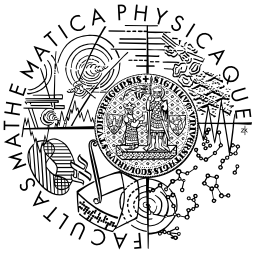


MATEMATICKO-FYZIKÁLNÍ FAKULTA
Univerzita Karlova

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

Studium uzlových invariantů a polynomů
co v které kapitole
Zmínit ty první dvě knihy A článek A tak

1. Definice a vlastnosti Jonesova polynomu

Vhodné věci kurzívou, Zmínit původní definici?, Nekonzistentní používání V(t), V? Dvojtečky po platí

Původní definice Jonesova polynomu vycházela z operátorových algeber. A že tady to je na lincích.

Zapracovat, že je to z těch dvou knih a článku. Zmínit původní paper?

1.1 Základní pojmy

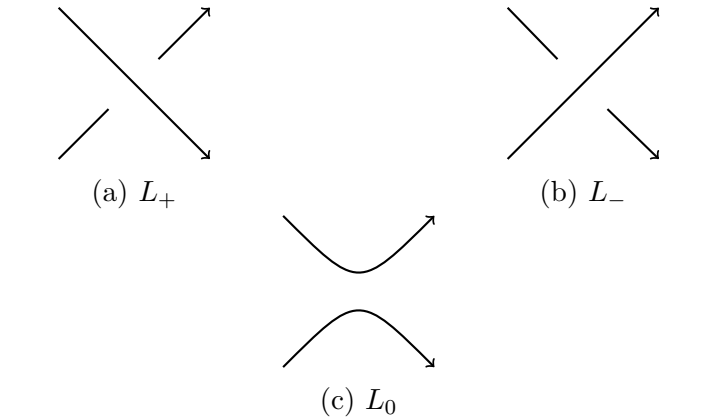
Jonesův polynom je invariant nejen uzlů, ale také linků, tedy více propletených uzlů. Pokud není řečeno jinak, pracujeme v textu s linky. Při definování Jonesova polynomu je důležité rozlišovat mezi linkem a jeho diagramem. Diagram je vhodné rovinné nakreslení nějaké linkové projekce, v němž je rozlišeno, jestli křížení vedou zvrchu nebo zespodu. Každý link má nekonečně mnoho diagramů.

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

(a) Kladná orientace (b) Záporná orientace

Obrázek 1.1: Orientace křížení

Pro popis polynomů na uzlech a lincích se často používají *skein vztahy* [1]. Skein vztahy určují, 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.



(a) L_+ (b) L_-

(c) L_0

Obrázek 1.2: Diagramy skein vztahu

[1] česky přadenové vztahy

1.2 Definice

Definice 1. Jonesův polynom *orientovaného linku L je Laurentův polynom v proměnné $\sqrt{t}$ (tj. polynom v $\mathbb{Z}[t^{1/2}, t^{-1/2}]$), značený $V_L(t)$, který:*

1. *je linkový invariant,*
2. *je normalizovaný, tedy polynom $V_{\circlearrowleft}$, kde $\circlearrowleft$ je orientovaný triviální uzel, má hodnotu* 1,
3. *splňuje skein vztah*

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

Lemma 1. *Buď L link, který se skládá z k neprotínajících se orientovaných triviálních uzlů. Pak pro Jonesův polynom linku L platí:*

$$V_L(t) = (-\sqrt{t} - \frac{1}{\sqrt{t}})^{k-1}.$$

Důkaz. Libovolně orientované triviální uzly jsou ekvivalentní. Vzorec tedy stačí dokázat pro diagram skládající se z k libovolně orientovaných disjunktních kružnic, použijeme matematickou indukci.
Pro $k = 1$ vzorec platí podle druhé podmínky v definici 1.
Předvedeme i případ, kdy $k = 2$. Pak $L_0 =$  , $L_- =$  a $L_+ =$  . Diagramy L_+ a L_- zobrazují triviální uzly, takže $V_{L_+} = V_{L_-} = 1$. Použitím skein vztahu získáme

$$V_L = V_{L_0} = -\sqrt{t} - \frac{1}{\sqrt{t}}.$$

Pro $k > 2$ jsou L_- a L_+ diagramy linků s $k - 1$ kružnicemi, z indukčního předpokladu a ze skein vztahu získáme vzorec

$$V_L = V_{L_0} = (-\sqrt{t} - \frac{1}{\sqrt{t}})^{k-1}.$$

□

Poznámka. Z každého uzlového diagramu lze změnou několika křížení vedených zvrchu na křížení vedených zespodu získat diagram triviálního uzlu. Z každého diagramu linku tedy můžeme změnou křížení získat diagram sjednocení triviálních uzlů, jejichž Jonesův polynom je podle předchozího lemmatu známý. Jonesův polynom každého linku lze tedy pomocí skein vztahu rekurzivně spočítat z jeho libovolného diagramu. Z toho plyno korektnost a jednoznačnost definice.

Definice Jonesova polynomu pomocí skein vztahů není vhodná pro algoritmický výpočet, neboť rozpoznání, jestli diagram odpovídá triviálnímu uzlu, je složitý problém. K výpočtu využijeme ekvivalentní definici založenou na použití *závorkového polynomu.*

1.3 Závorkový polynom

Závorkový polynom [2] je definován pouze pro diagramy neorientovaných linků, nikoli pro samotné linky.

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

1. $\langle \bigcirc \rangle = 1$, *kde* $\bigcirc$ *značí diagram s jednou komponentou bez křížení,*
2. $\langle \times \rangle = A\langle)(\rangle + A^{-1}\langle \asymp \rangle$, *kde* $\times$ *značí diagram obsahující toto křížení;* $)($ *je diagram, který je s ním shodný až na dané křížení, které je zde vertikální rozpojeno; a* $\asymp$ *je diagram, v němž je křížení rozpojeno horizontálně,*
3. $\langle D \cup \bigcirc \rangle = (-A^2 - A^{-2})\langle D\rangle$, *kde* $D \cup \bigcirc$ *značí sjednocení diagramu D a diagramu s jednou komponentou bez křížení.*

Poznámka. Pokud vztah v bodě *ii.* předchozí definice otočíme o 90°, získáme vztah $\langle \times \rangle = A\langle \asymp \rangle + A^{-1}\langle)(\rangle$.

Lemma 2. *Pro závorkové polynomy linků, jejichž diagramy obsahují smyčku, platí:*

1. $\langle \supset \rangle = -A^{-3}\langle) \rangle$
2. $\langle \subset \rangle = -A^{3}\langle) \rangle$

Důkaz. Tady formátovat rovnátka Použitím odvozovacích pravidel dokážeme první bod:
$\langle \supset \rangle = A\langle) \cup \bigcirc\rangle + A^{-1}\langle) \rangle = A(-A^2 - A^{-2})\langle) \rangle + A^{-1}\langle) \rangle = -A^{-3}\langle) \rangle$
Druhý bod se dokáže analogicky.

□

Dva diagramy znázorňují stejný link (jsou ekvivalentní), pokud mezi nimi existuje série Reidemeisterových pohybů (kde dokázané?). Z lemmatu 2 plyne, že závorkový polynom není invariantní vůči Reidemeisterovu pohybu typu I. Ukážeme, že je invariantní Reidemeisterovým pohybům typu II a III.

Tvrzení 3. *Závorkový polynom je invariantní vůči Reidemeisterovým pohybům typu II a III.*

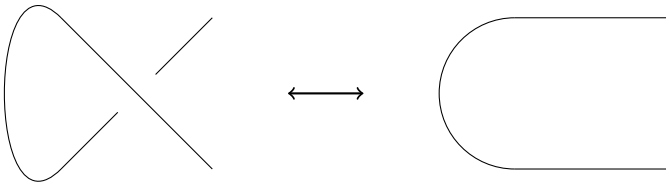
Důkaz. fsfsdfs

□

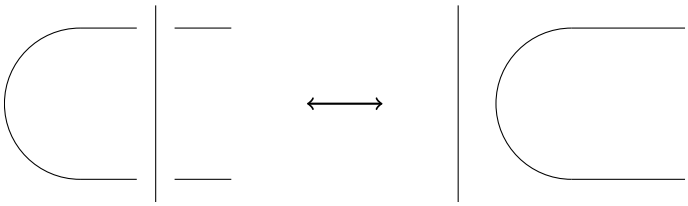
Aby byl závorkový polynom invariantní i vůči Reidemeisterovu pohybu typu I, je nutné vynásobit ho výrazem, který vyjadřuje míru zakroucení.

Definice 3. Zakroucení *(writhe) orientovaného diagramu D je součet znamení všech křížení v D. Zakroucení značíme $w(D)$.*

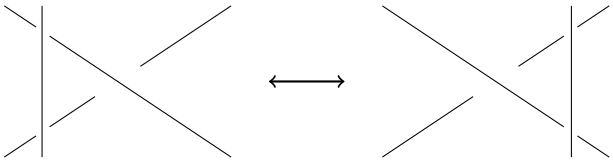
[2] anglicky bracket polynomial nebo Kauffman bracket



(a) Typ I



(b) Typ II



(c) Typ III

Obrázek 1.3: Reidemeisterovy pohyby

Lemma 4. *Zakroucení je invariantní vůči Reidemeisterovým pohybům typu II a III.*

Důkaz. Důkaz se provede rozborem případů.

□

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

Definice je korektní, neboť následující tvrzení ukazuje, že nezáleží na volbě diagramu.

Tvrzení 5. *Normalizovaný závorkový polynom je linkový invariant.*

Důkaz. Závorkový polynom i zakroucení jsou invariantní vůči Reidemeisterovým pohybům typu II a III, invarientní je tedy i jejich součin.

Invariance vůči typu I plyne z Lemmatu 2 a faktu, že křížení v  je vždy

kladné a křížení v  záporné.

□

Věta 6. *Při substituce proměnné $A = t^{1/4}$ je normalizovaný závorkový polynom $X_L(A)$ roven Jonesovu polynomu $V_L(t)$.*

Důkaz. Ověříme, že $X_L(t^{1/4})$ splňuje podmínky v Definici 1.

1. Podle předchozího tvrzení je linkový invariant.
2. Zamotání triviální uzlu $w(\bigcirc) = 0$, tedy $X_{\circlearrowleft} = (-A^3)^{w(\bigcirc)}\langle \bigcirc \rangle = 1$
3. fd

□

Vlastnosti: uzly mají jen celočíselné, Amphichiral knots,
Zminit, jake polynomy jsou zobecněním Jonesova?
Měla bych také říct, že je otevřená otázka, jestli má nějaký ne unknot polynom jedna.

2. Výpočet Jonesova polynomu

Je to v tride number P Popis algoritmu, vypocet horniho odhadu, dolní 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 kanonickych nakreslenich torus uzlu a preclikovych 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

Jaeger, Vertigan a Welsh (1990) dokázali, že problém určení Jonesova polynomu alternujího uzlu patří 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

Nejdriv 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 očíslují 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 ← `Bracket(`L`)` `/* v proměnné` A `*/`
zamotání ← `Writhe(`L`)`
normalizovanýPolynom ← $(-A^3)^{\text{zamotání}} \times$ závorkovýPolynom
jonesůvPolynom ← `Substituce(`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 jake jsou proměnné.

Algoritmus 2: Závorkový polynom

Function `Bracket(`L`)`
  Data: Diagram linku s n kríVL zeními
  Result: Závorkový polynom v promenne A
  if *link L je kruznice* **then**
    return 1
  vyber krizeni linku L
  HL ← link L , kde krizeni je rozpojeno horizontálne
  VL ← link L, kde krizeni je rozpojeno vertikálne
  if *v linku* HL *vznikla disjunktni kruznice* **then**
    Hk ← 1
  else
    Hk ← 0
  if *v linku* VL *vznikla disjunktni kruznice* **then**
    Vk ← 1
  else
    Vk ← 0
  zavorkPoly ← $A(-A^2 - A^{-2})^{\text{Hk}}$ `Bracket(HL)` $+ A^{-1}(-A^2 - A^{-2})^{\text{Vk}}$ `Bracket(VL)`
  return zavorkPoly

Závorkový polynom linku s n křížením se vypočte ze dvou závorkových polynomů linků s $n-1$ křížením. 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 smyčkua zbavíme se jednoho křížení, ovšem výsledný závorkový polynom se změní o mocninu A^3 podle lemmatu.

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

Algoritmus 3: Závorkový polynom s rozmotáváním

```
Function Bracket(L)
    Rozmotej link L prvnim Reid pohybem
    if neco rozmotano then
        e ← soucet znaminek rozmotanych krizeni
        return A^3e Bracket(rozmotany link)
    Rozmotej link L druhym Reid pohybem
    Jeste jednou rozmotej link L prvnim Reid pohybem
    if neco rozmotano then
        e ← soucet znaminek rozmotanych krizeni
        return A^3e Bracket(rozmotany link)
    Jeste kolik vzniklo samostatnych kruznic . . .
    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 popsaným 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 prasyn 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

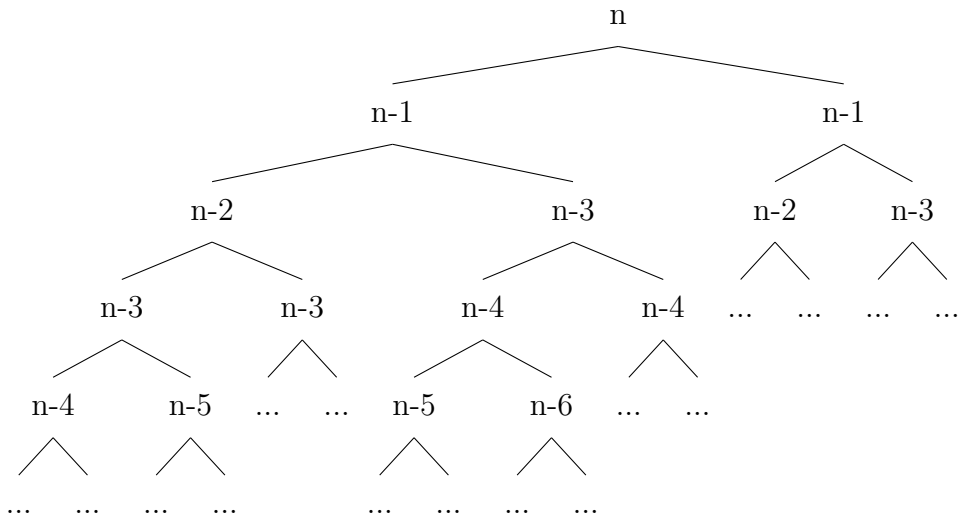
2.3.1 Horní odhad

Rychlost algoritmu je závislá na počtu křížení n a na míře rozmotávávaní. Například na diagramu, který je můžné absolutně rozmotat použitím prvních dvou Reidemasterových 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 rozborem 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.

Průběh algoritmu zakreslíme jako binární strom, kde číslo ve vrcholu je počet křížení diagramu. V kořeni máme diagram s n kříženími, v němž rozmotáme křížení stěny C, takže jeho synové jsou diagramy s $n-1$ kříženími obsahující stěnu typu A. Jejich synové budou tedy diagramy velikosti $n-2$ a $n-3$. V lichých hladinách stromu má tedy vrchol syny o jedna menší, v sudých hladinách je jeden syn o dva menší.



Získáme rekurentní rovnici na počet operací výpočtu závorkového polynomu diagramu velikosti n.

$$T(n) = 2T(n-2) + 2T(n-3) + C_L n + 2C_L(n-1)$$

Člen $C_L n + 2C_L(n-1)$, kde C_L je konstanta, vyjadřuje, že na rozdělení každého diagramu na syny (rozmotání a volba křížení) je potřeba lineární počet operací vzhledem k počtu křížení.

Rekurence lze řešit pomocí vytvořujících funkcí a rozkladu na parciální zlomky. Výpočtem získáme vzorec

$$T(n) \approx C_1 2^{0.82n} + C_2 n 2^{0.09n} + C_3 2^{0.09n} + C_4 n + C_5 = \mathcal{O}(2^{0.82n}),$$

kde C_i jsou jisté konstanty.

Podle sekce je na výpočet Jonesova polynomu ze závorkového pouze lineární počet operací. Dohromady má tedy náš algoritmus na výpočet Jonesova polynomu časovou složitost $\mathcal{O}(2^{0.82n})$.

2.3.2 Dolní odhad

Provedeme analýzu průběhu algoritmu na linku L s diagramem, který vznikne z diagramu Borromeovských kruhů přidáváním kružnic. Kružnice přidáváme tak, aby vždy protla dva další kruhy celkem ve čtyřech bodech a vzniklý diagram byl alternující, tj. aby se střídala křížení vedená zdola a zvrchu. Příklad na obrázku.

Diagram skládající se z k kružnic má $4k-6$ křížení.

Na obrázku je znázorněn možný průběh výpočtu závorkového polynomu na tomto diagramu takový, že jsou postupně odpojovány krajní kružnice.

Z obrázku plyne, že časová složitost algoritmu na diagramu s n kříženími splňuje rekurenci

$$T(n) = 8T(n-4) + C_1 n + C_0$$

pro jisté konstanty C_0, C_1.

A tedy $T(n) = \Omega(8^{\frac{n}{4}}) = \Omega O(2^{0.75n})$. Nebo i tady omega? To velké o je tu divné.

Dokázali jsme, že náš algoritmus na výpočet Jonesova polynomu má časovou složitost $\Omega(2^{0.75n})$.

3. Testování algoritmu na datech

V této sekci představíme výsledky implementace výpočtu Jonesova polynomu. Změříme délku běhu algoritmu a jeho variant na malých uzlech, náhodých uzlech a lincích a na speciálních typech uzlů.

Varianty

3.1 Malé tabulkové uzly

Algoritmy jsme vyzkoušely na všech uzlech, které mají projekci s maximálně dvanácti kříženími. PD notace uzlů byla získána z databáze KnotInfo. U všech tří algoritmů byl už u malých uzlů znatelný exponenciální růst doby běhu (viz Obrázek 3.1).

Ze srovnání průměrných dob běhu plyne, že nejlepší čas dosahuje algoritmus A, těsně následuje algoritmus RND a nejpomalejší je algoritmus B (viz Obrázek 3.2). Podobné výsledky se budou objevovat i nadále.

3.2 Náhodné uzly a linky

3.2.1 Generování náhodných linků a uzlů

Náhodné linky budeme generovat pomocí rovinných grafů, neboť mezi linky a rovinnými grafy existuje vzájemně jednoznačná korespondence, zdroj?. (Jedná se o jinou korespondenci, než tu mezi linky a rovinnými grafy s vrcholu stupně čtyři popsanou v sekci **).

Převod rovinného grafu na link

Rovinný graf z n hranami odpovídá linku s n kříženími. Každé hraně přiřadíme náhodně kladné, či záporné znamení a umístíme na ni křížení příslušného typu. Úseky mezi kříženími jsou tím již jednoznačně určené: musí spojovat křížení mezi nejbližšími hranami tak, aby nevznikla žádná další křížení.

OBRÁZEK, KRESLENÝ RUKOU

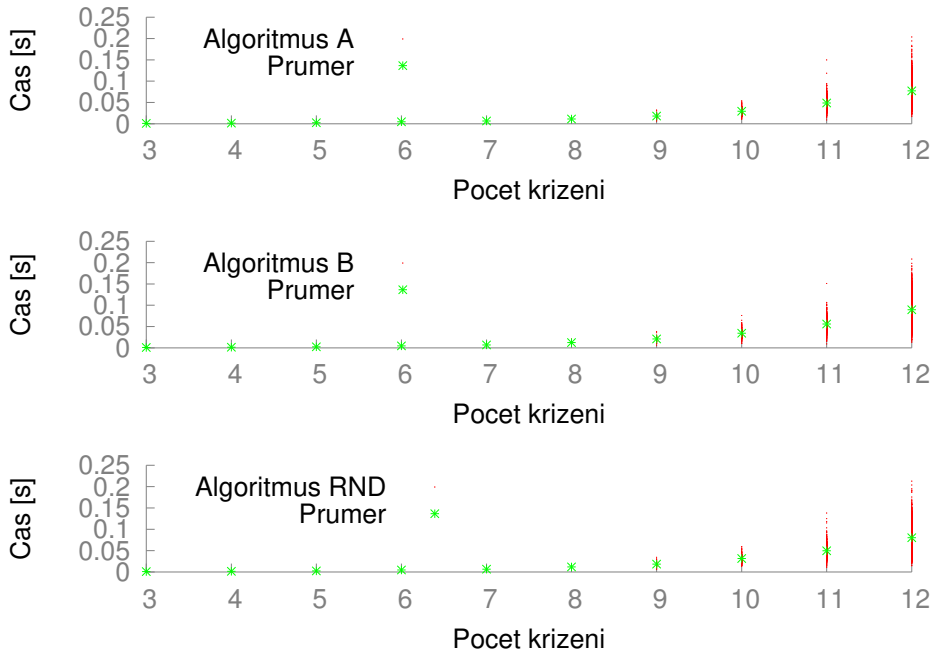
Generování náhodných rovinných grafů

Graf s n hranami získáme následujícím způsobem. Vygenerujeme vhodný počet náhodných bodů v rovině a nalezeneme jejich triangulaci, tj. spojíme body hranami tak, aby byl polygon tvořící konvexní obal bodů rozdělen na trojúhelníky.

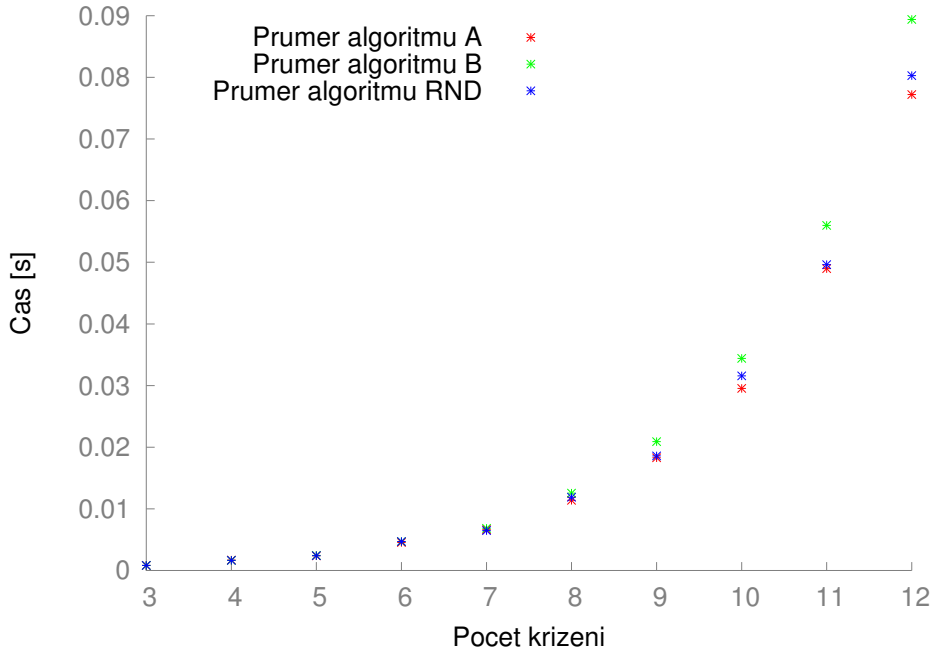
Získali jsme tak rovinný graf s původními body jako vrcholy. Pokud je počet hran menší než n, provedeme triangulaci znovu s větším počtem bodů. Pokud je počet hran větší než n, odstraníme potřebný počet náhodně zvolených hran.

Implementace

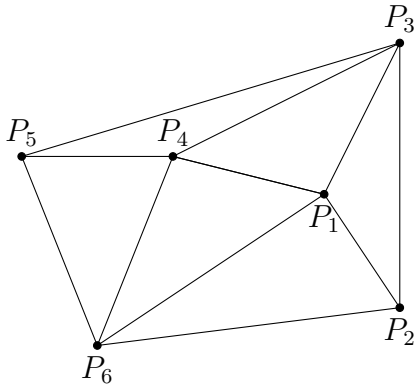
Triangulace bodů je snadno implementovatelný geometrický problém.
Se získaným rovinným grafem pracujeme jako s množinou vrcholů a k nim přísluš-



Obrázek 3.1: Doba běhu algoritmů na tabulkových uzlech do 12 křížení s vyznačenými průměry.



Obrázek 3.2: Průměrné časy algoritmů na tabulkových uzlech do 12 křížení.



Obrázek 3.3: Triangulace šesti bodů

ným hranám seřazeným v pořadí, jak k danému vrcholu přiléhají. Z této struktury je již možné získat PD notaci příslušného linku.

V PD notaci lze procházkou po vláknu snadno poznat, jestli je vygenerovaný link uzlem. Také jsme v PD notaci jednoduchou operací schopni uzel změnit na alternující, tedy takový, v němž se střídají křížení vedená zespodu a zvrchu.

Jsme tedy schopni nagenerovat uzly, alternující uzly a linky libovolné velikosti.

Takto generované uzly jsou také poměrně „zamotané“, tedy rozmotávací krok algoritmu neodstraní příliš mnoho křížení.

3.2.2 Test

Rychlost algoritmů jsme tedy otestovali na uzlech (Obrázek 3.4) , alternujících uzlech (Obrázek 3.5) a lincích (Obrázek 3.6) s počtem křížení do 46.

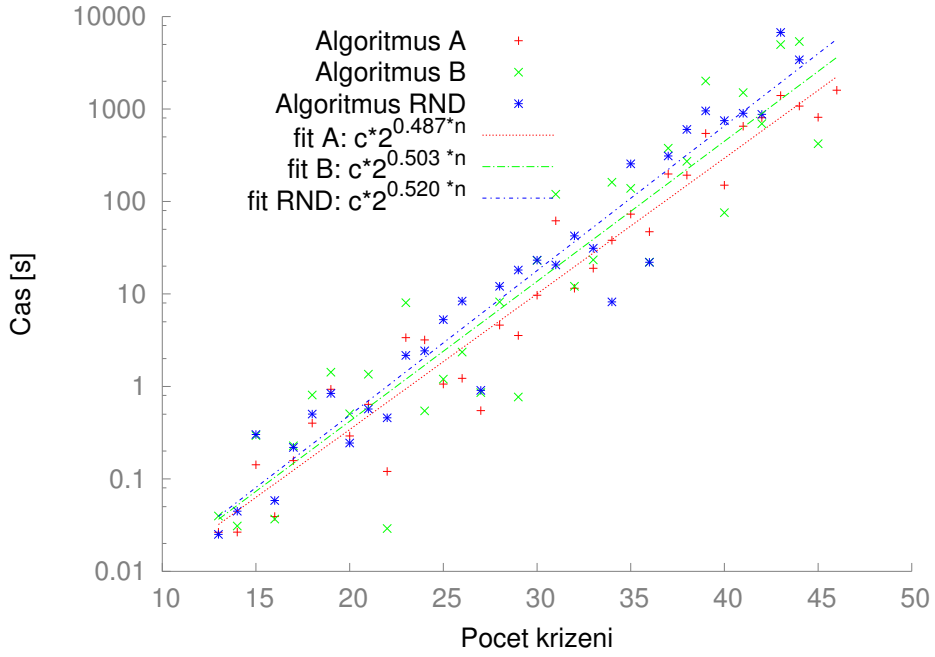
Podle výsledků není znatelný rozdíl mezi dobou výpočtu na uzlech, alternujících uzlech či lincích. Ani jednotlivé algoritmy se od sebe výrazně neliší, nejrychleji ale Jonesův polynom vždy počítá algoritmus A. Jeho průměrná časová složitost je podle naměřených dat menší než $\mathcal{O}(2^{0,5n})$.

Na uzlech a alternujících uzlech běží algoritmy B i RND podobnou dobu, liší se na ovšem na lincích. Zde se algoritmus B vyrovná algoritmu A, ale algoritmus RND zaostává. Náhodná volba křížení k rozpojení pravděpodobně způsobí rozpadnutí na větší počet stále zamotaných linků.

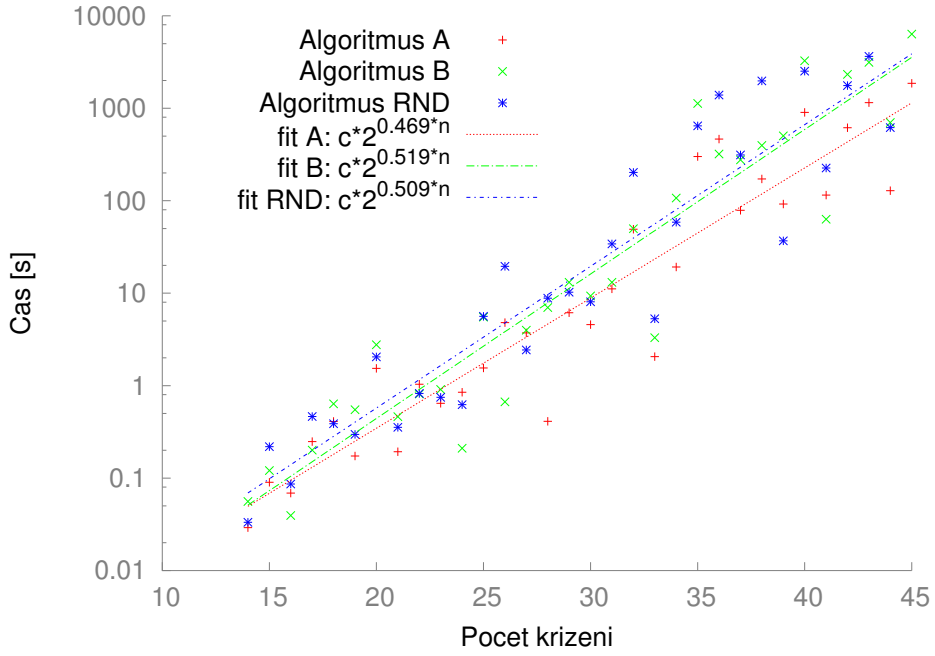
Odhad průměrné časové složitosti Chyby maximálně pět procent.

	A	**B**	**RND**
Uzel	0,487	0,503	0,520
Alternující uzel	0,469	0,519	0,509
Link	0,486	0,485	0,555

Tabulka 3.1: Odhady parametru k průměrné časové složitosti $\mathcal{O}(2^{kn})$ jednotlivých algoritmů.



Obrázek 3.4: Doby běhu algoritmů náhodných uzlech proložené křivkami, logaritmická škála.



Obrázek 3.5: Doby běhu algoritmů na náhodných alternujících uzlech proložené křivkami, logaritmická škála.

3.3 Torusové uzly

Rychlost algoritmů jsme vyzkoušeli na 36 nejmenších torusových uzlech. Jejich PD notaci jsme získali z databáze knot atlas.

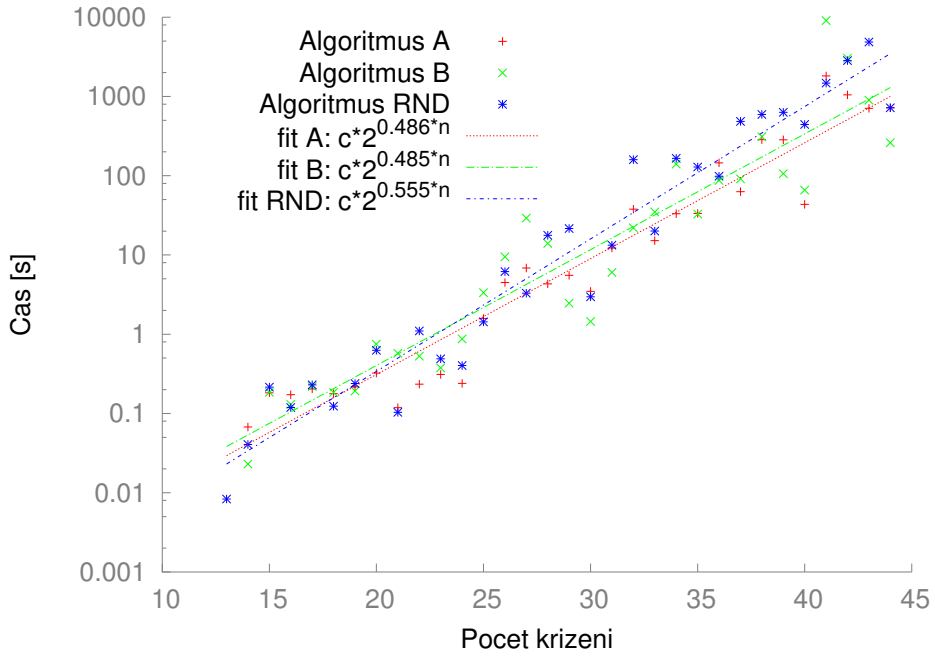
TO není torusový uzel!! Ale znamená to, že z toho dokážu vyrobit link? To by bylo fajn. Zmenším. A dám vedle toho. Pěkné to bude.

Na grafu dob běhu algoritmů lze rozlišit dva shluky bodů (viz Obrázek 3.7). Spodní shluk odpovídá torusovým uzlům $(2k+1,\, 2)$, tedy těm, které se obmotají dvakrát podél osy rotace torusu a jejich diagram má tvar dvou zakroucených vláken s $2k+1$ kříženími.

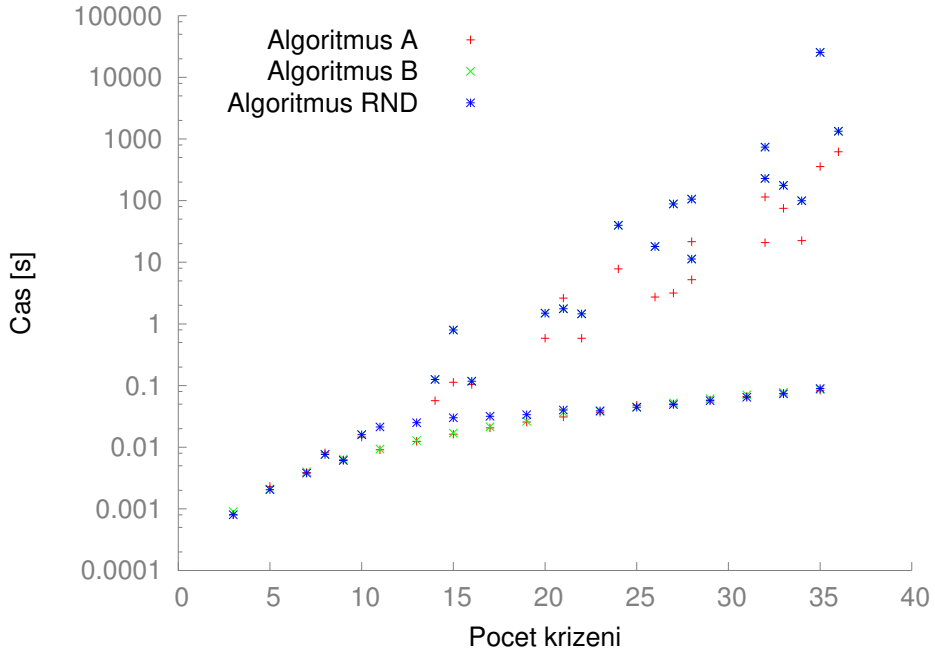
Všechny stěny tohoto uzlu jsou ohraničené pouze dvěma hranami. Rozpojením libovolného křížení jedním směrem vznikne torusový link $(2k,\, 2)$, druhým směrem triviální uzel, který lze rozmotat několika prvními Reidemeistrovými pohyby. Z linku zase rozpojením libovolného křížení vznikne uzel $(2k-1,\, 2)$. Rozpojení a rozmotávání má lineární časovou složitost, celkově tedy všechny tři algoritmy spočítají Jonesův polynom tohoto typu torusových uzlů v kvadratickém čase (viz Obrázek 3.9).

Na torusových uzlech tvaru $(p,\, q)$, kde $q \neq 2$ již doba běhu algoritmu roste exponenciálně. Nejrychlejší je algoritmus A, příslušná exponenciála je tvaru 2^{53n} (viz Obrázek 3.10).

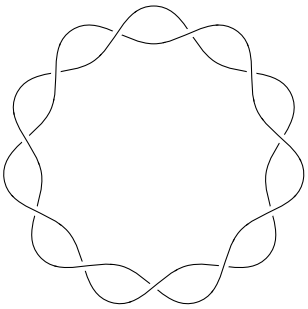
Nezapomenou dělat závěry a shrnovat.



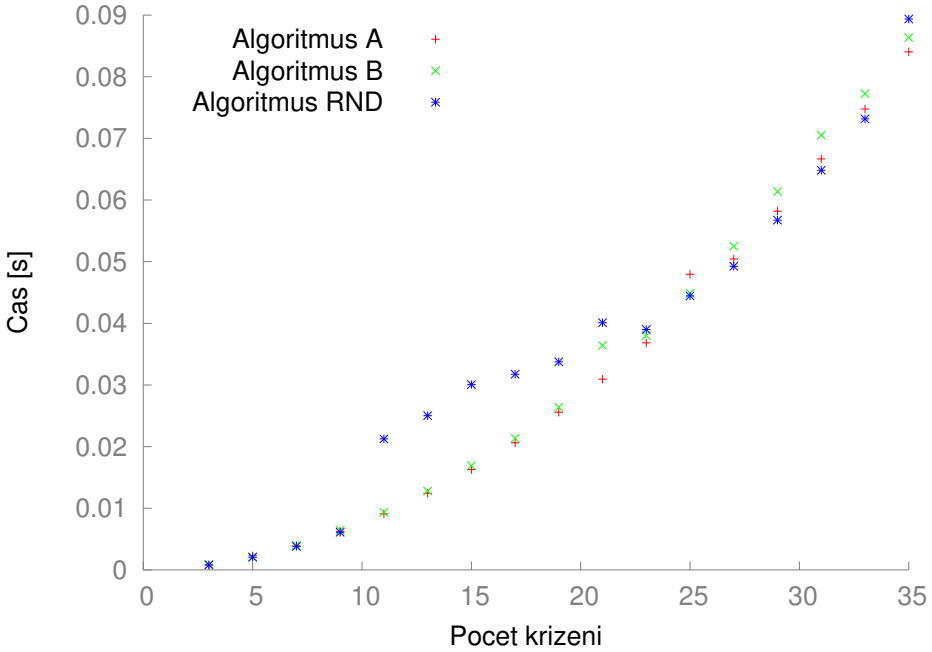
Obrázek 3.6: Doby běhu algoritmů na náhodných lincích proložené křivkami, logaritmická škála.



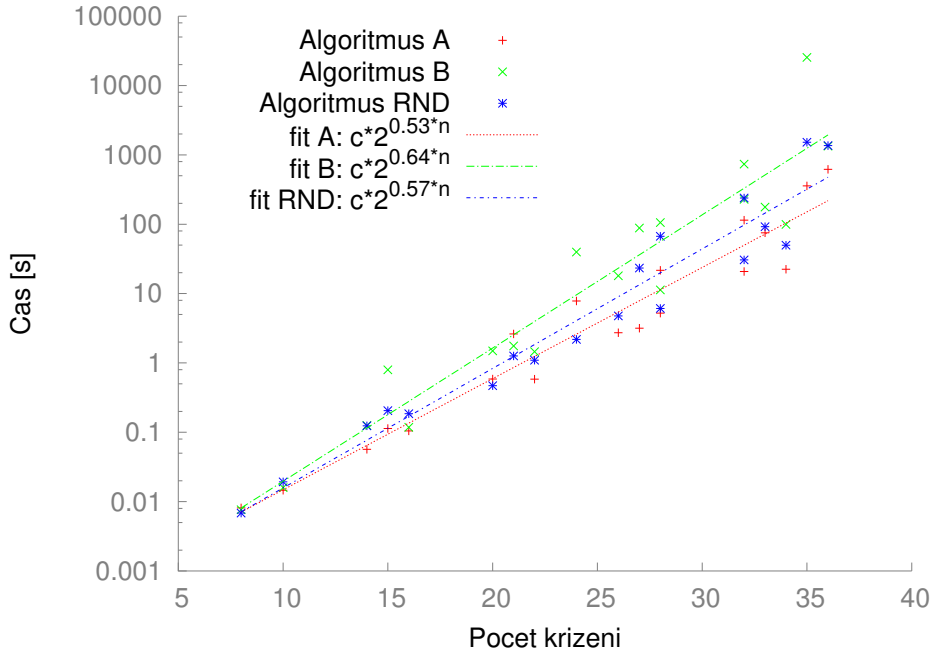
Obrázek 3.7: Doby běhu algoritmů na 36 torusových uzlech, logaritmická škála.



Obrázek 3.8: Torusový uzel (11,2).



Obrázek 3.9: Doby běhu algoritmů na torusových uzlech (3,2) až (35,2).



Obrázek 3.10: Doby běhu algoritmů na malých torusových uzlech (p, $q \neq 2$) proložené křivkami, logaritmická škála.

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?

Nedělitelné mezery

Labelování definic, obrázků, vět...

Seznam použité literatury

Aaronson, S. a kol. Complexity Zoo: Sharp-P. `https://complexityzoo.uwaterloo.ca/Complexity_Zoo:Symbols#sharpp`. [Online; navštíveno dne 10. 7. 2018].

Adams, C. (2004). *The knot book : an elementary introduction to the mathematical theory of knots*. American Mathematical Society, Providence, R.I. ISBN 978-0821836781.

Bar-Natan, D., Morrison, S. a kol. The Knot Atlas. `http://katlas.org`. [Online; navštíveno dne 2. 7. 2018].

Cha, J. C. a Livingston, C. KnotInfo: Table of Knot Invariants. `http://www.indiana.edu/~knotinfo`. [Online; navštíveno dne 2. 7. 2018].

Cromwell, P. (2004). *Knots and links*. Cambridge University Press, Cambridge, UK New York. ISBN 9780521839471.

Jaeger, F., Vertigan, D. L. a Welsh, D. J. A. (1990). On the computational complexity of the Jones and Tutte polynomials. *Mathematical Proceedings of the Cambridge Philosophical Society*, **108**(1), 35–53.

Jones, F. R. V. (2005). The Jones Polynomial. `https://math.berkeley.edu/~vfr/jones.pdf`.

Seznam obrázků

Seznam tabulek

Seznam použitých zkratek

A. Přílohy

A.1 První příloha