^3 podle lemmatu.

Použitím druhého Reidemastrova pohybu se zbavíme dvou křížení a polynom zůstane podle lemmatu stejný.

Algoritmus 3: Závorkový polynom s rozmotáváním

Function Bracket(L)

```

Rozmotej link L prvním Reid pohybem
if neco rozmotano then
    e ← součet znamének rozmotaných křížení
    return  $A^{3e}$  Bracket(rozmotaná link)
Rozmotej link L druhým Reid pohybem
Jeste jednou rozmotej link L prvním Reid pohybem
if neco rozmotano then
    e ← součet znamének rozmotaných křížení
    return  $A^{3e}$  Bracket(rozmotaná link)
Jeste kolik vzniklo samostatných kružnic . . .
return zavorkPoly

```

Zastaví + koretnost. Rozmotávání běží v lineárním čase vzhledem k n , celková časová složitost tedy zůstává $\mathcal{O}(2^n)$.

2.2.5 Vhodná volba křížení

Tady nějak nezáleží na orientaci.

Algoritmus dále můžeme zlepšit vhodnou volbou křížení, které rozpojíme. Dokážeme, že ke vzniku linku, který lze částečně rozmotat způsobem popsáním v předchozí části, je vždy potřeba rozpojit nejvýše dvě křížení. Algoritmus bude volit právě ta křížení, jejichž rozpojení nám v dalších krocích zajistí možnost rozmotání.

Diagram jako rovinný graf

Každý linkový diagram odpovídá rovinnému grafu, v němž křížení představují vrcholy (vždy stupně čtyři) a úseky mezi kříženími hrany. Diagram s n kříženími odpovídá grafu s n vrcholy a $2n$ hranami. Dále budeme v této sekci k popisu diagramů používat grafovou terminologii. Předpokládejme také, že pracujeme s diagramem, který už je rozmotaný ve smyslu rozmotávání v sekci bla.

Eulerova formule pro rovinné grafy říká, že $v - e + f = 2$, kde v značí počet vrcholů, e počet hran a f počet stěn.

V našem případě tedy dostáváme vzorec pro počet stěn $f = n + 2$.

Každá hrana náleží dvěma stěnám, rozdělujeme tedy $4n$ hran mezi $n + 2$. Z toho plyne, že musí existovat stěna, která je ohraničená méně než čtyřmi hranami.

Typy stěn

Stěna s jednou hranou by v linku odpovídala smyčce, ty jsou ovšem podle předpokladu už odstraněny rozmotáním.

Stěna se dvěma hranami, která není rozmotatelná, musí v diagramu odpovídat jedné ze situací na obrázku. Všimněme si, že rozpojením křížení a_1 i a_2 buď ve vertikálním, nebo horizontálním směru vznikne smyčka. Volbou křížení a_1 nebo a_2 je tedy zaručeno, že jeden syn má po rozmotání nejvýše $n - 2$ křížení.

Stěna se třemi hranami v diagramu odpovídá buď typu B, nebo C zobrazeným na obrázku.

Ve stěně typu B získáme horizontálním rozpojením křížení b_1 syna, jenž jde rozmotat druhým Reidematrovým pohybem. Jeho diagram tedy bude mít nejvýše $n - 3$ křížení.

Ve stěně typu C není rozpojením žádného křížení rozmotání zaručeno. Ovšem rozpojením kterého koli křížení získáme jednoho syna se stěnou typu A. Existuje tedy prasek s nejvýše $n - 3$ kříženími.

Jelikož chceme maximalizovat rozmotání synů, jsou preference algoritmu na volbu křížení $b_1 > a_i > c_i$. Z předchozího rozboru plyne, že alespoň jedno z těchto křížení musí existovat.

V PD notaci je možné toto křížení nalézt v lineárním čase vzhledem k n .

2.2.6 Konečný algoritmus

Pseudokód toho celého.

Vždy se zjevně zastaví. Je korektní, bo to platilo pořád. Každopádně tady uvést pseudokód.

2.2.7 Implementace

Python? Práce, záludnosti? Rozdělování? Rychlá práce s PD notací? Jak jsem toho docílila?

2.3 Analýza složitosti algoritmu

2.3.1 Horní odhad

Rychlost algoritmu je závislá na počtu křížení n a na míře rozmotávání. Například na diagramu, který je možné absolutně rozmotat použitím prvních dvou Reidemastrových pohybů poběží nejhůř v kvadratickém čase. Zdůvodnit?

Horní odhad rychlosti tedy provedeme na případu diagramu, v němž dojde k pouze k minimálnímu rozmotávání. Tedy nejdode k rozmotání jiných křížení než těch, která jsou zaručená rozbořem v sekci Volba vhodného křížení. Také v

případech, kdy není předchozí volbou křížení zaručena existence stěny typu A, bude existovat pouze stěna typu C, která zaručuje rozmotání pouze u vnuka.

To se napíše do stromu.

Z toho se udělá rekurentní vzorec.

Z toho vypočítáme, jak to vychází.

Algoritmus má tedy časovou složitost $O(\dots)$

2.3.2 Dolní odhad

————— Analýza jak to jde.

Dolní odhad Kolečka.

Plyne z toho složitost?

Citovat PD notaci

3. Třetí

3.1 Co v ní

Experiment, náhodné uzly, experimenty na jiných uzlech, různé algoritmy? Zatím určitě nechat proběhnout na dvanácti uzlech, Pak na velkých náhodných uzlech Pak na nějakých speciálních uzlech podle mého uvážení Jak generuji náhodné uzly, alternující uzly, linky,... Bylo by zajímavé identifikovat, pro které typy uzlů je algoritmus efektivní a pro které naopak dosahuje nejhorších výsledků.

Rychlý na torus: vzít těch 36. Pokusit se naprogramovat ta spojení koleček.

Nezapomenou dělat závěry.

Co náhodné alternující uzly?

Neměl by být experiment pouze na náhodných alternujících uzlech?

Udělám oboje dohromady!

Takže zvládne můj generátor ještě tohle? Já už nevím, jak to dělal!

Ale řekla bych, že by to mohl zvládnout.

Závěr

Nezapomenout dělat závěry průběžně. Shrnovat to. Já vím, co to znamená, ale oni ne! Takže do toho.

Takže hlavně shrnovat ty složitosti. Že to na některých uzlech běží dobře. Nachám to běžet na torus.

Myslet meta - co mi tak ještě chybí, aby to dobře shrnulo Jonesův polynom? Jakože je to práce o něm? Nebo mám jenom studovat?

Seznam použité literatury

- ANDĚL, J. (1998). *Statistické metody*. Druhé přepracované vydání. Matfyzpress, Praha. ISBN 80-85863-27-8.
- ANDĚL, J. (2007). *Základy matematické statistiky*. Druhé opravené vydání. Matfyzpress, Praha. ISBN 80-7378-001-1.
- COX, D. R. (1972). Regression models and life-tables (with Discussion). *Journal of the Royal Statistical Society, Series B*, **34**(2), 187–220.
- DEMPSTER, A. P., LAIRD, N. M. a RUBIN, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B*, **39**(1), 1–38.
- GENBERG, B. L., KULICH, M., KAWICHAJ, S., MODIBA, P., CHINGONO, A., KILONZO, G. P., RICHTER, L., PETTIFOR, A., SWEAT, M. a CELENTANO, D. D. (2008). HIV risk behaviors in sub-Saharan Africa and Northern Thailand: Baseline behavioral data from project Accept. *Journal of Acquired Immune Deficiency Syndrome*, **49**, 309–319.
- KAPLAN, E. L. a MEIER, P. (1958). Nonparametric estimation from incomplete observations. *Journal of the American Statistical Association*, **53**(282), 457–481.
- LEHMANN, E. L. a CASELLA, G. (1998). *Theory of Point Estimation*. Second Edition. Springer-Verlag, New York. ISBN 0-387-98502-6.
- STUDENT (1908). On the probable error of the mean. *Biometrika*, **6**, 1–25.

Seznam obrázků

Seznam tabulek

Seznam použitých zkratek

A. Přílohy

A.1 První příloha