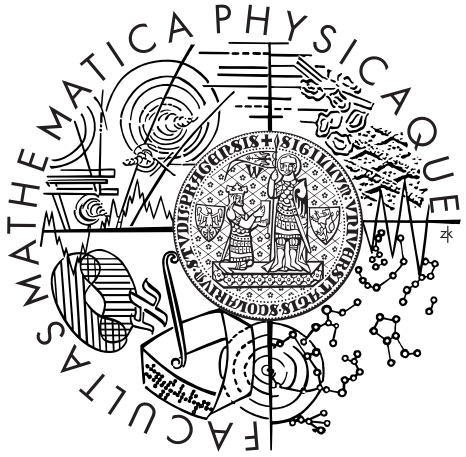
Univerzita Karlova v Praze
Matematicko-fyzikální fakulta

DIPLOMOVÁ PRÁCE



Anežka Pejlová

Generování polynomů pro číselné síto

Katedra algebry

Vedoucí diplomové práce:	prof. RNDr. Aleš Drápal, CSc., DSc.
Studijní program:	Matematika
Studijní obor:	Matematické metody informační bezpečnosti

Praha 2016

Prohlašuji, že jsem tuto diplomovou práci vypracovala 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 v Praze má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle §60 odst. 1 autorského zákona.

V Praze dne 13. května 2016 .....

Název práce: Generování polynomů pro číselné síto

Autor: Anežka Pejlová

Katedra: Katedra algebry

Vedoucí diplomové práce: prof. RNDr. Aleš Drápal, CSc., DSc., Katedra algebry

Abstrakt: V této práci se zaměřujeme zejména na Kleinjungův algoritmus pro generování polynomů v rámci obecného číselného síta, což je v současnosti nejefektivnější faktorizační algoritmus. Obecně užívané postupy jsou popsány s důrazem na vysvětlení, které části lze rigorózně dokázat a které jsou motivovány pouze heuristicky. Přínosem práce je také přiložená implementace Kleinjungova algoritmu vyvinutá v rámci projektu NFS vedeného na Katedře algebry. Empirická data získaná z této implementace podpírají vhodnost některých popsaných heuristik.

Klíčová slova: Číselné síto, Kleinjungův algoritmus

Title: Generating polynomials for number field sieve

Author: Anežka Pejlová

Department: Department of Algebra

Supervisor: prof. RNDr. Aleš Drápal, CSc., DSc., Department of Algebra

Abstract: The topic of this thesis is mainly focused on Kleinjung algorithm for generating polynomials within the General Number Field Sieve, which is the most efficient factorization algorithm nowadays. Commonly used consecutions are explained with respect to the fact whether they can be rigorously proven or they are based only on heuristic assumptions. Another contribution of this thesis is the attached implementation of Kleinjung algorithm developed as a part of the Number Field Sieve project led by the Department of Algebra. The appropriateness of some heuristics used in the theory beyond the Kleinjung algorithm is supported by empirical data obtained from this implementation.

Keywords: Number field sieve, Kleinjung algorithm

V první řadě bych chtěla velmi poděkovat svému vedoucímu diplomové práce prof. RNDr. Aleši Drápalovi, CSc., DSc. za velkou trpělivost při vedení práce, četné odborné rady a čas i energii, které mi věnoval během konzultací. Cennými radami při programování mi přispěli autoři softwarové implementace číselného síta Mgr. Robert El Bashir, Dr., Mgr. Jan Jeroným Zvánovec, Mgr. Lukáš Perůtka a RNDr. Přemysl Jedlička, Ph.D., za což jim patří velký dík. Za rady z oboru teorie čísel děkuji také Mgr. Víťovi Kalovi, Ph.D.

Poděkování za podporu a motivaci patří také mé rodině a nejbližším přátelům.

Obsah

Úvod **3**

1 Základy teorie číselného síta **4**
1.1 Číselná tělesa 4
1.2 Hladké hodnoty 5
1.3 Výpočetní složitost 6
1.4 Kořeny polynomů 12

2 Algoritmus číselného síta **14**
2.1 Princip Fermatovy faktorizace 14
2.2 Přehled algoritmu obecného číselného síta 15
2.2.1 Fáze generování polynomů 15
2.2.2 Prosívací fáze 16
2.2.3 Fáze zpracování relací 18
2.2.4 Lineární fáze 19
2.2.5 Odmocňovací fáze 19

3 Generování polynomů **21**
3.1 Base-m metoda 21
3.2 Kleinjungův algoritmus 22
3.2.1 Filtr polynomiálních kandidátů 25

4 Hodnocení polynomů **32**
4.1 Vlastnosti ovlivňující výtěžnost 33
4.1.1 Velikost koeficientů 33
4.1.2 Volba stupně nelineárního polynomu 38
4.1.3 Kořenové vlastnosti 48
4.2 Kombinované hodnocení 54
4.2.1 Výtěžnost zkosených polynomů 55
4.3 Optimalizace výtěžnosti 56

5 Optimalizace kořenových vlastností **58**
5.1 Kořenové síto 58
5.2 Optimalizace kořenového síta 60
5.3 Dvoufázové kořenové síto 63
5.3.1 První fáze – prohledávání stromu do hloubky 63
5.3.2 Druhá fáze – prosívání na mříži 64

6 Implementační aspekty generování polynomů **68**
6.1 Implementace Kleinjungova algoritmu 68
6.1.1 Výpočet base-(m,p) rozvoje 68
6.1.2 Výpočet Dickmanovy funkce 69
6.1.3 Vylepšení metody největšího spádu 71
6.2 Zjednodušení hodnocení polynomu 72

Závěr **76**

Literatura **78**

Seznam algoritmů **81**

Seznam obrázků **82**

Seznam tabulek **83**

Úvod

Kryptografie se, byť pro drtivou většinu lidí nevědomky, stala součástí každodenního života. Kupříkladu pokaždé, když na internetu otvíráme emailovou schránku, navazuje se mezi naším počítačem a vzdálenými servery šifrované spojení. V této a mnohých dalších úlohách nalezla své důležité postavení asymetrická kryptografie, která ve své nejvšednější variantě plně těží z domněnky, že *problém faktorizace* (rozkladu čísla na součin prvočísel) je těžký.

Problém faktorizace se vine v podstatě celou historií matematiky již od objevu prvočísel a postupně vznikla celá řada různých faktorizačních metod, které více či méně chytře aplikují zkusmé metody pro hledání (prvočíselného) rozkladu. Nicméně ještě v sedmdesátých letech minulého století bylo nemyslitelné rozkládat čísla s více než dvaceti decimálními ciframi. V následujících dvaceti letech šel vývoj rychle kupředu a vznikla myšlenka tzv. *číselného síta*. Nejprve jako *speciálního číselného síta* (Special Number Field Sieve) vhodného pro faktorizaci čísel tvaru $r^e \pm s$, kde r a s jsou malé. Z něj byl následně odvozen v současnosti nejefektivnější známý faktorizační algoritmus označovaný jako *obecné číselné síto* (General Number Field Sieve), jehož první část budeme studovat. Pro představu, rekordní faktorizace v dnešní době překročily hranici 230 decimálních cifer, což odpovídá 768 bitovému číslu.

V kapitole 1 zavedeme nejnutnější pojmy pro pochopení principu číselného síta a ukážeme vlastnosti, které dále využijeme. Obvykle je algoritmus číselného síta představován v řeči monických polynomů a následně jsou poznamenány změny, které je nutné učinit pro užití s polynomy nemonickými. Jelikož budeme pracovat zejména s nemonickými polynomy, formulujeme v kapitole 2 přehledově algoritmus číselného síta rovnou pro tento případ. Algoritmus číselného síta lze rozdělit do pěti fází, v dalším textu se zaměříme na fázi úvodní – generování polynomů. Naším cílem bude popsat všechny náležitosti první fáze, od samotného generování polynomů přes optimalizace jejich vlastností až po výběr nejlepšího kandidáta, který je předán do dalších fází číselného síta. V kapitole 3 popíšeme *Kleinjungův algoritmus*, který je nejefektivnější známou metodou generování polynomů pro číselné síto. Dokážeme také tvrzení, o která se princip algoritmu opírá. V navazující kapitole 4 podrobně rozebereme, jaké vlastnosti od generovaných polynomů požadujeme a jak polynomy na základě těchto vlastností hodnotíme. Výklad pojednáme se zřetelem na to, které dílčí známé a používané postupy mají rigorózní zdůvodnění a které jsou heuristické. Ve vybraných případech provedeme diskuzi vhodnosti zvolené heuristiky. V samostatné kapitole 5 se zaměříme na kořenové vlastnosti polynomů a jejich optimalizaci. Kleinjungův algoritmus včetně optimalizací a hodnocení polynomů jsme také implementovali jako součást celého algoritmu číselného síta vyvíjeného na Katedře algebry MFF UK. V poslední kapitole 6 shrneme vybrané praktické poznatky nabyté při implementaci.

Kapitola 1

Základy teorie číselného síta

1.1 Číselná tělesa

Pro účely popisu algoritmu číselného síta zavedeme nejprve několik základních pojmů z teorie číselných těles a ukážeme důležité vlastnosti.

Definice 1 (Číselné těleso)**.** *Nechť K je těleso, $K \leq \mathbb{C}$ a stupeň rozšíření $[K : \mathbb{Q}]$ je konečný. Pak K nazveme* číselným tělesem*. Navíc lze ukázat, že $K = \mathbb{Q}[\alpha]$ pro α algebraické nad $\mathbb{Q}$.*

Mějme monický ireducibilní polynom $f \in \mathbb{Q}[x]$ a α ať je nějaký komplexní kořen f. Potom $f = \mathrm{m}_{\mathbb{Q},\alpha}$ je minimální polynom prvku α nad $\mathbb{Q}[x]$. Příslušné číselné těleso je $K = \mathbb{Q}[\alpha]$, přitom $\mathbb{Q}[\alpha] \cong \mathbb{Q}[x]/(f)$, kde $h(\alpha) \leftrightarrow h(x) + (f)$ je korektně definovaný izomorfismus. Pro účely algoritmu číselného síta budeme uvažovat pevně zvolený ireducibilní polynom f (v další kapitole ukážeme, jak řešit situaci, když f není monický) a příslušné číselné těleso $K = \mathbb{Q}[\alpha]$ pro α komplexní kořen f.

Definice 2 (Hladkost)**.** *Buď $B \in \mathbb{N}$. Libovolné $n \in \mathbb{Z}$ nazveme B-*hladké*, pokud nemá prvočíselného dělitele většího než B.*

Definice 3 (Okruh algebraických celých čísel)**.** *Prvek $\beta \in K$, kde K je číselné těleso, budeme nazývat* celistvý*, pokud existuje monický polynom $g \in \mathbb{Z}[x]$ takový, že $g(\beta) = 0$.* Okruh algebraických celých čísel v číselném tělese K *je pak obor všech celistvých prvků z K a budeme ho dále značit $\mathbb{Z}_K$.*

Algebraická celá čísla tvoří okruh v číselném tělese. Lze také ukázat (např. [22, věta 2.29]), že $\mathbb{Z}_K$ je noetherovský obor integrity, kde každý nenulový prvoideál je maximální. Navíc, K je podílové těleso $\mathbb{Z}_K$ a každý prvek K, který je kořenem monického polynomu ze $\mathbb{Z}_K[x]$, už leží v $\mathbb{Z}_K$, tj. $\mathbb{Z}_K$ je celistvě uzavřené ve svém podílovém tělese. Takové obory nazýváme *Dedekindovými obory*. Využijeme dále vlastnosti, že v Dedekindových oborech existuje jednoznačný rozklad ideálů na prvoideály [22, kap. 2].

Pro α celistvý prvek K platí $\mathbb{Z}[\alpha] \subset \mathbb{Z}_K$ a můžeme zavést následující definici.

Definice 4 (Nespeciální prvočíslo)**.** *Za* nespeciální *prvočíslo označíme takové $p \in \mathbb{P}$, pro které platí $p \nmid [\mathbb{Z}_K : \mathbb{Z}[\alpha]]$. V opačném případě mluvíme o prvočíslu* speciálním*.*

Nespeciální prvočísla budou ta, jež nás budou ve většině případů zajímat. Speciální prvočísla vyžadují zvláštní zacházení, které případně nastíníme, nicméně podrobně se nespeciálními prvočísly zabývat nebudeme.

Definice 5 (Prvoideál nad prvočíslem)**.** *Řekneme, že prvoideál P je* nad *prvočíslem p, pokud $P \cap \mathbb{Z} = p\mathbb{Z}$.*

Definice 6 (p-valuace)**.** *Mějme $v \in \mathbb{Z}, v \neq 0$ a p prvočíslo, pak* p-valuace *prvku v je rovna exponentu nejvyšší mocniny p dělící v. Budeme ji dále značit jako* $\mathrm{val}_p(v)$.

Analogickým způsobem lze zavést pojem valuace u ideálů. Buď $I = \prod_i P_i^{e_i}$ rozklad ideálu I na prvoideály. Pak $\mathrm{val}_{P_i}(I) = e_i$.

Definice 7 (Norma)**.** *Buď $\mathrm{m}_{\mathbb{Q},\alpha}$ minimální polynom prvku $\alpha \in \mathbb{C}$, $\deg(\mathrm{m}_{\mathbb{Q},\alpha}) = d$ a mějme $K \cong \mathbb{Q}[x]/(\mathrm{m}_{\mathbb{Q},\alpha})$ číselné těleso. Označme $\alpha = \alpha_1, \ldots, \alpha_d$ konjugované prvky k α v $\mathbb{C}$ a zvolme libovolné $h(x) \in \mathbb{Q}[x]$ stupně nejvýše $d-1$. Pak* norma *prvku $h(\alpha) \in K$ je zobrazení*

$$\mathcal{N}\colon K \to \mathbb{Q} \quad \textit{definované} \quad \mathcal{N}(h(\alpha)) = \prod_{i=1}^{d} h(\alpha_i).$$

Lze ukázat, že norma je multiplikativní zobrazení, tedy

$$\mathcal{N}(h_1 \cdot h_2) = \mathcal{N}(h_1) \cdot \mathcal{N}(h_2).$$

Pro praktický výpočet hodnoty prvku $h(\alpha) = a - b\alpha \in \mathbb{Q}[\alpha]$ se hodí vztah

$$\mathcal{N}(a - b\alpha) = \frac{1}{a_d} |F(a,b)|, \tag{1.1}$$

kde F je homogenizace f. Vztah plyne z [22, lemma 2.11].

1.2 Hladké hodnoty

Pro hodnocení výtěžnosti polynomu budeme chtít znát nějaký odhad pro počet hladkých čísel menších než daná mez. Takový odhad poskytuje tzv. *Dickmanova funkce.*

Mějme $r, B \in \mathbb{N}$ a označme $\psi(r,B) = |\{v \leq r \mid v \text{ je } B\text{-hladké}\}|$. Odhad výpočtu $\psi(r,B)$ poskytuje následující tvrzení:

Věta 1 (Dickman [10])**.** *Pro všechna $u \geq 1$ má funkce $\frac{\psi(x, x^{\frac{1}{u}})}{x}$ vlastní nenulovou limitu v nekonečnu.*

Definice 8 (Dickmanova funkce)**.** *Limitu, jejíž existenci ukazuje věta 1, budeme dále značit $\rho(u)$ a budeme ji nazývat* Dickmanovou funkcí. *V intervalu $0 \leq u < 1$ dodefinujeme Dickmanovu funkci jako $\rho(u) = 1$.*

Pomocí Dickmanovy funkce budeme hodnotit polynomy podle jejich odhadované výtěžnosti na prosívací oblasti. Formulujme následující tvrzení:

Lemma 2. *Buď $B > 1$, pak $\rho\left(\frac{\ln r}{\ln B}\right)$ je limita pravděpodobnosti pro $r \to \infty$, že náhodné celé číslo menší nebo rovno r je B-hladké.*

Důkaz. Položme $u = \frac{\ln r}{\ln B}$, pro dostatečně velké r bude jistě $u \geq 1$. Vyjádřeme mez hladkosti $B = \mathrm{e}^{\frac{\ln r}{u}} = r^{\frac{1}{u}}$. Pak je podle Dickmanovy věty

$$\rho\left(\frac{\ln r}{\ln B}\right) = \rho(u) = \lim_{r\to\infty} \frac{\psi(r,r^{\frac{1}{u}})}{r} = \lim_{r\to\infty} \frac{\psi(r,B)}{r}.$$

□

Explicitní výpočet hodnoty Dickmanovy (někdy též označované jako Dickmanovy–de Bruijnovy) funkce je netriviální, výpočetně náročný problém. Podrobněji o problematice výpočtu Dickmanovy funkce pojednáváme v podkapitole 6.1.2.

1.3 Výpočetní složitost

Za účelem využití v odhadech výpočetní složitosti algoritmu číselného síta definujme značení podobné známé $\mathcal{O}$-notaci.

Definice 9 (L-notace)**.** *Pro konstanty $0 \leq \alpha \leq 1$ a $\beta > 0$ definujeme značení*

$$L_n[\alpha, \beta] = \exp\left[\beta(\ln n)^{\alpha}(\ln\ln n)^{1-\alpha}\right]$$

pro $n \to \infty$.

L-notace se často používá asymptoticky ve smyslu $L_n[\alpha, \beta + o(1)]$, kde $o(1)$ zastupuje konkrétní funkci závisející na n a jdoucí k nule. Říká se též, že $o(1)$ je uniformní v nějaké další proměnné t. Znamená to, že konkrétní funkci reprezentující $o(1)$ a závisející na n a t lze majorizovat funkcí závisející pouze na n a jdoucí k nule.

Poznámka. Asymptotická L-notace zjednodušuje práci se subexponenciální (a superpolynomiální) složitostí, protože pokud při volbě $\beta = b + o(1)$ pro $n \to \infty$ volíme $\alpha = 0$, jest

$$L_n[0, b + o(1)] = \exp[(b + o(1)) \ln\ln n] = (\ln n)^{b+o(1)},$$

což je polynomiální funkce v $\ln n$. Naopak pro $\alpha = 1$ dostáváme

$$L_n[1, b + o(1)] = \exp[(b + o(1)) \ln n] = n^{b+o(1)}$$

exponenciální funkci v $\ln n$.

Pro důkazy tvrzení v kapitole 4.1.2 využijeme dvě tvrzení převzatá z článku Buhlera, Lenstry a Pomerance [6].

Věta 3 ([6, věta 10.1.])**.** *Mějme funkci $h(y)$ definovanou pro $y \geq 2$, která splňuje $h(y) \geq 1$ a $h(y) = y^{1+o(1)}$ pro $y \to \infty$. Pak platí*

$$\frac{xh(y)}{\psi(x,y)} \geq L_x\left[\frac{1}{2}, \sqrt{2} + o(1)\right],$$

když $x \to \infty$ uniformně pro $y \geq 2$.

Důkaz této věty v článku [6] vynechává některé důležité detaily, které čtenáři nemusí být na první pohled zřejmé. Proto větu nyní dokážeme, včetně podrobnějšího vysvětlení. Celý odhad vychází z následujícího vztahu:

Věta 4 (Canfield, Erdös, Pomerance [7])**.** *Pro všechna $\epsilon > 0$ existuje reálná funkce $\delta_\epsilon(x)$ jdoucí k nule pro x rostoucí nade všechny meze taková, že když*

$$3 \leq u \leq \frac{(1-\epsilon)\ln x}{\ln\ln x}, \tag{1.2}$$

pak

$$\psi(x, x^{\frac{1}{u}}) = x \cdot \exp\left[-u \ln u(1 + o(1))\right], \tag{1.3}$$

kde $o(1) = \tau(x,u)$ takové, že

$$|\tau(x,u)| \leq |\delta_\epsilon(x)|\,.$$

Poznámka. Pro dolní odhad $\psi(x, x^{\frac{1}{u}})$ stačí $u \geq 3$, nerovnost $u \leq \frac{(1-\epsilon)\ln(x)}{\ln\ln x}$ je potřeba pro horní odhad, který vychází z klasického výsledku de Bruijna [5].

Funkce $\frac{\ln(x)}{\ln\ln x}$ je rostoucí pro $x \geq \mathrm{e}^{\mathrm{e}} \doteq 15{,}15$. Proto pokud $u \leq \frac{(1-\epsilon)\ln(x)}{\ln\ln x}$, pak pro všechna $x' > x$ je rovněž $u \leq \frac{(1-\epsilon)\ln(x')}{\ln\ln x'}$.

Tvrzení 5. *Ať $\epsilon > 0$. Pak existuje hodnota u_ϵ, že pro $u > u_\epsilon$ a $x \geq u^{u(1+\epsilon)}$ je*

$$\psi\left(x, x^{\frac{1}{u}}\right) = \frac{x}{u^{u(1+o(1))}},$$

kde $o(1) = \tau(x,u)$ je uniformně omezené $\sigma_\epsilon(u)$, tj.

$$|\tau(x,u)| \leq \sigma_\epsilon(u) \text{ a } \lim_{u\to\infty} \sigma_\epsilon(u) = 0.$$

Důkaz. Chceme zajistit, aby pro u dostatečně velké platil předpoklad (1.2)

$$u \leq (1-\delta)\frac{\ln x}{\ln\ln x}.$$

Přímočará volba $\delta = \epsilon$ nevede k cíli, poněvadž pro $x = u^{u(1+\epsilon)}$ bychom dostali

$$\begin{aligned} u &\leq (1-\epsilon)\frac{u(1+\epsilon)\ln u}{\ln(u(1+\epsilon)\ln u)} \\ u\ln\left(u(1+\epsilon)\ln u\right) &\leq u(1-\epsilon^2)\ln u \\ \ln u + \ln(1+\epsilon) + \ln\ln u &\leq (1-\epsilon^2)\ln u \\ \ln(1+\epsilon) + \ln\ln u &\leq -\epsilon^2 \ln u. \end{aligned}$$

Což neplatí, protože levá strana je kladná.

Napíšeme $1+\epsilon$ jako $\eta + \delta$, kde $\eta > 1$ a $\delta > 0$, a budeme zkoumat stejnou nerovnost pro $x = u^{u(\eta+\delta)}$

$$\begin{aligned} u &\leq (1-\delta)\frac{u(\eta+\delta)\ln u}{\ln(u(\eta+\delta)\ln u)} \\ u\ln\left(u(\eta+\delta)\ln u\right) &\leq u(1-\delta)(\eta+\delta)\ln u \\ \ln u + \ln(1+\epsilon) + \ln\ln u &\leq (1+\epsilon-\delta-\delta\epsilon)\ln u \\ \ln\ln u + \ln(1+\epsilon) &\leq (\epsilon - \delta(1+\epsilon))\ln u. \end{aligned}$$

Například volba $\delta = \frac{\epsilon}{2}$ vede na

$$\ln\ln u + \ln(1+\epsilon) \leq \left(\frac{\epsilon}{2} - \frac{\epsilon^2}{2}\right)\ln(u),$$

což je pro u dostatečně velké zjevně splněno. □

S využitím těchto vlastností funkce $\psi(x,y)$ ukážeme nyní několik dalších vztahů, se kterými budeme pracovat v důkazu věty 3. Jak jsme využili již v důkazu lemmatu 2, platí

$$\psi(x,y) = \psi\left(x, x^{\frac{1}{u}}\right) \qquad \text{pro} \qquad u = \frac{\ln x}{\ln y}.$$

Tvrzení 6. *Při zavedeném značení platí*

$$\psi\left(x, L_x\left[\frac{1}{2}, \beta\right]\right) = \frac{x}{L_x\left[\frac{1}{2}, \frac{1}{2\beta} + o(1)\right]}, \tag{1.4}$$

kde $\beta > 0$ a $o(1)$ značí funkci $\tau_\beta(x) \to 0$, která závisí na β.

Navíc, je-li $0 < B \leq \beta \leq C$, pak existuje funkce $T(x)$, $\lim_{x\to\infty} T(x) = 0$, která majorizuje $\tau_\beta(x)$ a je na β nezávislá.

Důkaz. K aplikaci vztahu (1.3) potřebujeme znát

$$u = \frac{\ln x}{\ln L_x\left[\frac{1}{2}, \beta\right]} = \frac{\sqrt{\ln x}}{\beta\sqrt{\ln\ln x}} = \beta^{-1}\sqrt{\frac{\ln x}{\ln\ln x}}$$

a chceme, aby platil předpoklad (1.2), tedy pro libovolné $\epsilon > 0$

$$\begin{aligned} u &\leq \frac{(1-\epsilon)\ln(x)}{\ln\ln x} \\ \beta^{-1}\sqrt{\frac{\ln x}{\ln\ln x}} &\leq \frac{(1-\epsilon)\ln(x)}{\ln\ln x} \\ \beta^{-1} &\leq (1-\epsilon)\sqrt{\frac{\ln x}{\ln\ln x}}, \end{aligned}$$

když x je velké. To ale od nějaké meze pro x platit musí, protože $\sqrt{\frac{\ln x}{\ln\ln x}}$ je funkce rostoucí.

Můžeme proto aplikovat vztah (1.3) a máme

$$\psi\left(x, x^{\frac{1}{u}}\right) = \psi\left(x, L_x\left[\frac{1}{2}, \beta\right]\right) = \frac{x}{u^{u(1+g_\beta(x))}},$$

kde $g_\beta(x)$ je uniformně omezená nějakou $\sigma_\beta(u)$ dle tvrzení 5. Jde-li $x \to \infty$, jde $u \to \infty$, takže $|g_\beta| \to 0$.

Pokusme se nyní zapsat výraz $u^{u(1+g_\beta(x))} = \exp\left[u\ln u(1+g_\beta(x))\right]$ v L-notaci. Pro přehlednost pišme $\ln_3 x = \ln\ln\ln x$. Víme

$$\begin{aligned} u &= \beta^{-1}\sqrt{\frac{\ln x}{\ln\ln x}} \\ \ln u &= -\ln\beta + \frac{1}{2}(\ln\ln x - \ln_3 x), \end{aligned}$$

tudíž

$$u\ln u = -\frac{\ln\beta}{\beta}\sqrt{\frac{\ln x}{\ln\ln x}} + \frac{\sqrt{\ln x\ln\ln x}}{2\beta} - \frac{\ln_3 x}{2\beta}\sqrt{\frac{\ln x}{\ln\ln x}} =$$
$$= \frac{\sqrt{\ln x\ln\ln x}}{2\beta}\left(\frac{2\ln\beta}{\ln\ln x} + 1 - \frac{\ln_3 x}{\ln\ln x}\right).$$

Označme $h_\beta(x) = \frac{2\ln\beta+\ln_3 x}{\ln\ln x}$, zřejmě platí $\lim_{x\to\infty} h_\beta(x) = 0$. Je pak

$$\exp\left[u\ln u(1+g_\beta(x))\right] = \exp\left[\frac{1}{2\beta}(1-h_\beta(x))(1+g_\beta(x))\sqrt{\ln x\ln\ln x}\right] =$$
$$= \exp\left[\left(\frac{1+g_\beta(x)-h_\beta(x)-g_\beta(x)h_\beta(x)}{2\beta}\right)\sqrt{\ln x\ln\ln x}\right] =$$
$$= \exp\left[\left(\frac{1}{2\beta}+\tau_\beta(x)\right)\sqrt{\ln x\ln\ln x}\right],$$

kde $\tau_\beta(x)\to 0$ pro $x\to\infty$. Poslední výraz je možno zapsat jako

$$L_x\left[\frac{1}{2},\frac{1}{2\beta}+\tau_\beta(x)\right] = L_x\left[\frac{1}{2},\frac{1}{2\beta}+o(1)\right].$$

Dokázali jsme vztah (1.4) pro $o(1)$ závislé na β libovolném. Buď nyní $0<B\le\beta\le C$ a hledáme majorizaci

$$\tau_\beta(x) = \frac{g_\beta(x)-h_\beta(x)-g_\beta(x)h_\beta(x)}{2\beta}$$

nezávislou na β. Nejdříve jest

$$h_\beta(x) \le \frac{2\ln C+\ln_3 x}{\ln\ln x},$$

kde pravá strana nerovnosti jde jistě do nuly pro $x\to\infty$. O $g_\beta(x)$ víme, že lze majorizovat nějakou $\rho_\beta(u)\to 0$ pro $u\to\infty$, tedy

$$|g_\beta(x)| \le \rho_\beta\left(\frac{1}{\beta}\sqrt{\frac{\ln x}{\ln\ln x}}\right) \le \rho\left(\frac{1}{B}\sqrt{\frac{\ln x}{\ln\ln x}}\right),$$

kde zřejmě $\lim_{x\to\infty}\rho\left(\frac{1}{B}\sqrt{\frac{\ln x}{\ln\ln x}}\right)=0$. Hledaná funkce $T(x)\to 0$ nezávislá na β tudíž existuje. □

Tvrzení 7. *Ať $0<B\le\theta\le C$ a mějme $h(y)\ge 1$ a $h(y)=y^{1+o(1)}$. Pak existují funkce $\gamma_\theta(x)$ a $\Gamma(x)$, pro kterou $\lim_{x\to\infty}\Gamma(x)=0$, že $|\gamma_\theta(x)|\le|\Gamma(x)|$ a*

$$\frac{xh\left(L_x\left[\frac{1}{2},\theta\right]\right)}{\psi\left(x,L_x\left[\frac{1}{2},\theta\right]\right)} = L_x\left[\frac{1}{2},\theta+\frac{1}{2\theta}+\gamma_\theta(x)\right].$$

(Jinými slovy, $\frac{xh\left(L_x\left[\frac{1}{2},\theta\right]\right)}{\psi\left(x,L_x\left[\frac{1}{2},\theta\right]\right)} = L_x\left[\frac{1}{2},\theta+\frac{1}{2\theta}+o(1)\right]$, kde $o(1)$ je pro $x\to\infty$ a je uniformní pro θ.)

Důkaz. Máme $h(y) = y^{1+o(1)}$, tedy existuje funkce $\rho(y)$, pro kterou $\lim_{y\to\infty} \rho(y) = 0$ a $h(y) = y^{1+\rho(y)}$. Dosaďme za $y = L_x[\theta]$, pak

$$h(y) = \exp\left[\theta\sqrt{\ln x \ln\ln x}\left(1 + \rho\left(\mathrm{e}^{\theta\sqrt{\ln x \ln\ln x}}\right)\right)\right]. \tag{1.5}$$

Z předpokladu platí

$$B\sqrt{\ln x \ln\ln x} \le \theta\sqrt{\ln x \ln\ln x} \le C\sqrt{\ln x \ln\ln x},$$

a protože obě meze konvergují do nekonečna, je $\rho_\theta(x) = \rho\left(\mathrm{e}^{\theta\sqrt{\ln x \ln\ln x}}\right)$ možno omezit nějakou $R(x)$ nezávislou na θ, pro kterou $\lim_{x\to\infty} R(x) = 0$.

Vynásobením vztahů (1.4) a (1.5) dostáváme

$$\begin{aligned}\frac{xh\left(L_x\left[\frac12,\theta\right]\right)}{\psi\left(x, L_x\left[\frac12,\theta\right]\right)} &= L_x\left[\frac12, \frac{1}{2\theta} + \tau_\theta(x)\right]\cdot h\left(L_x\left[\frac12,\theta\right]\right) = \\ &= \exp\left[\left(\frac{1}{2\theta} + \tau_\theta(x)\right)\sqrt{\ln x \ln\ln x}\right]\exp\left[(\theta + \theta\rho_\theta(x))\sqrt{\ln x \ln\ln x}\right] \\ &= \exp\left[\left(\frac{1}{2\theta} + \tau_\theta(x) + \theta + \theta\rho_\theta(x)\right)\sqrt{\ln x \ln\ln x}\right].\end{aligned}$$

Ovšem $|\gamma_\theta(x)| = |\tau_\theta(x) + \theta\rho_\theta(x)| \le |T(x)| + C\,|R(x)|$, takže skutečně existuje uniformní majorizace. □

Nyní již disponujeme všemi potřebnými vztahy, abychom mohli dokázat původní větu.

Důkaz. [věta 3] Je zřejmé, že pro $x \ge \mathrm{e}^{\mathrm{e}}$ je

$$L_x\left[\frac12,\gamma\right] > L_x\left[\frac12,\beta\right] \Leftrightarrow \gamma > \beta.$$

Asymptoticky totéž platí i pro $L_x\left[\frac12,\gamma+o(1)\right]$ a $L_x\left[\frac12,\beta+o(1)\right]$. Hledáme-li, jak se asymptoticky minimalizuje $L_x\left[\frac12,\theta+\frac{1}{2\theta}+o(1)\right]$, kde $o(1)$ značí majorizaci nezávislou na θ, tak hledáme $\min\left(\theta+\frac{1}{2\theta}\right)$. Pro $\theta > 0$ je derivace

$$\left(\theta + \frac{1}{2\theta}\right)' = 1 - \frac{1}{2\theta^2} = 0 \Leftrightarrow \theta = \sqrt{\frac12} = \frac{2}{\sqrt2},$$

přičemž

$$L_x\left[\frac12, \theta + \frac{1}{2\theta} + o(1)\right]\left(\sqrt{\frac12}\right) = \sqrt{\frac12} + \frac{1}{2\sqrt{\frac12}} = \frac{1}{\sqrt2} + \frac{1}{\sqrt2} = \frac{2}{\sqrt2} = \sqrt2.$$

Chceme-li však využít tvrzení 7, musíme stanovit meze B a C a ukázat, že pro $y \notin \left[L_x\left[\frac12,B\right], L_x\left[\frac12,C\right]\right]$ je $\frac{xh(y)}{\psi(x,y)}$ asymptoticky velké. Volby B a C mohou být nejrůznější, avšak minimum v bodě $\sqrt{\frac12}$ musí ležet uvnitř intervalu.

Zvolme $B = \frac14$, jelikož $h(y) \ge 1$, dostáváme pro $y \le L_x\left[\frac12,\frac14\right]$

$$\frac{xh(y)}{\psi(x,y)} \ge \frac{x}{\psi(x,y)} \ge \frac{x}{\psi\left(x, L_x\left[\frac12,\frac14\right]\right)} = L_x\left[\frac12, 2 + \sigma_\theta(x)\right],$$

kde $\lim_{x\to\infty} \sigma_\theta(x) = 0$.

Důkaz pro horní mez využívá faktu, že vždy $x \geq \psi(x,y)$, proto

$$\frac{xh(y)}{\psi(x,y)} \geq h(y),$$

nicméně je třeba zohlednit asymptotiku $h(y)$. Zvolme $C = 2$ a ukažme, že bude pro θ dostatečně velké asymptoticky v x platit $h\left(L_x\left[\frac{1}{2},\theta\right]\right) > L_x\left[\frac{1}{2},2\right]$. Pomocí vztahu (1.5) lze upravit

$$\begin{aligned} h\left(L_x\left[\frac{1}{2},\theta\right]\right) &\geq L_x\left[\frac{1}{2},2\right] \\ \exp\left[\left(\sqrt{\ln x \ln\ln x}\right)\theta\left(1+\rho\left(L_x\left[\frac{1}{2},\theta\right]\right)\right)\right] &\geq \exp\left[2\sqrt{\ln x \ln\ln x}\right] \\ \theta\left(1+\rho\left(L_x\left[\frac{1}{2},\theta\right]\right)\right) &\geq 2. \end{aligned} \tag{1.6}$$

Funkce $\rho\left(L_x\left[\frac{1}{2},\theta\right]\right) \to 0$, tedy pro velké hodnoty $L_x\left[\frac{1}{2},\theta\right]$ je v intervalu $[-\frac{1}{2},\frac{1}{2}]$. Pro $x \to \infty$ lze jistě θ stanovit tak, aby $\theta\sqrt{\ln x \ln\ln x}$ bylo od nějaké meze dostatečně velké, a platnost nerovnosti (1.6) byla nutně zajištěna.

Pro $y \in \left[L_x\left[\frac{1}{2},\frac{1}{4}\right], L_x\left[\frac{1}{2},2\right]\right]$ plyne nerovnost $\frac{xh(y)}{\psi(x,y)} \geq L_x\left[\frac{1}{2},\sqrt{2}+o(1)\right]$ přímo z tvrzení 7. Navíc, je-li $y = L_x\left[\frac{1}{2},\theta\right]$, tak existuje funkce $\omega(x) \to 0$, že

$$\frac{xh\left(L_x\left[\frac{1}{2},\theta\right]\right)}{\psi\left(x, L_x\left[\frac{1}{2},\theta\right]\right)} = L_x\left[\frac{1}{2},\sqrt{2}+\omega(x)\right] \Leftrightarrow \theta = \frac{\sqrt{2}}{2}.$$

□

Druhé tvrzení využívané v kapitole 4.1.2 je spíše technického charakteru.

Lemma 8 ([6, lemma 10.9.])**.** *Mějme $k,l \in \mathbb{R}$ taková, že $k \geq \mathrm{e}, l \geq 1$ a definujme $v = v(k,l)$ pomocí vztahu $\frac{v^2}{\ln v} = kv + l$ pro $v \geq \mathrm{e}$. Pak platí*

$$2v = (1+o(1))\left(k\ln k + \sqrt{(k\ln k)^2 + 2l\ln l}\right)$$

pro $k + l \to \infty$.

Poznámka. Ukažme, že v je pomocí vztahu $\frac{v^2}{\ln v} = kv + l$ definováno jednoznačně. Pro $k \geq \mathrm{e}$ a $l \geq 1$ označme $h(v) = v^2 - \ln v \cdot (kv + l)$ funkci definovanou na intervalu $[\mathrm{e},\infty)$. To je jistě spojitá funkce na tomto intervalu a platí $h(\mathrm{e}) = \mathrm{e}^2 - k\mathrm{e} - l$, což je pro $k \geq \mathrm{e}$ a $l \geq 1$ záporná hodnota, a $\lim_{v\to\infty} h(v) = \infty$. Funkce $h(v)$ bude mít tudíž alespoň jeden kořen. První derivace

$$h'(v) = 2v - k - \frac{l}{v} - k\ln v$$

je také v počátečním bodě záporná, $h'(\mathrm{e}) = 2\mathrm{e} - 2k - \frac{l}{\mathrm{e}} < 0$, a $\lim_{v\to\infty} h'(v) = \infty$. Aby $h(v)$ mohla mít více než jeden kořen, musely by existovat alespoň tři lokální extrémy na intervalu (e,∞). Musela by proto existovat alespoň tři řešení

$$\begin{aligned} h'(v) &= 0 \\ 2v^2 - kv - l &= kv\ln v. \end{aligned}$$

Jenže $2v^2 - kv - l$ je konvexní funkce a $kv\ln v$ je rostoucí pro $k, v \geq \mathrm{e}$. Nemohou se tedy protínat ve více než dvou bodech. Funkce $h(v)$ má proto právě jeden kořen.

1.4 Kořeny polynomů

V rámci hodnocení polynomů v kapitole 4 budeme pracovat s kořeny polynomů modulo malá prvočísla. Připomeňme nejdříve následující označení, pro r_k kořen polynomu $f(x)$ modulo p^k řekneme, že r_k je *násobný* kořen $f(x) \bmod p^k$, pokud $f'(x) \equiv 0 \bmod p$. V opačném případě hovoříme o kořenu *jednoduchém*. Henselovo lemma ukazuje podmínky, za kterých lze kořen polynomu $f(x) \bmod p$ zdvihnout na kořen polynomu $f(x) \bmod p^k$ pro $k \in \mathbb{N}$. Existuje celá řada verzí Henselova lemmatu, původní je v článku K. Hensela [13], my jej však v plné obecnosti potřebovat nebudeme. Proto ho nyní formulujeme s náležitostmi, které využijeme. Shledáváme jako vhodné lemma v uvedeném znění také dokázat.

Lemma 9 (Henselovo)**.** *Mějme polynom $f(x) \in \mathbb{Z}[x]$ a p prvočíslo, pak platí:*

(1) Je-li r_1 jednoduchý kořen $f(x) \bmod p$, pak existuje jednoznačné r_k, pro které platí $r_k \equiv r_1 \bmod p$ a zároveň $f(r_k) \equiv 0 \bmod p^k$ pro libovolné celé $k > 0$.

(2) Je-li r_k násobný kořen $f(x) \bmod p^k$ pro libovolné celé $k > 0$, pak nastává právě jedna z následujících možností:

a) r_k je také kořenem $f(x) \bmod p^{k+1}$, pak i $r_k + l \cdot p^k$ je kořenem $f(x) \bmod p^{k+1}$ pro všechna $l \in \mathbb{Z}_p$ nebo

b) r_k není kořenem $f(x) \bmod p^{k+1}$, pak r_k nelze zdvihnout na kořen modulo p^{k+1}.

Důkaz.

(1) Důkaz provedeme indukcí podle úrovně zdvihnutí. Pro $k = 1$ je tvrzení zřejmé. Předpokládejme tudíž, že tvrzení platí pro $k = n$ a chceme ukázat jednoznačnou existenci kořene r_{n+1}. Pro zachování kongruence zdvihnutého kořene s předchozím musí být ve tvaru $r_{n+1} = r_n + lp^n$ pro nějaké $l \in \mathbb{Z}_p$. Ukážeme, že takové l existuje právě jedno. Označíme-li $f(x) = \sum_{i=0}^{d} a_i x^i$, pak nás zajímá, zda

$$f(r_{n+1}) = f(r_n + lp^n) = \sum_{i=0}^{d} a_i (r_n + lp^n)^i \equiv 0 \bmod p^{n+1}.$$

S využitím binomické věty lze provést následující úpravy:

$$\begin{aligned}
f(r_{n+1}) &= \sum_{i=0}^{d} a_i (r_n + lp^n)^i = \\
&= a_0 + \sum_{i=1}^{d} a_i \sum_{j=0}^{i} \binom{i}{j} r_n^j (lp^n)^{i-j} = \\
&= a_0 + \sum_{i=1}^{d} a_i \left(r_n^i + i r_n^{i-1} lp^n + \ldots + i r_n (lp^n)^{i-1} + (lp^n)^i \right) = \\
&= a_0 + \sum_{i=1}^{d} a_i r_n^i + \sum_{i=1}^{d} a_i i r_n^{i-1} lp^n + (lp^n)^2 \left(\binom{i}{2} r_n^{i-2} + \ldots + (lp^n)^{i-2} \right) = \\
&= f(r_n) + lp^n f'(r_n) + (lp^n)^2 \left(\binom{i}{2} r_n^{i-2} + \ldots + (lp^n)^{i-2} \right) \equiv \\
&\equiv f(r_n) + lp^n f'(r_n) \bmod p^{n+1}. \qquad (1.7)
\end{aligned}$$

Z vlastností kongruencí plyne, že $f'(r_n)lp^n \equiv x \bmod p^{n+1}$ právě tehdy, když $f'(r_n) \cdot l \equiv x \bmod p$. Z indukčního kroku máme také vztahy $r_n \equiv r_1 \bmod p$ a $f(r_n) \equiv 0 \bmod p^n$. Proto lze celou kongruenci modulo p^{n+1} přepsat na ekvivalentní kongruenci modulo p

$$l \equiv -\frac{f(r_n)}{p^n} \cdot (f'(r1))^{-1} \bmod p.$$

Dělení p^n je celočíselné a inverz derivace $f'(r_1)$ v $\mathbb{Z}_p$ existuje, protože dle předpokladu je derivace nenulová modulo p. Hodnota l je tedy určena jednoznačně, proto i kořen $r_{n+1} = r_n + lp^n$ je určen jednoznačně, jelikož r_n je jednoznačný kořen modulo p^n z indukčního kroku.

(2) Pro případ násobného kořene r_k také nejprve využijeme konstrukci analogickou (1.7) z předchozího případu a získáme

$$f(r_{k+1}) \equiv f(r_k + lp^k) \equiv f(r_k) + lp^k f'(r_k) \bmod p^{k+1}.$$

Analogickým argumentem také odůvodníme, že smíme uvažovat kongruenci $l \cdot f'(r_k) \equiv l \cdot f'(r_1) \bmod p$, ale z předpokladu je $f'(r_1) \equiv 0 \bmod p$. Dostáváme tudíž kongruenci

$$f(r_k + lp^k) \equiv f(r_k) \bmod p^{k+1},$$

z které již oba případy a) i b) přímo plynou.

□

Spíše než s konkrétními hodnotami kořenů a jejich zdvihnutími budeme dále pracovat pouze s informací o korespondenci kořenů modulo prvočíselné mocniny.

Kapitola 2

Algoritmus číselného síta

2.1 Princip Fermatovy faktorizace

Základní myšlenkou, společnou všem faktorizačním algoritmům založeným na Fermatově faktorizaci, je známý fakt, že každé liché číslo N lze zapsat jako rozdíl dvou druhých mocnin nějakých přirozených čísel, a to (v případě složeného N) i několika způsoby. Máme tedy

$$N = x^2 - y^2 = (x - y)(x + y),$$

z čehož plyne, že hledání rozkladu N lze převést na hledání dvou druhých mocnin přirozených čísel, jejichž rozdíl je N. Platným řešením však zjevně může být i triviální rozklad

$$N = 1 \cdot N = \left(\frac{N+1}{2} - \frac{N-1}{2}\right)\left(\frac{N+1}{2} + \frac{N-1}{2}\right) = \left(\frac{N+1}{2}\right)^2 - \left(\frac{N-1}{2}\right)^2.$$

Cílem faktorizačních algoritmů je najít co nejefektivnější metodu, která by hledala rozklad netriviální. Moderní faktorizační algoritmy navíc podmínku rovnosti s rozkládaným N uvolňují na kongruenci

$$x^2 \equiv y^2 \pmod{N}$$

a s každou nalezenou dvojicí $(x, y) \in \mathbb{Z}_N^2$ se pokusí faktorizovat N pomocí $\mathrm{NSD}(N, x \pm y)$. Přístupem k hledání příslušných dvojic (x, y) se faktorizační algoritmy liší. Popišme si nyní ve stručnosti, jak řešení tohoto problému probíhá v obecném číselném sítu.

Poznámka. Následující podkapitola si neklade za cíl podat podrobný rigorózní popis algoritmu číselného síta. Zaměříme se přímo na případ nemonických polynomů, který nás bude zajímat, a uvedeme pouze přehledovou kostru nutnou pro pochopení myšlenek v dalších kapitolách. Popis algoritmu obecného číselného síta pro monické polynomy včetně potřebné teorie je již obsažen v pracích [22, 24] obhájených na MFF UK. Podrobným rozborem vývoje číselného síta se zabývají například Lenstra a kol. v textech [18] nebo Pomerance a Crandall v knize [9, kap. 6].

2.2 Přehled algoritmu obecného číselného síta

Základní myšlenka číselného síta posunuje princip Fermatovy faktorizace z oboru celých čísel do nějakého rozšíření celých čísel $\mathbb{Z}[\alpha]$ (podrobněji toto rozšíření popíšeme níže). Předpokládejme, že máme nějaké $\theta \in \mathbb{Z}[\alpha]$ a ϕ homomorfismus z tohoto rozšíření do $\mathbb{Z}_N$. Podaří-li se najít k prvků rozšíření tak, aby $\theta_1 \cdot \ldots \cdot \theta_k = \gamma^2$ pro nějaké $\gamma \in \mathbb{Z}[\alpha]$ a zároveň $\phi(\theta_1) \cdot \ldots \cdot \phi(\theta_k) \equiv v^2 \bmod N$ pro nějaké $v \in \mathbb{Z}$, pak, pokud označíme $u = \phi(\gamma) \in \mathbb{Z}_N$, dostáváme z vlastností homomorfismu

$$u^2 \equiv \phi(\gamma)^2 \equiv \phi(\gamma^2) \equiv \phi(\theta_1 \cdot \ldots \cdot \theta_k) \equiv \phi(\theta_1) \cdot \ldots \cdot \phi(\theta_k) \equiv v^2 \bmod N.$$

Hledaná celočíselná kongruence je $u^2 \equiv v^2 \bmod N$ a můžeme zkusit faktorizovat N pomocí $\mathrm{NSD}(N, u \pm v)$.

Pro přehlednost se algoritmus obecného číselného síta dělí do pěti fází, které podrobně řeší jednotlivé kroky předchozího odstavce:

1. fáze generování polynomů,
2. prosívací fáze,
3. fáze zpracování relací,
4. lineární fáze,
5. odmocňovací fáze.

2.2.1 Fáze generování polynomů

V celém textu uvažujeme faktorizované N liché, složené, které není mocninou. Dále v textu budeme na nejednom místě předpokládat nesoudělnost s N. Zdůrazněme, že se jedná o zcela legitimní požadavek, jelikož nalezení hodnoty soudělné s N implikuje ukončení algoritmu a faktorizaci N.

Poznámka. Neuvedeme-li jinak, budeme pracovat s N o 100 a více decimálních cifrách. Pro menší N je dle experimentálních výsledků efektivnější použít tzv. *kvadratické síto*, případně další faktorizační algoritmy pro dílčí faktorizace menších čísel.

V první fázi vygenerujeme dvojici ireducibilních (ne nutně monických) polynomů $f(x) = \sum_{i=0}^{d} a_i x^i$ a lineární $g(x)$, které sdílejí společný kořen $\tilde{m}$ modulo N. Dále budeme jako $F(x, y) = y^d f\left(\frac{x}{y}\right)$ a $G(x, y) = yg\left(\frac{x}{y}\right)$ označovat homogenizace polynomů $f(x)$ a $g(x)$.

Volba optimálního stupně nelineárního polynomu f je sama o sobě netriviální úlohou. O přehledovou analýzu tohoto problému se pokusíme v kapitole 4.1.2. V tuto chvíli přijměme fakt, že volíme $d \geq 4$, které je vhodné pro uvažovaná N.

Buď $p, m \in \mathbb{N}$ a položme $g(x) = px - m$. Pro $p \nmid N$ je $\tilde{m} \equiv \frac{m}{p} \bmod N$ kořen $g(x)$ modulo N. Vygenerovat nelineární polynom f, který by s lineárním g sdílel takový kořen $\tilde{m}$, umožňuje například tzv. *Kleinjungův algoritmus*, kterým se budeme podrobněji zabývat v dalších kapitolách. V současnosti se jedná o nejefektivnější známou metodou generování polynomů pro číselné síto.

Dále mějme $\alpha \in \mathbb{C}$ kořen $f(x)$, tudíž $\mathbb{Q}[\alpha] = K$ je číselné těleso (definice 1). Jako poslední definujme okruhový homomorfismus $\phi\colon \mathbb{Z}[\alpha] \to \mathbb{Z}_N$ definovaný pro $\mathbb{Z}[\alpha] \ni \beta = \sum_{i=0}^{d-1} b_i \alpha^i$, kde $b_i \in \mathbb{Z}$, jako

$$\phi(\beta) \equiv \sum_{i=0}^{d-1} b_i \tilde{m}^i \equiv \sum_{i=0}^{d-1} b_i \left(\frac{m}{p}\right)^i \pmod{N}.$$

Později se ukáže, že by bylo výhodnější pracovat s nelineárním polynomem monickým. Nicméně polynom $f(x)$ převedeme na monický polynom jednoduše jako $\hat{f}(x) = a_d^{d-1} f(\frac{x}{a_d})$, pak $\hat{\alpha} = a_d \alpha$ je kořenem monického polynomu $\hat{f}(x)$ a tudíž $Q[x]\big/_{(\hat{f})} = \mathbb{Q}[\hat{\alpha}] = \bar{K}$ je také číselné těleso. Jelikož je $\hat{\alpha}$ kořen $\hat{f}$ celočíselným násobkem α kořenu f, platí $\mathbb{Q}[\alpha] = \mathbb{Q}[\hat{\alpha}]$. Oba polynomy f i $\hat{f}$ tedy definují stejné číselné těleso $K = \bar{K}$. Proč bude výhodnější pracovat s $\hat{f}$ v $\mathbb{Z}[\hat{\alpha}]$ ukážeme později.

Nabízí se otázka, proč generujeme nemonické polynomy, když je pohodlnější pracovat s polynomy monickými? Nemonické polynomy přinášejí výhodu menších koeficientů i společného kořene $\tilde{m}$ modulo N, což se (v tuto chvíli pouze heuristicky) jeví jako výpočetně výhodnější. V kapitole 3.2 nejdříve podrobně vysvětlíme Kleinjungův algoritmus generování polynomů a poté využijeme vlastnosti těchto polynomů k tomu, abychom v části 4.1.2 ukázali závislost velikosti koeficientů, volby stupně polynomu a asymptotické výpočetní složitosti algoritmu číselného síta.

2.2.2 Prosívací fáze

Se zavedeným aparátem se nyní podívejme přesněji na myšlenku z úvodu této podkapitoly. Máme homomorfismus $\phi\colon \mathbb{Z}[\alpha] \to \mathbb{Z}_N$ a chceme najít k prvků $\theta_i \in \mathbb{Z}[\alpha]$, že bude zároveň platit

$$\begin{aligned} \prod_{i=1}^{k} \theta_i &= \gamma^2 \quad \text{pro } \gamma \in \mathbb{Z}[\alpha], \\ \prod_{i=1}^{k} \phi(\theta_i) &= v^2 \quad \text{pro } v \in \mathbb{Z}. \end{aligned} \tag{2.1}$$

Řekněme, že budeme hledat θ ve tvaru $a - b\alpha \in \mathbb{Z}[\alpha]$, a vyřešíme nejprve druhou kongruenci, tzv. *celočíselnou část*. Z definice homomorfismu ϕ hledáme množinu dvojic (a, b) takových, že

$$\prod_{\{(a,b)\} \subset \mathbb{Z}^2} (a - b\tilde{m}) \equiv v^2 \pmod{N}. \tag{2.2}$$

Jak takovou podmnožinu $\mathbb{Z}^2$ najít? Pokud by nějaký prvek $a_1 - b_1\tilde{m}$ měl velkého prvočinitele, museli bychom najít jiný prvek $a_2 - b_2\tilde{m}$, který by obsahoval stejného velkého prvočinitele. Proto se jeví jako výhodné omezit množinu prvočinitelů, které chceme uvažovat v rozkladu všech $a - b\tilde{m}$. Zvolme tedy nějakou vhodnou mez $b_{\mathbb{Z}} \in \mathbb{N}$ a uvažujme tzv. *celočíselnou faktorizační bázi*

$$\mathrm{B}_{\mathbb{Z}} = \{-1\} \cup \{p \text{ prvočíslo} \mid p \leq b_{\mathbb{Z}}\}.$$

Fakt, že $a - b\tilde{m}$ je součinem prvků z $\mathrm{B}_{\mathbb{Z}}$, je zřejmě ekvivalentní tomu, že $|a - b\tilde{m}|$ je $b_{\mathbb{Z}}$-hladké (definice 2).

V první fázi jsme vygenerovali polynom $g(x) = px - m$. Uvědomme si, že platí

$$\frac{1}{p}G(a,b) \equiv \frac{1}{p}(pa - mb) \equiv a - b\frac{m}{p} \equiv a - b\tilde{m} \bmod N.$$

Pro nalezení množiny $S = \{(a,b) \in \mathbb{Z}^2 \mid \mathrm{NSD}(a,b) = 1\}$ splňující rovnost (2.2) proto budeme hledat dvojice (a,b), pro které je hodnota $|G(a,b)|$ $b_{\mathbb{Z}}$-hladká. Nesoudělnost dvojic (a,b) požadujeme kvůli vyloučení triviální závislosti.

Kolik takových $b_{\mathbb{Z}}$-hladkých hodnot musíme najít, abychom měli zajištěnou existenci řešení rovnosti (2.2)? V první řadě koeficient $\frac{1}{p} \bmod N$ nemusí být nutně $b_{\mathbb{Z}}$-hladký, jelikož se ale vyskytuje v prvočíselném rozkladu každého $a - b\tilde{m}$, stačí zajistit, aby množina S měla sudý počet prvků. V druhé řadě mějme $b_{\mathbb{Z}}$-hladké $c = \prod_{i=1}^{\pi(b_{\mathbb{Z}})} p_i^{e_i}$, kde $\pi(b_{\mathbb{Z}})$ značí počet prvočísel menších nebo rovných $b_{\mathbb{Z}}$, a uvažujme vektor exponentů $\vec{v}(c) = (e_1, \ldots, e_{\pi(b_{\mathbb{Z}})})$. Pro $c_1, \ldots, c_k$ $b_{\mathbb{Z}}$-hladkých hodnot platí

$$\prod_{i=1}^{k} c_i \text{ je čtverec } \Leftrightarrow \sum_{i=1}^{k} \vec{v}(c_i) \text{ má samé sudé souřadnice.}$$

Pokud tedy všechny vektory $\vec{v}(c)$ redukujeme modulo 2 a budeme na ně nahlížet ve vektorovém prostoru $(\mathbb{Z}_2)^{\pi(b_{\mathbb{Z}})}$, můžeme problém hledání neprázdné množiny čísel, jejichž součin je čtverec, převést na hledání lineární závislosti v množině vektorů ze $(\mathbb{Z}_2)^{\pi(b_{\mathbb{Z}})}$. Z lineární algebry víme, že pokud je velikost množiny vektorů větší než dimenze příslušného vektorového prostoru, pak jsou tyto vektory lineárně závislé. Stačí nám tedy najít $\pi(b_{\mathbb{Z}}) + 1$ nesoudělných dvojic (a,b).

Máme tedy vyřešenou celočíselnou část a vrátíme se k první rovnosti (2.1), tzv. *algebraické části*. Nyní vysvětlíme výhodu práce s monickým $\hat{f}$ v $\mathbb{Z}[\hat{\alpha}]$. Předpokládejme, že bychom hledali $\theta = a - b\alpha \in \mathbb{Z}[\alpha]$ tak, aby $\prod_i \theta_i$ byl čtverec v $\mathbb{Z}[\alpha]$. Rádi bychom využili podobný argument jako výše v $\mathbb{Z}$, nicméně obecně $\mathbb{Z}[\alpha]$ nemusí být Gaussův obor, tudíž nemusí existovat jednoznačný rozklad na ireducibilní činitele. Budeme tedy místo toho pracovat v okruhu algebraických celých čísel $\mathbb{Z}_K$ (definice 3), což je Dedekindův obor, a místo prvků $a - b\alpha$ budeme pracovat s ideály $(a - b\alpha)\mathbb{Z}_K$. V Dedekindově oboru máme zajištěn jednoznačný rozklad ideálů na prvoideály. Jediný problém je, že α není celistvý, tedy $\mathbb{Z}[\alpha] \not\subseteq \mathbb{Z}_K$. Ale máme k dispozici celistvý $\hat{\alpha} = a_d\alpha$, a tudíž $\mathbb{Z}[\hat{\alpha}] \subseteq \mathbb{Z}_K$. Také $(a - b\alpha)\mathbb{Z}_K$ je lomený ideál, ale lze jej zapsat následovně

$$(a - b\alpha)\mathbb{Z}_K = (a_d \cdot a_d^{-1})(a - b\alpha)\mathbb{Z}_K = (aa_d - b\hat{\alpha})\mathbb{Z}_K \cdot (a_d\mathbb{Z}_K)^{-1}.$$

Ne zcela triviální teorie, která je již popsaná podrobně v pracích [22, 24], vyústí v následující větu, jež poskytuje návod na prosívání v algebraické části. Přebíráme zavedené značení a definice 4–7.

Věta 10. *Buď p nespeciální prvočíslo, $p \nmid a_d$, kde a_d je vedoucí koeficient polynomu f, dále buď $\hat{f} = \prod t_i^{e_i} \bmod p$ rozklad na ireducibilní polynomy a $a, b \in \mathbb{Z}$ nesoudělné. Pak platí následující:*

- *$p\mathbb{Z}_K = \prod_i P_i^{e_i}$, kde $P_i = p\mathbb{Z}_K + t_i(\hat{\alpha})\mathbb{Z}_K$ jsou po dvou různé prvoideály nad p. Navíc jsou to právě všechny prvoideály nad p.*
- *Pro každý prvoideál P nad p existuje právě jedno $r_P \in \mathbb{Z}_p$, které je kořenem f modulo p. Pokud $P \neq P'$, pak také $r_P \neq r_{P'}$.*

- *Pro dané* r_P *je* $a \equiv br_P \bmod p \Leftrightarrow aa_d - b\hat{\alpha} \in P$.
- $\mathrm{val}_P\left((aa_d - b\hat{\alpha})\mathbb{Z}_K\right) = \mathrm{val}_p\left(\mathcal{N}(a - b\alpha)\right)$.

Samotné prosívání v algebraické části probíhá analogicky prosívání v celočíselné části. Nejprve stanovíme mez hladkosti $b_{\mathbb{A}}$, procházíme nespeciální prvočísla menší než tato mez a ověřujeme, zda f má nějaký kořen modulo p. Tyto kořeny si zaznamenáváme a do faktorizační báze $\mathrm{B}_{\mathbb{A}}$ přidáváme všechny prvoideály z rozkladu $p\mathbb{Z}_K$. Následně hledáme nesoudělné dvojice (a, b), pro které je $(aa_d - b\hat{\alpha})\mathbb{Z}_K$ součinem ideálů z $\mathrm{B}_{\mathbb{A}}$, což je dle uvedené věty ekvivalentní s tím, zda $\mathcal{N}(a - b\alpha)$ je $b_{\mathbb{A}}$-hladké. Podle vztahu (1.1) je $\mathcal{N}(a - b\alpha) = \frac{1}{a_d}|F(a, b)|$, a protože předpokládáme $p \nmid a_d$ (navíc lze ukázat, že pro nespeciální p je toto splněno vždy), pak stačí hledat $b_{\mathbb{A}}$-hladké $|F(a, b)|$. Rozklad $|F(a, b)|$ na prvočinitele dá navíc také rozklad $(aa_d - b\hat{\alpha})\mathbb{Z}_K$ na prvoideály.

Definice 10 (Hladká relace)**.** *Nesoudělné dvojice* (a, b)*, pro které* $|F(a, b)|$ *je* $b_{\mathbb{A}}$*-hladké a zároveň* $|G(a, b)|$ *je* $b_{\mathbb{Z}}$*-hladké, budeme dále označovat jako* hladké relace.

Existují různé postupy, jak konkrétně volit a procházet prosívací oblast, pro efektivní hledání hladkých relací. Konkrétní detaily těchto metod však nejsou nosné pro ústřední téma této práce, a proto je nebudeme uvádět. Stejně tak vynecháme případ nespeciálních prvočísel.

2.2.3 Fáze zpracování relací

Abychom dosáhli úspěchu v dalších částech síta a nalezli hledané čtverce, musíme mít dostatek hladkých relací. U celočíselné části jsme již odvodili, že postačuje $\pi(b_{\mathbb{Z}}) + 1$ relací, nicméně chceme, aby to byl sudý počet. Analogickým argumentem dostáváme, že $\pi(b_{\mathbb{A}}) + 1$ je dostatečný počet relací pro jistotu, že $\prod_{\{(a,b)\}} (aa_d - b\hat{\alpha})\mathbb{Z}_K$ bude kvadrát. Pro vypořádání se s lomeným ideálem $(a_d\mathbb{Z}_K)^{-1}$ také postačuje, když počet hladkých relací bude sudý. Hledáme tedy alespoň $2\pi(\max\{b_{\mathbb{A}}, b_{\mathbb{Z}}\}) + 2$ hladké relace. Množinu všech hladkých relací značíme S.

Umíme nyní sestavit $\prod_S (a - b\alpha)\mathbb{Z}_K = \prod_i P_i^{2e_i} = I^2$ pro nějaký ideál I, který však nemusí být nutně hlavní. Navíc chceme vlastně znát pouze $\gamma^2 = \prod_S (a - b\alpha)$, jenže takové γ vůbec nemusí existovat. Kvůli tomu po prosívací fázi dopočítáme ještě tzv. *kvadratické charaktery* (podrobnosti viz [22, kap. 2.5]), které přidáme k faktorizačním bázím a zařídíme tak s dostatečnou pravděpodobností

$$\prod_S (a - b\alpha) = \gamma^2$$

pro $\gamma \in \mathbb{Z}_K$. Zvýší se tím také počet hladkých relací, které musíme nalézt (nesmíme navíc zapomenout na podmínku sudého počtu hladkých relací).

Pro každou nalezenou hladkou relaci spočítáme nyní její vektor exponentů tak, že v první části souřadnic budou prvoideály z algebraické faktorizační báze a kvadratické charaktery, druhou část souřadnic tvoří prvočísla z celočíselné báze. Všechny vektory uspořádáme do matice, se kterou budeme dále pracovat.

2.2.4 Lineární fáze

Myšlenka převedení vektorů exponentů do vektorového prostoru nad $\mathbb{Z}_2$, jak jsme ji zavedli výše, má uplatnění i v lineární fázi. Řešíme totiž zcela obecný problém hledání lineární závislosti, na který lze uplatnit známé algoritmy lineární algebry. Ačkoliv by se mohlo zdát, že se jedná o triviální problém hledání řešení homogenní soustavy, je třeba si uvědomit, že řídká binární matice, se kterou pracujeme, může mít rozměry v řádech milionů, což standardním algoritmům, jako je např. Gaussova eliminace, značně ubírá na praktičnosti. Využívají se proto efektivní iterační metody, které využívají právě řídkosti matice, jako např. Lanczosova bloková metoda [17] nebo Wiedemanův algoritmus [26], případně jeho paralelizovatelné vylepšení Coppersmithem [8].

2.2.5 Odmocňovací fáze

S jistou pravděpodobností závislou na kvadratických charakterech skončí lineární fáze úspěchem a nalézáme kvadrát v $\mathbb{Z}_K$. Rádi bychom však získali kvadrát ze $\mathbb{Z}[\alpha]$, což nám umožní následující lemma.

Lemma 11 ([22, věta 2.26])**.** *Buď $K = \mathbb{Q}[\hat{\alpha}]$ číselné těleso, $\mathbb{Z}_K$ obor algebraických celých čísel a $\hat{f}$ minimální polynom $\hat{\alpha} \in \mathbb{C}$ nad $\mathbb{Z}$. Pak $\hat{f}'(\hat{\alpha})\mathbb{Z}_K \subseteq \mathbb{Z}[\hat{\alpha}]$.*

Z lineární fáze dostáváme podmnožinu $\bar{S}$ množiny hladkých relací S, pro kterou $\prod_{\bar{S}} (aa_d - b\hat{\alpha})$ je čtverec v $\mathbb{Z}_K$. Z uvedeného lemmatu dostáváme, že

$$\left(\hat{f}'(\hat{\alpha})\right)^2 \cdot \prod_{\bar{S}} (aa_d - b\hat{\alpha})$$

je čtverec v $\mathbb{Z}[\hat{\alpha}]$. Protože $\bar{S}$ má sudý počet prvků, řekněme $\left|\bar{S}\right| = 2k$, lze tento výraz upravit na

$$\left(\hat{f}'(a_d\alpha)\right)^2 \cdot \prod_{\bar{S}} (aa_d - ba_d\alpha) = \left(\hat{f}'(\hat{\alpha})a_d^k\right)^2 \cdot \prod_{\bar{S}} (a - b\alpha)\,,$$

což je čtverec v $\mathbb{Z}[\alpha]$.

Nyní máme množinu hladkých relací $\bar{S}$, pro kterou platí

$$\begin{aligned} \left(\hat{f}'(a_d\alpha)a_d^k\right)^2 \cdot \prod_{\bar{S}} (a - b\alpha) &= \gamma^2 \quad \text{pro } \gamma \in \mathbb{Z}[\alpha] \\ \prod_{\bar{S}} (a - b\tilde{m}) &= v^2 \quad \text{pro } v \in \mathbb{Z}. \end{aligned} \tag{2.3}$$

Nejprve odmocníme celočíselnou část. Z vektorů exponentů známe prvočíselný rozklad $v^2 = \prod_i p_i^{2e_i}$. Pro každou prvočíselnou mocninu $p_i^{e_i}$ spočítáme zbytek $p_i^{e_i} \bmod N$ rychlým modulárním mocněním a vynásobíme v $\mathbb{Z}_N$.

Jádro odmocňovací fáze nicméně spočívá v nalezení odmocniny v algebraické části. V té sice také v principu známe rozklad γ^2, nicméně jeho přímé vyčíslení by mohlo být výpočetně náročnější než celý zbytek algoritmu. Navíc nás více než přesná hodnota γ zajímá $\phi(\gamma) \equiv u \bmod N$. Efektivní metodu řešení navrhl P. Montgomery v článku [20]. Tuto i další známé metody popisuje Perůtka [22, kap. 3.6.].

Než odvodíme kongruenci dvou čtverců modulo N, pomocí kterých zkusíme faktorizovat N, ukážeme několik vztahů plynoucích z vlastností homogenních polynomů a jejich derivací.

Lemma 12. *Mějme* $f(x) = \sum_{i=0}^{d} a_i x^i$, *homogenizaci* $F(x,y)$ *a* $\hat{f}(x) = a_d^{d-1} f\left(\frac{x}{a_d}\right)$ *monický. Pro derivace* $\hat{f}'$ *a* $F_x(x,y) = \frac{\partial F}{\partial x}(x,y)$ *platí*

- $\hat{f}'(x) = \frac{1}{a_d} F_x(x, a_d)$;
- $F_x(cx, cy) = c^{d-1} F_x(x,y)$;
- $F_x\left(\frac{x}{y}, 1\right) = \frac{1}{y^{d-1}} F_x(x,y)$.

Důkaz. Z definice homogenního polynomu plyne $F(x, a_d) = a_d^d f\left(\frac{x}{a_d}\right) = a_d \hat{f}(x)$. Proto je $F_x(x, a_d) = a_d \hat{f}'(x)$.

Parciální derivaci F podle x lze zapsat jako $F_x(x,y) = y^d \sum_{i=1}^{d} i a_i \frac{x^{i-1}}{y^i}$. Dosazením do tého rovnosti dostáváme druhé dva vztahy ze znění lemmatu:

$$F_x(cx, cy) = (cy)^d \sum_{i=1}^{d} i a_i \frac{(cx)^{i-1}}{(cy)^i} = \frac{c^d}{c} \cdot y^d \sum_{i=1}^{d} i a_i \frac{x^{i-1}}{y^i} = c^{d-1} F_x(x,y),$$
$$F_x\left(\frac{x}{y}, 1\right) = \sum_{i=1}^{d} i a_i \left(\frac{x}{y}\right)^{i-1} = \frac{1}{y^d} \cdot y^d \sum_{i=1}^{d} i a_i x^{i-1} \frac{y}{y^i} = \frac{1}{y^{d-1}} F_x(x,y).$$

□

Hodnotu $\phi(\gamma^2)$ lze nyní vyjádřit dvěma různými způsoby. Z vlastností homomorfismu je nejprve

$$\phi(\gamma^2) \equiv (\phi(\gamma))^2 \equiv u^2 \pmod{N},$$

ale zároveň také s využitím vztahů (2.3) a lemmatu 12

$$\begin{aligned}
\phi(\gamma^2) &\equiv \phi\left(\left(\hat{f}'(a_d\alpha) a_d^k\right)^2 \cdot \prod_{\bar{S}} (a - b\alpha)\right) \equiv \\
&\equiv \left(\phi\left(F_x(a_d\alpha, a_d) a_d^{k-1}\right)\right)^2 \cdot \prod_{\bar{S}} \left(a - b\frac{m}{p}\right) \equiv \\
&\equiv \left(F_x(a_d \frac{m}{p}, a_d) a_d^{k-1}\right)^2 \cdot v^2 \equiv \left(\frac{a_d^{d+k-2}}{p^{d-1}} F_x(m,p) \cdot v\right)^2.
\end{aligned}$$

Můžeme se proto pokusit faktorizovat N pomocí NSD $\left(N, u \pm \frac{a_d^{d+k-2}}{p^{d-1}} F_x(m,p) v\right)$.

Ukazuje se tedy, že volba vhodných polynomů f a g je pro faktorizaci pomocí číselného síta zcela zásadní otázkou, jelikož přímo ovlivňuje efektivitu prosívání, které je výpočetně nejnáročnější fází celého algoritmu číselného síta. V následujících kapitolách podrobně rozebereme, jak vhodné polynomy generovat Kleinjungovým algoritmem a jak se volba některých vstupních parametrů projeví na efektivitě celého číselného síta.

Kapitola 3

Generování polynomů

Z popisu algoritmu obecného číselného síta v předchozí kapitole je zřejmé, že úspěšnost při hledání hladkých relací (neboli *výtěžnost polynomů*), a tudíž rychlost a efektivita celého algoritmu, velice úzce souvisí s kvalitou generovaných polynomů. Optimalizací a hodnocením vlastností polynomů a výběrem nejlepšího kandidáta se budeme zabývat v kapitole následující. Nyní ve stručnosti představíme tzv. *base-m metodu* pro generování polynomů, která byla navrhnuta pro užití ve *speciálním číselném sítu* (Pollard [23]), jež bylo přímým předchůdcem obecného číselného síta (GNFS), v kterém lze tuto metodu také použít. V roce 2006 však T. Kleinjung představil v článku [16] nový algoritmus pro generování polynomů, který z původní metody vychází, nicméně zavádí nové koncepty, které přináší větší efektivitu algoritmu. Detaily Kleinjungova algoritmu se budeme zabývat po zbytek textu.

Připomeňme, že v první fázi algoritmu číselného síta hledáme dva nesoudělné ireducibilní polynomy f a g, které sdílí kořen $\tilde{m}$ modulo N, pro rozkládané složené N. V druhé fázi se pak snažíme najít co nejvíce hladkých relací, tedy párů $(a,b) \in \mathbb{Z}^2$ nesoudělných čísel, pro které $|F(a,b)|$ a $|G(a,b)|$ jsou hladké hodnoty. V dalším textu budeme společnou mez hladkosti označovat jako B, nicméně je možné volit i dvě meze B_f a B_g pro každý polynom zvlášť. Připomeňme, že značením $F(x,y)$, $G(x,y)$ myslíme homogenní polynomy příslušné k polynomům f a g. Proces hledání hladkých relací je obvykle označován jako *prosívání* a je časově nejnáročnější fází GNFS. Fáze generování polynomů musí tudíž vyřešit, jak hledat polynomy f a g splňující výše uvedené podmínky a jak ze všech nalezených kandidátů vybrat takový pár, který minimalizuje dobu prosívání.

3.1 Base-m metoda

Base-m metoda využívá prostého rozkladu faktorizovaného čísla N v bázi m. Mějme vhodně zvolené d, obvykle $d \in \{4, 5, 6\}$, v závislosti na velikosti rozkládaného N, zvolme $m \approx \sqrt[d]{N}$ a zapišme N v bázi m, tedy

$$N = a_d m^d + a_{d-1} m^{d-1} + \ldots + a_1 m + a_0, \quad \text{kde} \quad \forall i \quad 0 \leq a_i < m.$$

Potom můžeme položit $f(x) = \sum_{i=0}^{d} a_i x^i$ a platí

$$f(m) = \sum_{i=0}^{d} a_i m^i \equiv 0 \pmod{N},$$

tedy m je kořen f modulo N. Proto volíme lineární polynom jako $g(x) = x - m$, aby oba polynomy sdílely stejný kořen m modulo N.

V případě, že base-m rozkladem vznikne polynom f, který není ireducibilní, získáme rozklad N přímo, protože pro $f(x) = f_1(x) \cdot f_2(x)$ je hledaný rozklad N roven

$$N = f(m) = f_1(m) \cdot f_2(m).$$

Původní myšlenka base-m metody vycházela z předpokladu, že báze m je volena dostatečně velká, aby polynom f byl monický, což umožňuje jednodušší práci v číselném sítu. Jak jsme však ukázali v předchozí kapitole, podmínka moničnosti nelineárního polynomu není nutná, a navíc praxe ukázala, že nemonické polynomy vedou dokonce ke zrychlení celého algoritmu. Pro velká N (řádově nad 100 decimálních cifer) ovšem polynomy generované base-m metodou přestávají mít dostatečnou výtěžnost.

3.2 Kleinjungův algoritmus

Thorsten Kleinjung ve svém článku [16] přichází s novým konceptem generování polynomů, který sice z původní base-m metody částečně vychází, ale snaží se zhodnotit nové poznatky o hodnocení výtěžnosti polynomů, které přináší Murphy ve své dizertaci [21]. Podrobněji se metodami hodnocení polynomů budeme zabývat až v kapitole 4, pro vysvětlení motivace Kleinjungova postupu ale již nyní uvádíme, že u nelineárního polynomu nám jde především o co nejmenší koeficienty u monomů vysokého stupně a fakt, aby měl polynom co nejvíce kořenů modulo malé prvočíselné mocniny.

Těžiště algoritmu spočívá v záměně base-m rozvoje za tzv. base-(m, p) rozvoj rozkládaného čísla N a v přechodu výhradně na nemonické polynomy. Popišme nyní podrobně, co myslíme pojmem base-(m, p) rozvoje a jak ho najít.

Řekneme, že jsme nalezli base-(m, p) rozvoj čísla N, pokud pro dané hodnoty $d, p, m \in \mathbb{N}$ najdeme koeficienty $a_0, \ldots, a_d \in \mathbb{Z}$ takové, že platí

$$N = a_d m^d + a_{d-1} m^{d-1} p + \ldots + a_1 m p^{d-1} + a_0 p^d.$$

Nalezené koeficienty pak uvažujeme jako koeficienty polynomu $f(x) = \sum_{i=0}^{d} a_i x^i$. Ekvivalentně, base-$(m, p)$ rozvoj čísla N říká, že chceme vyjádřit $N = F(m, p)$, kde F je homogenní polynom stupně d. Dále budeme uvažovat pouze m a p nesoudělná, protože pokud by existoval společný dělitel $l = \mathrm{NSD}(m, p)$, pak l^d dělí N. Bez újmy na obecnosti můžeme také předpokládat $1 < p < m < N$. Parametry d, p a m ale nelze volit zcela libovolně náhodně, protože ne pro všechny kombinace base-(m, p) rozvoj čísla N existuje. Připomeňme, že chceme generovat polynomy, které mají malé koeficienty u monomů vysokého stupně. To nabízí myšlenku postavit generování polynomů na iteraci přes všechny (vhodné) kandidáty na vedoucí koeficient a_d a parametry m a p dopočítávat tak, aby platilo

$$N \equiv a_d m^d \bmod p. \tag{3.1}$$

Získáváme navíc jednoduchou kontrolu nad velikostí vedoucího koeficientu. Jak využít princip base-(m, p) rozvoje pro konstrukci polynomu, u kterého jsou i další koeficienty malé, ukazuje následující věta.

Věta 13. *Nechť $N, d, a_d \in \mathbb{N}$ a označme $m_0 = \sqrt[d]{\frac{N}{a_d}}$. Zvolme nesoudělná $m, p \in \mathbb{N}$ tak, že $m \geq m_0$ a platí $N \equiv a_d m^d \pmod{p}$. Pak existuje homogenní polynom $F(x,y) = \sum_{i=0}^{d} a_i x^i y^{d-i}$, pro nějž je*

(i) $F(m,p) = N$,

(ii) $|a_{d-1}| < p + da_d \frac{m-m_0}{p}$ *a*

(iii) $|a_i| < p + m$ *pro* $0 \leq i \leq d-2$.

Důkaz. Nejprve provedeme konstruktivní důkaz existence takového polynomu. Poté ukážeme, že takto zvolený polynom splňuje podmínky na velikost koeficientů.

ad (i) Pro dané parametry dokážeme existenci $r_i, a_i \in \mathbb{Z}$ takových, že $p \mid r_i - a_i m^i$ pro všechna $1 \leq i \leq d$ a platí $N = \sum_{i=0}^{d} a_i m^i p^{d-i} = F(m,p)$.

Položme $r_d = N$ a dále rekurentně pro $i = d-1, \ldots, 0$

$r_i = \frac{r_{i+1} - a_{i+1} m^{i+1}}{p}$ a
$a_i = \frac{r_i}{m^i} + \delta_i$ pro $0 \leq \delta_i < p$ takové, že platí $r_i \equiv a_i m^i \pmod{p}$.

Existenci r_{d-1} máme zajištěnu z předpokladů věty a vhodnou volbou δ_i zajistíme, že podíl $\frac{r_{i+1} - a_{i+1} m^{i+1}}{p}$ bude celočíselný i v každé další iteraci. Navíc pro libovolné $0 \leq i \leq d$ platí

$$r_i = \frac{N - a_d m^d - a_{d-1} m^{d-1} p - \ldots - a_{i+1} m^{i+1} p^{d-i-1}}{p^{d-i}}$$
$$N = a_d m^d + a_{d-1} m^{d-1} p + \ldots + a_{i+1} m^{i+1} p^{d-i-1} + r_i p^{d-i},$$

což jsme chtěli ukázat.

ad (ii) Z výpočtu koeficientu a_{d-1} máme $|a_{d-1}| = \frac{|r_{d-1}|}{m^{d-1}} + \delta_{d-1}$. Odhadněme proto nejdříve velikost

$$|r_{d-1}| = \frac{1}{p} \left| N - a_d m^d \right| = \frac{a_d}{p} \left| \frac{N}{a_d} - m^d \right|.$$

Z volby m_0 a předpokladu $m_0 \leq m$ dostáváme odhad

$$|r_{d-1}| = \frac{a_d}{p} \left| \frac{N}{a_d} - m^d \right| = \frac{a_d}{p} \left| m_0^d - m^d \right| = \frac{a_d}{p} (m^d - m_0^d) < \frac{a_d}{p} (m - m_0) d m^{d-1}.$$

Tuto nerovnost dosadíme do vztahu pro $|a_{d-1}|$ a díky nerovnosti $\delta_i < p$ dostáváme

$$|a_{d-1}| = \frac{|r_{d-1}|}{m^{d-1}} + \delta_{d-1} < \frac{a_d}{p} (m - m_0) d + \delta_{d-1} < \frac{a_d}{p} (m - m_0) d + p,$$

což je hledaný odhad.

ad (iii) Podobným způsobem ukážeme vztah pro další $|a_i|$. Nejdříve si všimneme, že platí

$$|a_i| = \frac{|r_i|}{m^i} + \delta_i \Leftrightarrow \delta_i m^i = a_i m^i - r_i.$$

S využitím tohoto faktu a znovu nerovnosti $0 \leq \delta_i < p$ dostáváme odhad pro obecné r_i, konkrétně

$$|r_i| = \frac{1}{p}\left|r_{i+1} - a_{i+1}m^{i+1}\right| = \frac{1}{p}\delta_{i+1}m^{i+1} < m^{i+1}.$$

Dosazením do již známých vztahů dostáváme hledaný odhad

$$|a_i| = \frac{|r_i|}{m^i} + \delta_i < \frac{m^{i+1}}{m^i} + \delta_i < m + p.$$

□

Zvolíme-li $f(x)$ jako dehomogenizaci $F(x, y)$, uvedená věta nám teoreticky dává přesný rekurzivní návod, jak hledat vhodné nelineární polynomy $f(x)$. Pro dané a_d a p dopočítáme všechna vhodná m jako řešení kongruence (3.1) a spočítáme base-(m, p) rozvoj N. Princip base-(m, p) rozkladu budeme v následujícím textu často používat, proto jej pro přehlednost formulujeme jako algoritmus 3.1.

Algoritmus 3.1 Base-(m, p) rozklad čísla N

input : rozkládané číslo $N, d = \deg(f)$, vedoucí koeficient a_d, parametry rozkladu m, p

output : koeficienty base-(m, p) rozkladu

1 : inicializuj pole $R[0..d], A[0..d]$

2 : $R[d] \leftarrow N$

3 : $A[d] \leftarrow a_d$

4 : $M \leftarrow m^d$

5 : **for** $(i = d - 1, \ldots, 0)$

6 : $\quad R[i] \leftarrow \dfrac{R[i+1] - A[i+1] \cdot M}{p}$

7 : $\quad$ najdi vhodné $\delta \in \mathbb{Z}_p$

8 : $\quad A[i] \leftarrow \dfrac{R[i]}{M} + \delta$

9 : $\quad M \leftarrow \dfrac{M}{m}$

10 : **return** A

Poznámka. Uvedený postup má jedno drobné implementační úskalí při výpočtu koeficientů base-(m, p) rozvojem, který podrobněji popisujeme v kapitole 6.1.1.

Lineární polynom je volen $g(x) = px - m$, jak jsme psali v kapitole 2.2.1. Ověřme, že také polynom f má kořen $\tilde{m} = \frac{m}{p} \bmod N$ společný s polynomem g. Z věty 13 máme vztah $F(m, p) = N$. Pokud platí $p \nmid N$, pak hodnota p^d má inverz modulo N, a tudíž platí

$$\frac{F(m,p)}{p^d} = f\left(\frac{m}{p}\right) = \frac{N}{p^d} \equiv 0 \pmod{N}$$

a $\tilde{m} = \frac{m}{p}$ je tedy opravdu společný kořen f i g modulo N.

Prakticky tímto způsobem dostáváme velké množství trojic (a_d, p, m), každá definuje dvojici polynomů f a g. Bylo by tedy vhodné již v rámci algoritmu aplikovat nějaká omezení, která by filtrovala jen ty nejlepší polynomy.

3.2.1 Filtr polynomiálních kandidátů

V kapitole 4 se budeme podrobněji zabývat hodnocením vygenerovaných polynomů a ukáže se, že lépe jsou hodnoceny polynomy, které mají menší koeficienty u monomů vyššího stupně. Zvolme vhodně horní meze $a_{d,max}, a_{d-1,max}$ a $a_{d-2,max}$ prvních tří koeficientů polynomu $f(x)$. Spočítáme jen první tři iterace base-(m,p) rozvoje a pouze pro kandidáty splňující daná omezení opravdu spočítáme celý base-(m,p) rozvoj.

Pro vhodnou volbu p je navíc možné pracovat efektivně s celou dávkou kandidátů najednou.

Věta 14. *Zvolme $p = \prod_{i=1}^{l} p_i$, kde $p_i \equiv 1 \bmod d$ jsou prvočísla menší než mez B_p tak, aby platilo $\mathrm{NSD}(a_d N, p) = 1$ a zároveň $p \leq a_{d-1,max}$. Pak kongruence (3.1), tedy*

$$N \equiv a_d x^d \pmod{p},$$

má buď d^l řešení nebo nemá řešení žádné. Navíc lze všechna tato řešení zapsat ve tvaru

$$x_{\vec{\mu}} = x_{(\mu_1,\ldots,\mu_l)} = \sum_{i=1}^{l} x_{i,\mu_i}, \tag{3.2}$$

kde $1 \leq \mu_i \leq d$, pro každý jednotlivý sčítanec platí $0 \leq x_{i,\mu_i} < p$ a $\frac{p}{p_i} \mid x_{i,\mu_i}$, přičemž $\{x_{i,1} \bmod p_i, x_{i,2} \bmod p_i, \ldots, x_{i,d} \bmod p_i\}$ je všech d řešení kongruence $N \equiv a_d x^d \bmod p_i$.

Důkaz. Nejprve objasníme, jak je to s kořeny $N \equiv a_d x^d \bmod p_i$ pro libovolné pevné $i \in \{1, \ldots, l\}$. Z teorie čísel plyne pozorování, že konečné těleso $\mathbb{Z}_{p_i}$ obsahuje primitivní d-tou odmocninu z jedné právě tehdy, když d dělí $p_i - 1$. Jelikož volíme $p_i \equiv 1 \bmod d$, budou v našem případě d-té odmocniny z jedné existovat. Předpokládáme $\mathrm{NSD}(a_d N, p) = 1$, pak musí také $\mathrm{NSD}(a_d, p_i) = 1$. Proto existuje $c \in \mathbb{Z}$ takové, že $\mathrm{NSD}(c, p_i) = 1$ a zároveň

$$c \equiv \frac{N}{a_d} \equiv x^d \pmod{p_i}.$$

V případě, že najdeme alespoň jedno řešení kongruence $c \equiv x^d \bmod p_i$, pak vynásobením každou z d odmocnin z jedné dostáváme právě d řešení.

Pro existující řešení získáváme přímočarou aplikací Čínské zbytkové věty v Lagrangeově algoritmu d^l řešení $x_{\vec{\mu}}$ v uvedeném tvaru. □

Z rovnosti (3.2) vidíme, že všech možných d^l řešení, které lze zapsat jako $x_{\vec{\mu}}$ na levé straně, dokážeme vyjádřit jako lineární kombinaci pouhých $d \cdot l$ proměnných x_{i,μ_i} z pravé strany. To umožňuje efektivní práci s celou dávkou polynomů, neboť každé z řešení definuje jednoho polynomiálního kandidáta. Multi-indexovou notací $\vec{\mu}$ proto budeme dále označovat vždy všechny proměnné korespondující s příslušnými řešeními (3.1).

Pokusme se nyní tento dávkový princip uplatnit při kontrole velikosti koeficientů. Z věty 13 plyne, že pro m volené blízko k m_0 umíme najít koeficient a_{d-1} řádově stejně velký jako a_d a další koeficienty řádu m, pokud $p \ll m$. Předpokládáme $\prod_{i=1}^{l} = p < m$, je tedy v našem zájmu zajistit, aby p bylo součinem jen několika malých prvočísel.

Abychom dodrželi předpoklad, že pracujeme s řešeními ležícími blízko m_0, a udrželi tak koeficient a_{d-1} dostatečně malý, zvolme přirozené $\bar{m}$ nejbližší násobek p vyšší než m_0. Pak $m_{\vec{\mu}} = \bar{m} + x_{\vec{\mu}}$ je zřejmě d^l řešení (3.1) blízko m_0. Pro všechna $\mu_j \in \{1, \ldots, d\}$ definujme

$$m_{i,\mu_j} = \begin{cases} \bar{m} + x_{i,\mu_j} & \text{pro } i = 1, \\ x_{i,\mu_j} & \text{pro } i > 1. \end{cases}$$

Tedy i tato posunutá řešení umíme zapsat jako lineární kombinaci dl proměnných

$$m_{\vec{\mu}} = \sum_{i=1}^{l} m_{i,\mu_i}.$$

Podle věty 13 jsou nyní zbylé koeficienty $a_{i,\vec{\mu}}$ pro $0 \leq i \leq d-1$ jednoznačně určeny, navíc jsou díky volbě $m_{\vec{\mu}}$ blízko m_0 koeficienty $a_{d-1,\vec{\mu}}$ dostatečně malé. Nyní chceme vybrat jen ty hodnoty $\vec{\mu}$, pro které budou přiměřeně malé i koeficienty $a_{d-2,\vec{\mu}}$. Pro efektivní postup chceme stále pracovat s celou dávkou současně. Ukážeme, jak najít dl parametrů, které budou efektivně reprezentovat celou dávku d^l koeficientů $a_{d-1,\vec{\mu}}$.

Z algoritmu 3.1 plyne pro libovolné $\vec{\mu}$ platnost

$$a_{d-1,\vec{\mu}} m_{\vec{\mu}}^{d-1} \equiv r_{d-1} \pmod{p} \tag{3.3}$$

a současně

$$r_{d-1} = \frac{N - a_d m_{\vec{\mu}}^d}{p}.$$

Dosazením za r_{d-1} do (3.3), vynásobením obou stran kongruence výrazem $\frac{a_d m_{\vec{\mu}}}{N}$ a aplikací vztahu (3.1) dostáváme vzorec pro výpočet všech koeficientů a_{d-1} příslušných k $\vec{\mu}$ modulo p

$$a_{d-1,\vec{\mu}} \equiv \frac{a_d m_{\vec{\mu}}}{N} \frac{N - a_d m_{\vec{\mu}}^d}{p} \pmod{p}. \tag{3.4}$$

Následující pomocné lemma ozřejmí volbu hledaných dl proměnných pro reprezentaci dávky $a_{d-1,\vec{\mu}}$. Značení přebíráme z předchozího textu.

Lemma 15. *Zvolme pevně $i \in \{1, \ldots, d\}$ a dvojici vektorů lišících se pouze v i-té souřadnici $\vec{\mu} = (\mu_1, \ldots, \mu_i, \ldots, \mu_d)$, $\vec{\bar{\mu}} = (\bar{\mu}_1, \ldots, \bar{\mu}_i, \ldots, \bar{\mu}_d)$, kde $\mu_i \neq \bar{\mu}_i$ a $\mu_j = \bar{\mu}_j$ pro $j \neq i$. Pak rozdíl $a_{d-1,\vec{\mu}} - a_{d-1,\vec{\bar{\mu}}}$ modulo p závisí pouze na souřadnicích μ_i a $\bar{\mu}_i$.*

Důkaz. Zvolme jinou dvojici vektorů lišících se pouze v jedné souřadnici $\vec{\nu} = (\nu_1, \ldots, \nu_i, \ldots, \nu_d)$, $\vec{\bar{\nu}} = (\bar{\nu}_1, \ldots, \bar{\nu}_i, \ldots, \bar{\nu}_d)$, kde $\nu_i \neq \bar{\nu}_i$ a $\nu_j = \bar{\nu}_j$ pro $j \neq i$

a zároveň $\mu_i = \nu_i$ a $\bar{\mu}_i = \bar{\nu}_i$. Nezávislost rozdílu $a_{d-1,\vec{\mu}} - a_{d-1,\vec{\bar{\mu}}}$ modulo p na souřadnicích $j \neq i$ dokážeme, pokud bude platit

$$a_{d-1,\vec{\mu}} - a_{d-1,\vec{\bar{\mu}}} \equiv a_{d-1,\vec{\nu}} - a_{d-1,\vec{\bar{\nu}}} \pmod{p}.$$

Podle vztahu (3.4) je

$$\begin{aligned} a_{d-1,\vec{\mu}} - a_{d-1,\vec{\bar{\mu}}} &= \frac{a_d}{N}\left(m_{\vec{\mu}}\frac{N - a_d m_{\vec{\mu}}^d}{p} - m_{\vec{\bar{\mu}}}\frac{N - a_d m_{\vec{\bar{\mu}}}^d}{p}\right) = \\ &= \frac{a_d}{N}\left(\frac{N\left(m_{\vec{\mu}} - m_{\vec{\bar{\mu}}}\right)}{p} - \frac{a_d\left(m_{\vec{\mu}}^{d+1} - m_{\vec{\bar{\mu}}}^{d+1}\right)}{p}\right). \end{aligned}$$

Pro $k \neq i$ platí $p_k \mid x_{i,\mu_j}$ pro libovolné μ_j, tudíž

$$m_{\vec{\mu}} - m_{\vec{\bar{\mu}}} = x_{\vec{\mu}} - x_{\vec{\bar{\mu}}} = x_{i,\mu_i} - x_{i,\bar{\mu}_i} \equiv 0 \pmod{\frac{p}{p_i}}. \tag{3.5}$$

To aplikujeme do vztahu výše a s využitím vzorce pro výpočet rozdílu dvou mocnin dostáváme

$$\frac{a_d}{N}\left(\frac{N\left(m_{\vec{\mu}} - m_{\vec{\bar{\mu}}}\right)}{p} - \frac{a_d\left(m_{\vec{\mu}}^{d+1} - m_{\vec{\bar{\mu}}}^{d+1}\right)}{p}\right) \equiv \frac{a_d}{N}\cdot\frac{-a_d\left(m_{\vec{\mu}}^{d+1} - m_{\vec{\bar{\mu}}}^{d+1}\right)}{p} \equiv$$

$$\equiv \frac{a_d}{N}\cdot\frac{-a_d(m_{\vec{\mu}} - m_{\vec{\bar{\mu}}})\sum_{j=1}^{d+1} m_{\vec{\mu}}^{d-j} m_{\vec{\bar{\mu}}}^{j-1}}{p} \pmod{\frac{p}{p_i}}.$$

Dalším využitím vztahu (3.5) dostáváme

$$\sum_{j=1}^{d+1} m_{\vec{\mu}}^{d-j} m_{\vec{\bar{\mu}}}^{j-1} \equiv (d+1) m_{\vec{\mu}}^{d-1} \pmod{\frac{p}{p_i}},$$

proto také s aplikací (3.1) platí

$$\begin{aligned} \frac{a_d}{N}\cdot\frac{-a_d(m_{\vec{\mu}} - m_{\vec{\bar{\mu}}})\sum_{j=1}^{d+1} m_{\vec{\mu}}^{d-j} m_{\vec{\bar{\mu}}}^{j-1}}{p} &\equiv \frac{a_d}{N}\cdot\frac{-a_d(x_{i,\mu_i} - x_{i,\bar{\mu}_i})(d+1)m_{\vec{\mu}}^d}{p} \equiv \\ &\equiv \frac{a_d(d+1)(x_{i,\bar{\mu}_i} - x_{i,\mu_i})}{p} \pmod{\frac{p}{p_i}}. \end{aligned}$$

Hodnota p sice není invertibilní modulo $\frac{p}{p_i}$, ale člen $x_{i,\bar{\mu}_i} - x_{i,\mu_i}$ lze $\frac{p}{p_i}$ vydělit a p_i již inverz modulo $\frac{p}{p_i}$ má. Díky tomu umíme vyjádřit

$$\begin{aligned} a_{d-1,\vec{\mu}} - a_{d-1,\vec{\bar{\mu}}} &\equiv \frac{a_d d}{p_i}\frac{x_{i,\bar{\mu}_i} - x_{i,\mu_i}}{\frac{p}{p_i}} \pmod{\frac{p}{p_i}} \\ a_{d-1,\vec{\nu}} - a_{d-1,\vec{\bar{\nu}}} &\equiv \frac{a_d d}{p_i}\frac{x_{i,\bar{\nu}_i} - x_{i,\nu_i}}{\frac{p}{p_i}} \pmod{\frac{p}{p_i}}, \end{aligned}$$

tedy platí

$$a_{d-1,\vec{\mu}} - a_{d-1,\vec{\bar{\mu}}} \equiv a_{d-1,\vec{\nu}} - a_{d-1,\vec{\bar{\nu}}} \pmod{\frac{p}{p_i}},$$

protože $\mu_i = \nu_i$ a $\bar{\mu}_i = \bar{\nu}_i$.

Zbývá nám vyřešit situaci modulo p_i. Z vlastnosti $\frac{p}{p_i} \mid x_{i,\mu_j}$ pro libovolné μ_j plyne

$$m_{\vec{\mu}} - m_{\vec{\nu}} = \sum_{j=1}^{d} x_{j,\mu_j} - \sum_{j=1}^{d} x_{j,\nu_j} \equiv x_{i,\mu_i} - x_{i,\nu_i} \equiv 0 \pmod{p_i},$$

jelikož $\mu_i = \nu_i$. Proto máme

$$a_{d-1,\vec{\mu}} - a_{d-1,\vec{\nu}} \equiv \frac{a_d m_{\vec{\mu}}}{N} \frac{N - a_d m_{\vec{\mu}}^d}{p} - \frac{a_d m_{\vec{\nu}}}{N} \frac{N - a_d m_{\vec{\nu}}^d}{p} \equiv 0 \pmod{p_i}.$$

Analogicky, $a_{d-1,\vec{\mu}} \equiv a_{d-1,\vec{\nu}} \pmod{p_i}$. Aplikací Čínské zbytkové věty dostáváme tvrzení. □

Věta 16. *Buď $1 \leq i \leq l$ a $1 \leq \mu_j \leq d$. Existuje dl hodnot e_{i,μ_j} takových, že libovolný z d^l koeficientů $a_{d-1,\vec{\mu}}$ lze vyjádřit jako lineární kombinaci těchto dl hodnot*

$$a_{d-1,\vec{\mu}} = \sum_{i=1}^{l} e_{i,\mu_i}.$$

Takto vyjádřené $a_{d-1,\vec{\mu}}$ splňuje

$$a_{d-1,\vec{\mu}} m_{\vec{\mu}}^{d-1} \equiv \frac{N - a_d m_{\vec{\mu}}^d}{p} \pmod{p} \tag{3.6}$$

a lze ho proto použít v base-$(m_{\vec{\mu}}, p)$ rozvoji N.

Důkaz. Vztah (3.6) je kongruencí modulo p, proto stačí najít hodnoty e_{i,μ_j} také pouze modulo p (tato redukce však není nutná). Navíc hodnoty e_{i,μ_j} nejsou určeny jednoznačně, protože je možné odečíst libovolnou konstantu od hodnot e_{i,μ_j} pro všechna μ_j a přičíst stejnou konstantu ke všem hodnotám e_{i',μ_j}. Podle lemmatu 15 rozdíl $a_{d-1,\vec{\mu}} - a_{d-1,\vec{\bar{\mu}}}$ modulo p závisí pouze na souřadnicích μ_i a $\bar{\mu}_i$, proto můžeme položit například

$e_{1,\mu_j} \equiv a_{d-1,(\mu_j,1,\ldots,1)} \bmod p$ pro $1 \leq \mu_j \leq d$,

$e_{i,1} = 0$ pro $1 < i \leq l$ a

$e_{i,\mu_j} = a_{d-1,(1,\ldots,1,\mu_j,1,\ldots,1)} - a_{d-1,(1,\ldots,1)} \bmod p$ pro $1 < i \leq l$ a $1 < \mu_j \leq d$, kde μ_j je na i-té souřadnici.

□

Poznámka. Uvědomme si, že dokázaná věta není návodem, jak najít hodnoty $a_{d-1,\vec{\mu}}$, prakticky je umíme dopočítat z base-$(m_{\vec{\mu}}, p)$ rozvoje N. Přínosem věty je zjednodušení, jak z dl dopočítaných hodnot vyjádřit zbývající do celé dávky.

Nyní se konečně dostáváme k odhadu koeficientu a_{d-2}. Ukažme nejdříve, že můžeme k polynomu $f(x)$ přičítat násobky $(px - m)x^{d-2}$, aniž bychom porušili vlastnosti, které od polynomu požadujeme. Pro libovolné $k \in \mathbb{Z}$ označme $\bar{f} = f(x) + k(px - m)x^{d-2}$ a $\bar{F}$ homogenizaci $\bar{f}$. Pak společný kořen $\tilde{m} = \frac{m}{p}$ je také kořenem $\bar{f}$ modulo N, protože

$$\bar{f}(\tilde{m}) = f\left(\frac{m}{p}\right) + k \cdot \left(p\frac{m}{p} - m\right)\left(\frac{m}{p}\right)^{d-2} \equiv 0 + k \cdot 0 \cdot \left(\frac{m}{p}\right)^{d-2} \equiv 0 \pmod{N}.$$

A také base-(m,p) rozvoj čísla N zůstane zachován, jelikož

$$\bar{F}(m,p) = F(m,p) + kp^{d-1}\left(p\frac{m}{p} - m\right)\left(\frac{m}{p}\right)^{d-2} = F(m,p) = N.$$

Mohu tedy zmenšovat koeficient $a_{d-2,\vec{\mu}}$ o libovolný celočíselný násobek $m_{\vec{\mu}}$ (na úkor zvětšení koeficientu $a_{d-1,\vec{\mu}}$ o stejný násobek p, ale předpokládáme $p \ll m_{\vec{\mu}}$). Aproximujme libovolné $m_{\vec{\mu}} \approx \bar{m}$, protože $m_{\vec{\mu}}$ se od $\bar{m}$ bude lišit maximálně o $l \cdot p$. Z věty 13 dostáváme, že pak je zhruba $a_{d-2,\vec{\mu}} \approx \bar{m}$. Čím blíže bude podíl $\frac{a_{d-2,\vec{\mu}}}{\bar{m}}$ nějakému celému číslu, tím více bude možné koeficient $a_{d-1,\vec{\mu}}$ pomocí celočíselných násobků $\bar{m}$ zmenšit. V následujícím tvrzení užíváme zavedené značení.

Tvrzení 17. *Pro $1 \le i \le l$ a $1 \le \mu_j \le d$ položme*

$$h_0 = \frac{N - a_d\bar{m}^d}{p^2\bar{m}^{d-1}} \quad a \quad h_{i,\mu_j} = -\frac{a_d dx_{i,\mu_j}}{p^2} - \frac{e_{i,\mu_j}}{p}.$$

Pak lze aproximovat

$$\frac{a_{d-2,\vec{\mu}}}{\bar{m}} \approx h_0 + \sum_{i=1}^{l} h_{i,\mu_i}.$$

Důkaz. Protože uvažujeme $p \ll m_0$ lze pomocí rekurze z věty 13 aproximovat

$$\frac{a_{d-2,\vec{\mu}}}{\bar{m}} \approx \frac{r_{d-2,\vec{\mu}}}{\bar{m}^{d-1}} = \frac{N - a_d m_{\vec{\mu}}^d - a_{d-1,\vec{\mu}} m_{\vec{\mu}}^{d-1} p}{p^2\bar{m}^{d-1}}.$$

Zřejmě je $m_{\vec{\mu}}^n = (\bar{m} + (m_{\vec{\mu}} - \bar{m}))^n$. Pro $n \in \{d, d-1\}$ nahradíme $m_{\vec{\mu}}^n$ binomickým rozvojem až po stupeň $d-1$ u $\bar{m}$. Dostáváme

$$\begin{aligned}\frac{N - a_d m_{\vec{\mu}}^d - a_{d-1,\vec{\mu}} m_{\vec{\mu}}^{d-1} p}{p^2\bar{m}^{d-1}} &\approx \frac{N - a_d(\bar{m}^d + d\bar{m}^{d-1}(m_{\vec{\mu}} - \bar{m})) - a_{d-1,\vec{\mu}}\bar{m}^{d-1}p}{p^2\bar{m}^{d-1}} = \\ &= \frac{N - a_d\bar{m}^d}{p^2\bar{m}^{d-1}} + \frac{-a_d d(m_{\vec{\mu}} - \bar{m})}{p^2} - \frac{a_{d-1,\vec{\mu}}}{p} = \\ &= h_0 + \sum_{i=1}^{l} h_{i,\mu_i}.\end{aligned}$$

□

I v tomto případě jsme tudíž ukázali, že celou dávku d^l hodnot $\frac{a_{d-2,\vec{\mu}}}{\bar{m}}$ lze vyjádřit jako lineární kombinaci dl hodnot x_{i,μ_j} a e_{i,μ_j}. Pro libovolnou dávku d^l polynomů definovaných pomocí trojice $(a_d, p, m_{\vec{\mu}})$ umíme najednou efektivně zkontrolovat, zda koeficienty u tří nejvyšších mocnin budou dostatečně malé.

Právě výběr těch kandidátů, pro něž jsou hodnoty $\frac{a_{d-2,\vec{\mu}}}{\bar{m}}$ blízko celému číslu, je ovšem výpočetně nejnáročnějším krokem. Kleinjung navrhuje efektivní postup, který jsme také implementovali. Zvolme $l' = \lfloor \frac{l}{2} \rfloor$ a místo $h_0 + \sum_{i=1}^{l} h_{i,\mu_i}$ spočítejme dvě množiny o d^l hodnotách

$$H_{1,\vec{\mu}} = \left\{ h_0 + \sum_{i=1}^{l'} h_{i,\mu_i} - \left\lfloor h_0 + \sum_{i=1}^{l'} h_{i,\mu_i} \right\rfloor \right\} \text{ a}$$

$$H_{2,\vec{\mu}} = \left\{ -\left(\sum_{i=l'+1}^{l} h_{i,\mu_i} - \left\lfloor \sum_{i=l'+1}^{l} h_{i,\mu_i} \right\rfloor \right) \right\}.$$

Množiny seřadíme podle velikosti a najdeme největší n takové, že n-tý prvek $H_{1,\vec{\mu}}$ leží v ϵ-okolí n-tého prvku z $H_{2,\vec{\mu}}$ pro $\epsilon = \frac{a_{d-2,max}}{\bar{m}}$. Za kandidáty označíme prvních n hodnot $\vec{\mu}$.

Výběr vyhovujících polynomů z d^l kandidátů tedy spočívá ve vygenerování množin $H_{1,\vec{\mu}}$ a $H_{2,\vec{\mu}}$ v čase $\mathcal{O}(dl)$, seřazení lze udělat efektivně v průměrném čase $\mathcal{O}\left(d^{\frac{l}{2}} \log d^{\frac{l}{2}}\right)$ a najití hranice dobrých kandidátů je v seřazené množině otázkou zhruba $\mathcal{O}(l \log d)$ kroků. V průměru tedy zkontrolujeme jednoho polynomiálního kandidáta v čase $\mathcal{O}\left(ld^{-\frac{l}{2}} \log d\right)$. Proto chceme volit l co největší. Příliš vysoké l však na druhou stranu může mít za následek nevygenerování žádného kandidáta, protože nenalezneme dostatečně malé kandidáty. V průběhu implementace jsme empiricky došli k závěru, že je vhodné volit $l \in \{4, 5, 6, 7\}$ v závislosti na mezích koeficientů a mezi pro prvočísla B_p. Podrobnější zkoumání ale nebylo možné vzhledem k obrovským časovým nárokům na prosívací fázi, které algoritmus číselného síta má pro velká čísla.

Ve zkratce probíhá generování polynomů Kleinjungovým algoritmem takto: Zvolíme vhodný stupeň d vzhledem k velikosti rozkládaného čísla (podrobněji řešíme v kap. 4.1.2) a iterujeme přes malé vedoucí koeficienty. Pro každé a_d procházíme všechna p (součiny l malých prvočísel), pro které má kongruence $\frac{N}{a_d} \equiv x^d \bmod p$ právě d^l řešení, které značíme $m_{\vec{\mu}}$, a požadujeme, aby tato zajistila malé koeficienty $a_{d-1,\vec{\mu}}$ (viz věta 13). Popsaným dávkovým zpracováním najdeme a vybereme jen ta $\vec{\mu}$, pro která je také koeficient $a_{d-2,\vec{\mu}}$ dostatečně malý. Na závěr dopočítáme zbylé koeficienty z base-$(m_{\vec{\mu}},p)$ rozkladu. Celý postup Kleinjungova algoritmu shrnujeme v pseudokódu jako algoritmus 3.2 se zavedeným značením.

Poznámka. Thorsten Kleinjung v roce 2008 publikoval ještě další verzi algoritmu pro generování polynomů, kde navrhuje efektivní postup jak velikost koeficientu a_{d-2} při generování omezit. Tato verze algoritmu je popsána v práci [24], my jsme se jí více nezabývali.

Algoritmus 3.2 Kleinjungův algoritmus

input : rozkládané číslo $N, d = \deg(f)$, meze koeficientů $a_{d,max}, a_{d-1,max}, a_{d-2,max}$, mez l pro počet prvočinitelů čísla p, mez pro tyto prvočinitele B_p

output : polynom $f(x)$ sdílející kořen modulo N s polynomem $g(x)$ mající první tři koeficienty menší než zvolené meze

1 : $P \leftarrow \{r \equiv 1 \bmod d \mid r \text{ prvočíslo a } r < B_p\}$

2 : $a_d \leftarrow 0$

3 : **while** $(a_d < a_{d,max})$

4 : $a_d \leftarrow a_d + 1$

5 : $Q \leftarrow \{r \in P \mid \dfrac{N}{a_d} \not\equiv 0 \bmod r \text{ je } d\text{-tá mocnina modulo } r\}$

6 : $m_0 \leftarrow \sqrt[d]{\dfrac{N}{a_d}}$

7 : **for** $(P' \subset Q, |P'| = l, p = \prod_{r \in P'} r \leq a_{d-1,max})$

8 : **for** $(i = 1, \ldots, l)$

9 : **for** $(\mu_j = 1, \ldots, d)$

10 : spočítej příslušná $x_{i,\mu_j}, m_{i,\mu_j}, e_{i,\mu_j}$ // viz věta 14 a 16

11 : spočítej příslušná h_0 a h_{i,μ_j} // viz věta 17

12 : $\epsilon \leftarrow \dfrac{a_{d-2,max}}{\bar{m}}$

13 : najdi vektory $\vec{\mu}$, pro které $h_0 + \sum_{i=1}^{l} h_{i,\mu_i} \bmod \mathbb{Z} \in [-\epsilon, \epsilon]$

14 : **return** polynomy příslušné $(a_d, p, m_{\vec{\mu}})$

// z base-$(m_{\vec{\mu}}, p)$ rozvoje N s vedoucím koeficientem a_d

Kapitola 4

Hodnocení polynomů

Způsobem popsaným v předchozí kapitole získáme mnoho polynomů, které nám všechny teoreticky umožňují rozložit dané N. Nicméně, jak jsme již výše zmiňovali, volba polynomu zásadně ovlivňuje dobu běhu celého číselného síta. Proto budeme chtít generované polynomy nějakým způsobem hodnotit a pro další fáze číselného síta vybrat nejvhodnější polynom. Jak jsme popsali v kapitole 2, hledáme polynom, pomocí kterého získáme velké množství hladkých relací, tedy polynom s velkou výtěžností.

Jedním možným způsobem, jak zvolit polynom s největší výtěžností, je spustit tzv. *zkušební prosívání*, kdy na malém vzorku prosívací oblasti prosíváme pomocí každého polynomu a zvolíme ten, který dá nejvíce hladkých relací. Tato metoda je však vhodná jen pro malé množství polynomů; zkušební prosívání s desítkami či stovkami polynomů by bylo neúměrnou zátěží pro běh celého algoritmu. Proto budeme nejdříve hledat jinou, efektivnější metodu, která by vygenerované polynomy seřadila podle očekávané výtěžnosti, a buď rovnou vybereme nejlepší polynom, nebo s několika nejlepšími provedeme právě popsané zkušební prosívání.

V následující podkapitole nejprve popíšeme vlastnosti, které ovlivňují výtěžnost, a pro každou vlastnost odvodíme metodu jak polynomy na základě této vlastnosti efektivně hodnotit. Vyjdeme z výsledků, které publikoval B. A. Murphy ve své dizertaci [21]. V oblasti jeho zájmu však byly pouze monické polynomy generované jednoduchou base-m metodou, ukážeme proto rozšíření i pro nemonické polynomy generované base-(m, p) metodou, tudíž polynomy získané z Kleinjungova algoritmu. V části 4.2 vysvětlíme Murphyho model, jak hodnocení vlastností zkombinovat. Výpočetní náročnost kombinované metody je ovšem vysoká, není tudíž praktické hodnotit tímto způsobem všechny vygenerované polynomy. V závěru 4.3 proto popíšeme, jak nejprve vygenerované polynomy optimalizujeme vzhledem k definovaným vlastnostem a efektivními metodami hodnocení identifikujeme a do dalšího zpracování nepropustíme ty, které nelze dostatečně dobře optimalizovat. Na závěr kombinovaně hodnotíme jen ty polynomy, které úspěšně prošly celým procesem optimalizace. Všechny uvedené aspekty optimalizace a hodnocení polynomů jsme také prakticky využili při implementaci algoritmu.

4.1 Vlastnosti ovlivňující výtěžnost

V celé kapitole budeme pracovat s polynomem $f \in \mathbb{Z}[x]$ stupně $d \geq 4$ a jeho homogenizací $F(x,y) = y^d \cdot f\left(\frac{x}{y}\right) \in \mathbb{Z}[x,y]$. Jak jsme diskutovali v kapitole 2, výtěžnost polynomu f na nějaké oblasti $S \subseteq \mathbb{Z}$ úzce souvisí s prvočíselným rozkladem hodnoty $F(a,b)$ pro $a, b \in S$. Zaměříme se proto na studium vlastností homogenního polynomu F. Podle dosavadních výsledků ([3, 4, 12, 16, 21, 27]) ovlivňují výtěžnost polynomu f dva faktory – *velikost koeficientů* a *kořenové vlastnosti*.

4.1.1 Velikost koeficientů

Aby mnoho polynomiálních hodnot v prosívací oblasti bylo hladkých, je zřejmé, že se budeme snažit, aby tyto hodnoty byly co nejmenší. Dobrým předpokladem je omezení velikosti koeficientů polynomu.

Zvolíme vhodnou mez M a budeme požadovat $|x| \leq M$ i $|y| \leq M$, tedy $(x,y) \in \mathcal{M} = [-M, M] \times [-M, M] \subseteq \mathbb{Z} \times \mathbb{Z}$. Oblast $\mathcal{M}$ budeme dále označovat jako *prosívací oblast*. Uvažujme nyní $a > 0, b > 0$, ze vztahu $F(a,b) = b^d f\left(\frac{a}{b}\right)$ plyne následující:

- $F(a,-b) = (-1)^d F(-a,b)$;
- $F(-a,-b) = (-1)^d F(a,b)$.

Na místech, kde budeme v dalším textu pracovat s hodnotou $|F(x,y)|$, se tudíž můžeme bez újmy na obecnosti omezit na práci s oblastí $\mathcal{M}' = [-M,M] \times [1,M]$.

Dále zvolme vhodnou mez hladkosti $1 < B \leq M$ na této oblasti. Kvalitu polynomu F (přesněji dvojice polynomů F, G) budeme posuzovat dle jeho výtěžnosti, tedy úspěšnosti při hledání hladkých relací. Konkrétně chceme spočítat počet $(a,b) \in \mathcal{M}'$, pro které $|F(a,b)|$ je B-hladká hodnota (polynom G začleníme později). Z kapitoly 1.2 víme, že umíme počet všech B-hladkých hodnot na nějakém intervalu $[0,r]$ (značíme $\psi(r,B)$) zhruba odhadnout jako

$$r \cdot \rho\left(\frac{\ln r}{\ln B}\right),$$

kde $\rho\left(\frac{\ln r}{\ln B}\right)$ zastupuje pravděpodobnost, že náhodné celé číslo z intervalu $[0,r]$ je B-hladké. Pro odhad počtu dvojic (a,b) z prosívací oblasti, pro něž $|F(a,b)|$ je B-hladká hodnota, proto použijeme vzorec

$$\sum_{\substack{(a,b)\in\mathcal{M}' \\ \mathrm{NSD}(a,b)=1}} \rho\left(\frac{\ln|F(a,b)|}{\ln B}\right).$$

Navíc hledáme dvojici polynomů F, G, které chceme generovat tak, aby v prosívací fázi algoritmu číselného síta přinesly co nejvíce hladkých relací, tedy nesoudělných dvojic $(x,y) \in \mathcal{M}'$, které vedou na hladké polynomiální hodnoty jak $|F(x,y)|$, tak $|G(x,y)|$. Budeme-li se držet zavedeného předpokladu vzájemné nezávislosti

hladkosti polynomiálních hodnot $|F(x,y)|$ a $|G(x,y)|$, můžeme počet hladkých relací v prosívací oblasti $\mathcal{M}'$ aproximovat jako

$$\sum_{\substack{(a,b)\in\mathcal{M}'\\ \mathrm{NSD}(a,b)=1}} \rho\left(\frac{\ln|F(a,b)|}{\ln B}\right)\rho\left(\frac{\ln|G(a,b)|}{\ln B}\right), \tag{4.1}$$

případně nahradíme společnou mez B za B_F resp. B_G, pokud uvažujeme jinou mez hladkosti pro polynom F resp. G. Korektnost vzhledem k definici Dickmanovy funkce je zajištěna předpokladem ireducibility f i g v $\mathbb{Z}[x]$.

Pravděpodobnost, že dvojice $(a,b) \in \mathcal{M}'$ je nesoudělná, je pro dostatečně velké M rovna $\frac{6}{\pi^2}$, odvození této hodnoty lze nalézt například v knize [11]. Při využití zavedeného předpokladu vzájemné nezávislosti jevů nesoudělnosti (a,b) a hladkosti polynomiálních hodnot můžeme přímé ověřování nesoudělnosti dvojic (a,b) nahradit násobením touto pravděpodobností. Vzorec (4.1) pak počítáme jako

$$\frac{6}{\pi^2}\cdot\sum_{(a,b)\in\mathcal{M}'} \rho\left(\frac{\ln|F(a,b)|}{\ln B}\right)\rho\left(\frac{\ln|G(a,b)|}{\ln B}\right).$$

Řada autorů pro aproximaci počtu hladkých relací v prosívací oblasti $\mathcal{M}'$ využívá spojitou variantu uvedeného vzorce, pravděpodobně z výpočetních důvodů. Tuto verzi přejímáme a jako referenční vzorec pro další výpočty budeme uvažovat

$$\frac{6}{\pi^2}\cdot\iint_{\mathcal{M}'} \rho\left(\frac{\ln\left|\tilde{F}(x,y)\right|}{\ln B}\right)\rho\left(\frac{\ln\left|\tilde{G}(x,y)\right|}{\ln B}\right)\mathrm{d}x\,\mathrm{d}y, \tag{4.2}$$

kde

$$\tilde{F}(x,y) = \begin{cases} F(x,y) & \text{pokud } F(x,y) \notin (-1,1),\\ 1 & \text{jinak.}\end{cases}$$

Toto drobné omezení je nutné pro zajištění korektnosti s ohledem na definici Dickmanovy funkce jen pro nezáporné hodnoty. Neformálně řečeno, nám vznikají jakési „díry“ v (nyní spojité) prosívací oblasti kolem bodů vedoucích ke kořenům F a G. Empiricky jsme zjistili, že v případě polynomů stupně 4 zaujímá každá z těchto „děr“ plochu maximálně o velikosti $(10^{-8})^2$, v případě polynomů vyšších stupňů se jedná ještě o řádově menší číslo. Horní mez na procento takto zanedbané oblasti lze obecně vyjádřit jako $\frac{d\cdot 10^{-16}}{2(M^2+M)}$, což je pro $d = 4$ a $M = 10^5$ rovno přibližně $2\cdot 10^{-26}\%$. Chyba způsobená tímto omezením je tedy zcela zanedbatelná.

Výpočet hodnoty (4.2) je však relativně náročný a při velkém množství dvojic polynomů F a G by se stal neúměrným zatížením celého algoritmu číselného síta. Proto se nyní pokusíme o několik zjednodušení tohoto výpočtu, která sice vnesou do hodnocení polynomů nějakou chybu, nicméně budeme předpokládat, že tato chyba je zanedbatelná vzhledem ke zrychlení výpočtu, které zjednodušení přinesou. Předpoklad navíc podpíráme empirickými daty získanými v rámci experimentu, který popisujeme v kapitole 6.2.

U base-m metody pracujeme s homogenním polynomem $G(x,y) = x - my$ pro pevné $m \approx N^{\frac{1}{d}}$, v případě base-$(m,p)$ metody uvažujeme $G(x,y) = px - my$ pro pevná $m \approx N^{\frac{1}{d+1}}$ a $p \ll m$. Hodnoty (x,y) jsou v obou případech brány z prosívací oblasti $\mathcal{M}' = [-M,M]\times[1,M]$. Volbu meze M však předpokládáme (a v praxi

také volíme) o několik řádů menší než $N^{\frac{1}{d}}$ resp. $N^{\frac{1}{d+1}}$, abychom omezili dobu prosívání. Koeficient m bude proto v $G(x,y)$ dominantní pro libovolnou volbu (x,y) z prosívací oblasti. Lze tudíž předpokládat, že hodnota $\ln|G(x,y)|$ se přes prosívací oblast nebude příliš měnit, a můžeme vynechat i výpočet $\rho\left(\frac{\ln\left|\tilde{G}(x,y)\right|}{\ln B}\right)$, aniž bychom porovnání zatížili příliš velkou chybou.

Výpočetní náročnost funkce ρ je však netriviální a pro porovnávání velkého množství polynomů F je tento odhad ještě třeba zjednodušit. Dickmanova funkce $\rho(u)$ je klesající. Proto pro $(a,b),(a',b') \in \mathcal{M}'$ je $|F(a,b)| \geq |F(a',b')|$ právě když $\rho\left(\frac{\ln\left|\tilde{F}(a,b)\right|}{\ln B}\right) \leq \rho\left(\frac{\ln\left|\tilde{F}(a',b')\right|}{\ln B}\right)$. To nabízí heuristický argument, že pro maximalizaci $\iint\limits_{\mathcal{M}'} \rho\left(\frac{\ln\left|\tilde{F}(x,y)\right|}{\ln B}\right) \mathrm{d}x\,\mathrm{d}y$ je možné použít minimalizaci $\iint\limits_{\mathcal{M}'} |F(x,y)|\,\mathrm{d}x\,\mathrm{d}y$. Pro nalezení vhodných poynomů se tedy jeví jako možnost vybírat ty, pro které je

$$\iint\limits_{\mathcal{M}'} |F(x,y)|\,\mathrm{d}x\,\mathrm{d}y$$

co nejmenší. Absolutní hodnota však není spojitá funkce, z výpočetního hlediska je proto výhodnější počítat L^2-normu

$$\sqrt{\iint\limits_{\mathcal{M}'} F^2(x,y)\,\mathrm{d}x\,\mathrm{d}y}. \tag{4.3}$$

Jelikož pracujeme s obrovskými hodnotami, je pohodlné pro účely porovnání uvažovat logaritmus této hodnoty, volitelně je pak možné vynechat odmocninu

$$\ln\left(\iint\limits_{\mathcal{M}'} F^2(x,y)\,\mathrm{d}x\,\mathrm{d}y\right).$$

Neumíme přinést rigorózní argumenty, že je popsané zjednodušení oprávněné, nicméně praxe ukazuje, že záměna přesného výpočtu pomocí Dickmanovy funkce (4.2) za odhad pomocí L^2-normy (4.3) je možná s uspokojivými výsledky. Za účelem odhadu chyby vnesené tímto zjednodušením jsme provedli experiment, který navíc ukázal, že je možné i další výrazné zjednodušení výpočtu se srovnatelnými výsledky při porovnání polynomů. Podrobněji tento experiment popisujeme v kapitole 6.2.

Velikost zkosených koeficientů

Dosud popsaný přístup se hodí pro polynom f získaný z base-m metody, který má všechny koeficienty řádově stejně velké. My budeme dále pracovat s polynomy z base-(m,p) metody v Kleinjungově algoritmu, proto je třeba uvedený postup mírně změnit. Mají-li hodnoty $F(x,y)$ být co nejmenší, musíme koeficienty u monočlenů s vyšším stupněm zmenšit, což se promítne do výrazného zvětšení koeficientů u monočlenů nižšího stupně. Pokud bychom polynom f zapsali s odřádkováním po každém monočlenu, vizuálně dochází ke zkosení tvaru polynomu (příklad viz obr. 4.1), proto budeme tímto způsobem upravené polynomy dále označovat jako *zkosené* (anglicky *skewed*). Zkosené polynomy nás budou zajímat, protože jsou výstupem Kleinjungova algoritmu (viz věta 13).

$$
\begin{aligned}
f_1(x) = {} & 208755831981502781461 \cdot x^5 \\
& + 965399696680667191836 \cdot x^4 \\
& + 960996358064245284662 \cdot x^3 \\
& + 260498118977004930560 \cdot x^2 \\
& + 775890222450124624696 \cdot x \\
& + 330920632919740624389
\end{aligned}
$$

Polynom generovaný base-m metodou.

$$
\begin{aligned}
f_2(x) = {} & 5340 \cdot x^5 \\
& - 181518938024032 \cdot x^4 \\
& + 1814988218032924459794632 \cdot x^3 \\
& - 350044386037941859685977462212 \cdot x^2 \\
& + 345871306420787297428189960190 0193 \cdot x \\
& - 352276330613888573945325126209507594616
\end{aligned}
$$

Polynom generovaný base-(m, p) metodou.

Obrázek 4.1: Příklad nezkoseného a zkoseného polynomu pro N se 124 decimálními ciframi.

Zkosení polynomu f (a v důsledku toho i homogenního F) vynucuje také redefinici čtvercové prosívací oblasti. Protáhnutím oblasti ve směru x-osy a zúžením ve směru y-osy dosáhneme přirozeného vyrovnání váhy zkoseného polynomu. Formálně, zvolíme vhodně (tj. aby zůstala zachována stejná plocha jako u $\mathcal{M}$) meze A a B a budeme pracovat s prosívací oblastí $\mathcal{S} = [-A, A] \times [-B,B] \subseteq \mathbb{Z} \times \mathbb{Z}$. Takovou prosívací oblast budeme dále označovat jako *zkreslená oblast* (anglicky *skewed area*), protože dříve užívaný pojem „zkosená oblast“ je z geometrického hlediska zavádějící. Podobně jako v předchozím případě stačí u homogenizovaných polynomů pracovat pouze s polovinou oblasti v kladné polorovině, tedy oblastí $\mathcal{S}' = [-A,A] \times [1,B] \subseteq \mathbb{Z} \times \mathbb{Z}$. Pracovat budeme také s parametrem *zkreslení*, neboli poměrem $s = \frac{A}{B}$. Vliv parametru zkreslení s na tvar prosívací oblasti, velikost koeficientů polynomu $F(x, y)$ a hodnocení polynomu L^2-normou shrnuje následující lemma.

Lemma 18. *Buď s parametr zkreslení, $\mathcal{M} = [-M, M] \times [-M, M]$ původní čtvercová prosívací oblast a $F(x, y) = \sum_{i=0}^{d} c_i x^i y^{d-i}$ homogenní polynom. Pak*

(i) zkreslená prosívací oblast je $\mathcal{S} = [-M\sqrt{s}, M\sqrt{s}] \times [-\frac{M}{\sqrt{s}}, \frac{M}{\sqrt{s}}]$;

(ii) pro $(a, b) \in \mathcal{S}$ libovolné je $|F(a, b)| \leq M^d \sum_{i=0}^{d} \left|c_i s^{i-\frac{d}{2}}\right|$;

(iii) pro L^2-normu přes zkreslenou oblast platí

$$\frac{1}{2}\ln\left(\iint\limits_{\mathcal{S}} F^2(x,y)\,\mathrm{d}x\,\mathrm{d}y\right) = \frac{1}{2}\ln\left(\frac{M^{2d+2}}{s^d}\iint\limits_{-1}^{1} F^2(xs,y)\,\mathrm{d}x\,\mathrm{d}y\right).$$

Důkaz.

ad (i) Z definice parametru zkreslení a z požadavku na zachování stejné plochy prosívací oblasti sestavíme jednoduchou soustavu rovnic

$$\frac{A}{B} = s$$
$$A \cdot B = M^2$$

Vyřešením této soustavy dostaneme volbu pro meze $A = M\sqrt{s}$ a $B = \frac{M}{\sqrt{s}}$, tudíž $\mathcal{S} = [-M\sqrt{s}, M\sqrt{s}] \times [-\frac{M}{\sqrt{s}}, \frac{M}{\sqrt{s}}]$.

ad (ii) Využitím trojúhelníkové nerovnosti a dosazením mezních hodnot z prosívací oblasti, tedy $a = M\sqrt{s}$ a $b = \frac{M}{\sqrt{s}}$, získáme horní odhad pro $|F(a,b)|$

$$|F(a,b)| \le \sum_{i=0}^{d}\left|c_i a^i b^{d-i}\right| \le \sum_{i=0}^{d}\left|c_i (M\sqrt{s})^i \left(\frac{M}{\sqrt{s}}\right)^{d-i}\right| = M^d \cdot \sum_{i=0}^{d}\left|c_i s^{i-\frac{d}{2}}\right|.$$

ad (iii) Dosazením výše odvozených mezí prosívací oblasti $\mathcal{S}$ a aplikací věty o substituci pro určité integrály dostáváme vztah pro L^2-normu

$$\frac{1}{2}\ln\left(\iint\limits_{\mathcal{S}} F^2(x,y)\,\mathrm{d}x\,\mathrm{d}y\right) = \frac{1}{2}\ln\left(\int\limits_{-\frac{M}{\sqrt{s}}}^{\frac{M}{\sqrt{s}}}\int\limits_{-M\sqrt{s}}^{M\sqrt{s}} F^2(x,y)\,\mathrm{d}x\,\mathrm{d}y\right) =$$

$$= \frac{1}{2}\ln\left(\int\limits_{-1}^{1}\int\limits_{-1}^{1} F^2\left(xM\sqrt{s}, y\frac{M}{\sqrt{s}}\right) M^2\,\mathrm{d}x\,\mathrm{d}y\right) =$$

$$= \frac{1}{2}\ln\left(\iint\limits_{-1}^{1}\left(\sum_{i=0}^{d} c_i\left(xM\sqrt{s}\right)^i\left(y\frac{M}{\sqrt{s}}\right)^{d-i}\right)^2 M^2\,\mathrm{d}x\,\mathrm{d}y\right) =$$

$$= \frac{1}{2}\ln\left(\iint\limits_{-1}^{1}\left(\left(\frac{M}{\sqrt{s}}\right)^d\sum_{i=0}^{d} c_i x^i\left(\sqrt{s}\right)^{2i} y^{d-i}\right)^2 M^2\,\mathrm{d}x\,\mathrm{d}y\right) =$$

$$= \frac{1}{2}\ln\left(\frac{M^{2d+2}}{s^d}\iint\limits_{-1}^{1} F^2(xs,y)\,\mathrm{d}x\,\mathrm{d}y\right).$$

□

Ze vztahu v bodě (ii) vyplývá, že i-tý monom polynomu $F(x,y)$ je shora omezen hodnotou $M^d c_i s^{\left(i-\frac{d}{2}\right)}$. To implikuje možnost stanovovat různé meze pro parametr zkreslení s tak, aby vhodně ovlivňoval některé koeficienty (jak ukazuje Kleinjung [16] či Bai [2]). Z rovnosti v bodě (iii) vyplývá, že pro účely porovnání polynomů $F(x,y)$ L^2-normou není nutné uvažovat velikost prosívací oblasti, postačuje zohlednit parametr zkreslení. V dalším textu budeme pro takovou referenční „jednotkovou“ zkreslenou prosívací oblast používat označení $\mathcal{S}'' = [-\sqrt{s}, \sqrt{s}] \times [-\frac{1}{\sqrt{s}}, \frac{1}{\sqrt{s}}]$.

Geometrický pohled na volbu koeficientů

Jiný úhel pohledu na optimální volbu koeficientů polynomu f přináší M. Yang a kol. v článku [27]. Minimalizaci polynomiálních hodnot $F(x, y)$, a tudíž větší výtěžnost polynomu f, se snaží zajistit přiblížením co největšího počtu poměrů $\frac{x}{y}$ k reálným kořenům polynomu f. Toho lze dosáhnout, bude-li graf funkce f co nejvíce pozvolný a blízko osy x. Studiem vlastností grafů polynomických funkcí dospěli autoři k těmto požadavkům na koeficienty polynomu $f(x) = \sum_{i=0}^{d} a_i x^i$:

(1) koeficient a_{d-2} má být záporný;

(2) koeficienty a_{d-1} a a_{d-3} mají mít opačná znaménka.

Vedoucí koeficient a_d by měl zůstat co nejmenší a výrazně lepších výsledků dosáhneme, pokud budeme potenciální hodnoty a_d iterovat v malých krocích, přestože se proto musíme omezit na menší interval. Příklad vlivu uvedených doporučení demonstruje obrázek 4.2.

4.1.2 Volba stupně nelineárního polynomu

Optimální volba stupně d nelineárního polynomu f hraje nezanedbatelnou roli při generování polynomů (jelikož omezuje maximální velikost koeficientů) a nepřímo tak ovlivňuje dobu běhu celého algoritmu číselného síta. Všechny nám známé zdroje pouze citují výsledky původního článku Buhlera, Lenstry a Pomerance [6] a přebírají asymptotický odhad pro dobu běhu algoritmu číselného síta

$$\exp\left[(1+o(1))\left(d\ln d+\sqrt{(d\ln d)^2+4\ln N^{\frac{1}{d}}\ln\ln N^{\frac{1}{d}}}\right)\right]$$

pro $N \to \infty$, jehož minimalizací získávají asymptotický odhad pro stupeň nelineárního polynomu bez dalšího vysvětlení jako

$$d \approx \sqrt[3]{\frac{\delta \ln N}{\ln\ln N}}$$

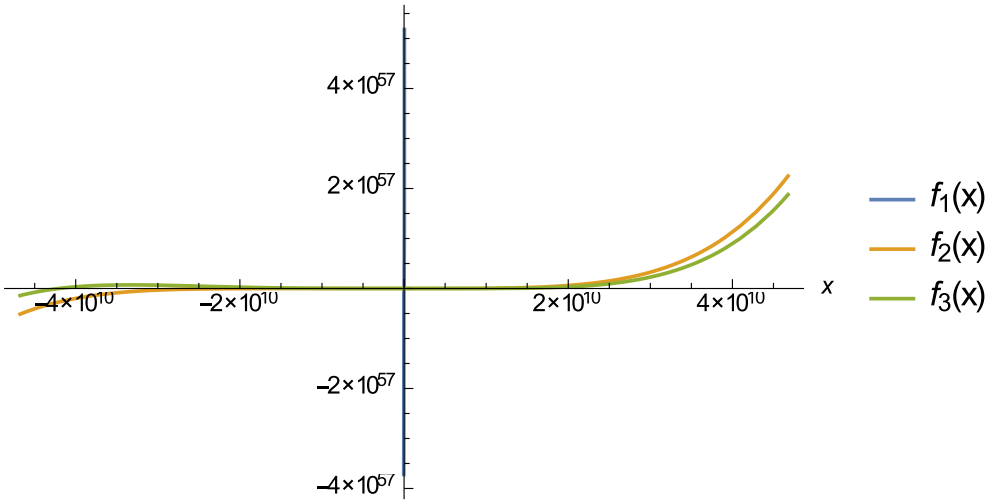
pro $\delta = 3 + o(1)$ a $N \to \infty$. Oba tyto vzorce jsou i v původním článku [6] zčásti motivovány pouze heuristicky bez rigorózních důkazů. Murphy v [21] navíc ještě uvádí, že v případě polynomů generovaných base-(m, p) metodou lze dobu běhu algoritmu číselného síta asymptoticky odhadnout lépe pomocí

$$\exp\left[(1+o(1))\left(d\ln d+\sqrt{(d\ln d)^2+4\ln N^{\frac{1}{d+1}}\ln\ln N^{\frac{1}{d+1}}}\right)\right] \tag{4.4}$$

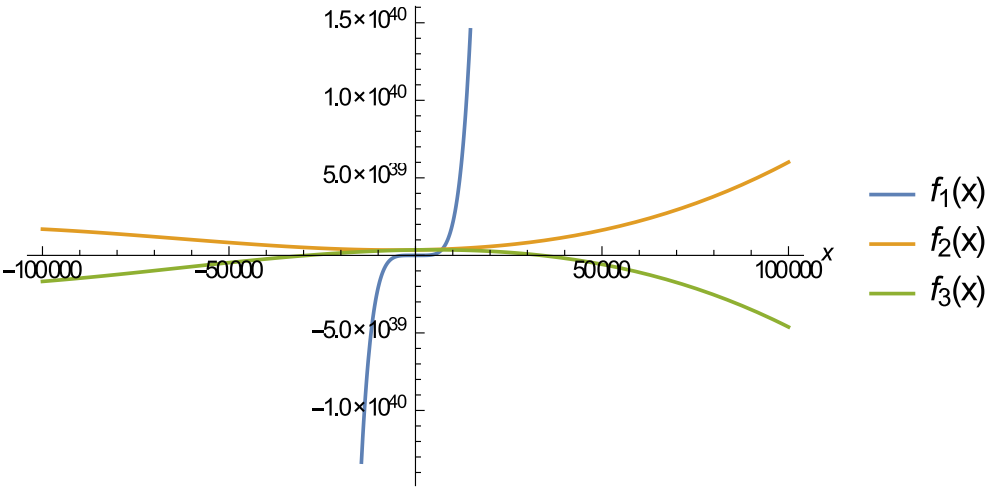
pro $N \to \infty$. Asymptoticky sice dostáváme minimalizací (4.4) stejný vzorec pro stupeň d, nicméně v praktických úlohách nefaktorizujeme nekonečná N. Proto jsme si položili otázku, zda pro konečná N neumíme asymptotickou chybu v δ vyjádřit přesněji.

V dalším textu jsme se pokusili o sepsání heuristických argumentů, které zřejmě vedly Murphyho ke vzorci (4.4). Na základě toho jsme také dopočítali optimální volbu parametru δ a provedli porovnání volby stupně mezi base-m a base-(m, p) metodou pro různé velikosti N.

$$
\begin{aligned}
f_1(x) &= 208755831981502781 46x^5 + 965399696680667191836x^4 + \\
&+ 960996358064245284662x^3 + 260498118977004930560x^2 + \\
&+ 77589022245012462469 6x + 330920632919740624389 \\
f_2(x) &= a_5x^5 + a_4x^4 + a_3x^3 + a_2x^3 + a_1x + a_0 \\
f_3(x) &= a_5x^5 + a_4x^4 - a_3x^3 - a_2x^3 + a_1x + a_0 \\
\text{pro } a_5 &= 5340 \\
a_4 &= 181518938024032 \\
a_3 &= 181498821803292445979463 2 \\
a_2 &= 350044386037941859685977462212 \\
a_1 &= 345871306420787297428189960190 0193 \\
a_0 &= 352276330613888573945325126209507594616
\end{aligned}
$$



Grafy funkcí polynomů f_1, f_2 a f_3.



Grafy funkcí polynomů f_1, f_2 a f_3 v detailu kolem počátku.

Obrázek 4.2: Vliv koeficientů na graf funkce polynomu.

Výchozím bodem pro úvahy o výpočetní složitosti algoritmu číselného síta se ukázalo být následující lemma.

Lemma 19 (Buhler, Lenstra, Pomerance [6])**.** *Mějme N, d přirozená s vlastností $N > d^{2d^2} > 1$ a meze $M \geq B \geq 2$ volené v závislosti na N a d. Pokud pro číslo $r = 2dN^{\frac{2}{d}}M^{d+1}$ platí $\frac{M^2\psi(r,B)}{r} \geq h(B)$ pro nějakou funkci h splňující $h(B) \geq 1$ a $h(B) = B^{1+o(1)}$, pak*

$$2\ln M \geq (1+o(1))\left(d\ln d + \sqrt{(d\ln d)^2 + 4\ln N^{\frac{1}{d}}\ln\ln N^{\frac{1}{d}}}\right)$$

pro $N \to \infty$ uniformně v d.

Poznámka. Důkaz lemmatu je uveden v článku [6, Lemma 10.12.], na tomto místě jej nebudeme opakovat, neboť později uvedené lemma drobně přeformulujeme (viz lemma 22) a s využitím myšlenky původního důkazu podrobně dokážeme lemma nové.

Rádi bychom lemma 19 využili pro odhad (alespoň heuristický) doby běhu algoritmu číselného síta. Jak jsme již několikrát zmínili, druhá fáze (prosívání) je výpočetně nejnáročnější a dobou běhu výrazně převažuje nad ostatními fázemi, proto dobu běhu celého algoritmu číselného síta nyní aproximujeme právě dobou nutnou pro průběh prosívání.

Prosívací fáze skončí, až když nalezneme dostatek hladkých relací. Označme B společnou horní mez hladkosti celočíselné i algebraické faktorizační báze, pak (jak jsme diskutovali v kapitole 2.2.3) hledáme alespoň $2\pi(B) + 2 \geq \pi(B)$ hladkých relací, kde $\pi(B)$ značí počet prvočísel menších než B.

Lemma 20. *Uvažujme prosívací oblast $\mathcal{M}' = [-M,M] \times [1,M] \subseteq \mathbb{Z} \times \mathbb{Z}$ a mějme polynomy f stupně d a g lineární z první fáze algoritmu číselného síta. Buď F, G homogenizace těchto polynomů a označme $r = \max_{(a,b)\in\mathcal{M}'} |F(a,b)| \cdot |G(a,b)|$.*

(1) Jsou-li f, g generovány base-m metodou, pak $r \approx 2dN^{\frac{2}{d}}M^{d+1}$.

(2) Jsou-li f, g generovány base-(m,p) metodou, pak $r \approx 2dN^{\frac{2}{d+1}}M^{d+1}$.

Důkaz.

ad (1) Mějme polynom f generovaný base-m rozkladem, tedy $f = \sum_{i=0}^{d} c_i x^i$ a lineární $g(x) = x - m$. Každý koeficient c_i lze tak shora odhadnout $m \approx N^{\frac{1}{d}}$. Pro všechny dvojice $(a,b) \in \mathcal{M}'$ platí $|a| \leq M$ i $b \leq M$. Součin homogenizací obou polynomů lze pro libovolný bod prosívací oblasti shora odhadnout jako

$$|F(a,b)| \cdot |G(a,b)| \leq \sum_{i=0}^{d} \left|mM^d\right| \cdot (M+m) \approx 2dN^{\frac{2}{d}}M^{d+1} \tag{4.5}$$

a to bez ohledu na to, zda je f monický či není.

ad (2) Polynomy generované base-(m,p) metodou (tedy náš případ Kleinjungova algoritmu), mají koeficienty menší, konkrétně $|c_i| \leq m \approx N^{\frac{1}{d+1}}$. Koeficienty u nejvyšších tří mocnin jsou dokonce mnohem menší, ale to pro jednoduchost

nyní zanedbáme. Lineární polynom je $g(x) = px - m$, kde $p \ll m$. V tom případě lze součin homogenizací v rámci prosívací oblasti odhadnout jako

$$|F(a,b)| \cdot |G(a,b)| \leq \sum_{i=0}^{d} \left|mM^d\right| \cdot (mM + m) \approx 2dN^{\frac{2}{d+1}} M^{d+1} \tag{4.6}$$

□

Prosíváme přes oblast velikosti řádově M^2 a z lemmatu 20 vidíme, že pro polynomy z base-m metody lze uvažovat $r = 2dN^{\frac{2}{d}} M^{d+1}$ jako horní mez pro velikosti hladkých hodnot v prosívací oblasti. Pak náhodný bod v této oblasti je B-hladkou relací s pravděpodobností $\frac{\psi(r,B)}{r}$ (viz lemma 2). Lze tedy předpokládat, že hodnota $\frac{M^2\psi(r,B)}{r}$ je přiměřeně dobrým odhadem počtu hladkých relací v prosívací oblasti. V našem zájmu pochopitelně je volit parametry M a B tak, aby hladkých relací v prosívací oblasti bylo více, než kolik jich potřebujeme nalézt, tedy aby platilo

$$\frac{M^2\psi(r,B)}{r} \geq \pi(B),$$

čehož lze vhodnou volbou $M > B$ jistě dosáhnout.

Zvolme $e(B) = \frac{1}{\ln\ln B}$, pak z prvočíselné věty plyne (viz také obr. 4.1.2)

$$B^{1-e(B)} \leq \frac{B}{\ln B}(1 - e(B)) \leq \pi(B) \leq \frac{B}{\ln B}(1 + e(B)) \leq B^{1+e(B)}. \tag{4.7}$$

Okomentujme první nerovnost, která nemusí být zřejmá přímo. Pro dostatečně velké B jistě platí

$$\begin{aligned} \frac{\ln B}{\ln\ln B} &\geq \ln 2 + \ln\ln B \\ B^{\frac{1}{\ln\ln B}} &\geq 2 \cdot \ln B. \end{aligned} \tag{4.8}$$

Limita $\lim\limits_{B\to\infty} \frac{\ln\ln B - 1}{\ln\ln B} = 1$, tudíž od jistého indexu musí platit

$$\frac{1}{2} \leq \frac{\ln\ln B - 1}{\ln\ln B}. \tag{4.9}$$

Složením (4.8) a (4.9) dostáváme

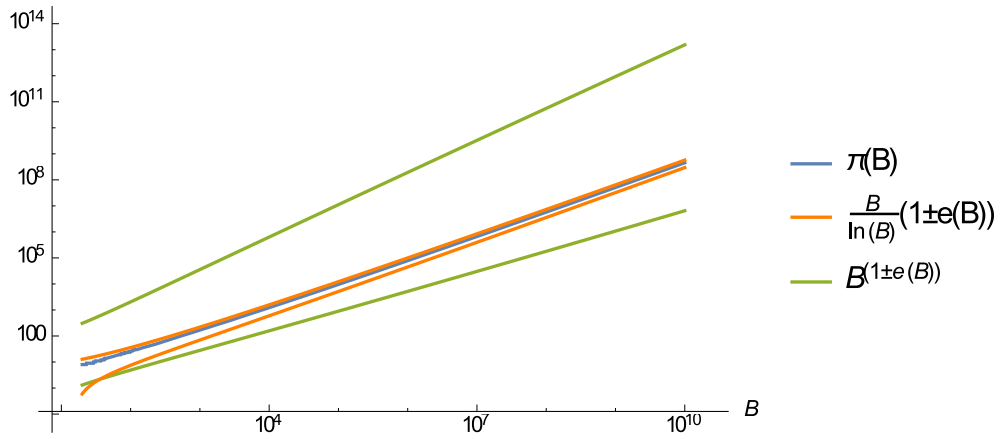
$$B^{1-\frac{1}{\ln\ln B}} = \frac{B}{B^{\frac{1}{\ln\ln B}}} \leq \frac{B}{2\ln B} \leq \frac{\ln\ln B - 1}{\ln\ln B}\frac{B}{\ln B} = \frac{B}{\ln B}\left(1 - \frac{1}{\ln\ln B}\right),$$

což jsme chtěli ukázat.

Jinými slovy z (4.7) plyne, že funkce $\pi(B)$ je $B^{1+o(1)}$ pro $B \to \infty$ a protože $\pi(B) \geq 1$ pro $B \geq 2$, splňuje $\pi(B)$ všechny předpoklady pro funkci $h(B)$ v lemmatu 19. Budeme předpokládat (později uvidíme, že v praktických úlohách je opravdu splněno), že platí $N > d^{2d^2} > 1$, tudíž můžeme použít lemma 19

$$\begin{aligned} 2\ln M &\geq (1 + o(1))\left(d\ln d + \sqrt{(d\ln d)^2 + 4\ln N^{\frac{1}{d}}\ln\ln N^{\frac{1}{d}}}\right) \\ M^2 &\geq \exp\left[(1 + o(1))\left(d\ln d + \sqrt{(d\ln d)^2 + 4\ln N^{\frac{1}{d}}\ln\ln N^{\frac{1}{d}}}\right)\right]. \end{aligned}$$

Doba běhu algoritmu číselného síta bude určitě větší než počet kandidátů na hladkou relaci, kterých je řádově M^2. Dostáváme heuristický odhad pro dobu běhu algoritmu číselného síta, kde v první fázi využíváme pro generování polynomů base-m metodu.



Obrázek 4.3: Odhad funkce $\pi(B)$ nerovností (4.7).

Hypotéza 21 (Buhler, Lenstra, Pomerance [6, hypotéza 11.4.])**.** *Pro všechna přirozená N, d taková, že $N > d^{2d^2} > 1$, lze zvolit meze $M > B > 2$, že algoritmus číselného síta s těmito parametry úspěšně skončí v čase nejvýše*

$$\exp\left[(1+o(1))\left(d\ln d + \sqrt{(d\ln d)^2 + 4\ln N^{\frac{1}{d}}\ln\ln N^{\frac{1}{d}}}\right)\right] \tag{4.10}$$

pro $N \to \infty$ uniformně v d. Navíc je tato hodnota optimální ve smyslu, že pro obecné N, pro všechna d splňující $N > d^{2d^2} > 1$ a pro všechny volby M, B, pro které je algoritmus úspěšný, je čas potřebný pro běh číselného síta alespoň (4.10).

Korektnost hypotézy 21 je však podložena argumenty zatíženými rigorózně nedokázaným předpokladem, že většina rozkládaných N je „obecná“, tj. k jejich rozložení je zapotřebí celá doba běhu algoritmu číselného síta, a speciální případy (N je mocnina prvočísla, je částečně B-hladké nebo se podaří najít neireducibilní polynom apod.), které lze rozložit v rychlejším čase, se vyskytují velmi zřídka. Další nedokázanou heuristikou je předpoklad, že pro dostatečné množství nasbíraných hladkých relací algoritmus číselného síta pravděpodobně úspěšně skončí. Na druhou stranu nám však tento předpoklad navíc zajišťuje, že v uvedeném čase jsme opravdu schopni faktorizaci provést.

Stěžejním krokem při formulaci hypotézy 21 je vhodný odhad pro mez M prosívací oblasti, aby algoritmus číselného síta skončil úspěšně. Zásadním kritériem pro závěrečné fáze algoritmu číselného síta je počet hladkých relací, který je však závislý na výtěžnosti generovaného polynomu. Pokud se tedy v případě volby jiného algoritmu pro generování polynomu zvýší výtěžnost, je intuitivní, že bude možné zmenšit M. Zmenšení pro případ base-(m,p) metody zachycuje vztah (2) v lemmatu 20. Na základě těchto úvah se domníváme, že k lepšímu odhadu doby běhu algoritmu číselného síta v případě polynomů generovaných base-(m,p) metodou vedla Murphyho nižší horní mez (4.6) v případě polynomů z base-(m,p) metody. S využitím této meze je totiž možné lemma 19 reformulovat.

Poznámka. V případě polynomů generovaných base-(m,p) metodou sice uvažujeme prosívání na zkreslené prosívací oblasti $\mathcal{S}' = [-M\sqrt{s}, M\sqrt{s}] \times [-1, \frac{M}{\sqrt{s}}]$, nicméně vzhledem k tomu, že požadujeme, aby zkreslená prosívací oblast měla stejnou plochu jako původní prosívací oblast, lze při diskuzi odvození odhadu

z hypotézy 21 použít obdobnou argumentaci i v případě zkreslené prosívací oblasti.

Lemma 22. *Mějme N, d přirozená s vlastností $N > d^{2d^2} > 1$ a meze $M \geq B \geq 2$ volené v závislosti na N a d. Pokud pro číslo $r = 2dN^{\frac{2}{d+1}}M^{d+1}$ platí nerovnost $\frac{M^2\psi(r,B)}{r} \geq h(B)$ pro nějakou funkci h splňující $h(B) \geq 1$ a $h(B) = B^{1+o(1)}$, pak*

$$2\ln M \geq (1+o(1))\left(d\ln d + \sqrt{(d\ln d)^2 + 4\ln N^{\frac{1}{d+1}}\ln\ln N^{\frac{1}{d+1}}}\right)$$

pro $N \to \infty$ uniformně v d.

Důkaz. Pro přehlednost rozdělíme průběh důkazu do následujících pěti kroků:

(1) Nejprve ukážeme, že pokud $N \to \infty$, pak také $r \to \infty$.

(2) S využitím věty 3 ukážeme $M^2 \geq L_r\left[\frac{1}{2}, \sqrt{2}+o(1)\right]$.

(3) Technickými úpravami převedeme odhad na $\frac{\ln^2 M}{\ln\ln M} \geq (1+o(1))\ln r$.

(4) Dosazením za r dostaneme $\frac{\ln^2 M}{\ln\ln M} \geq (1+o(1))\left(2\ln N^{\frac{1}{d+1}} + (d+1)\ln M\right)$.

(5) S využitím lemmatu 8 odvodíme tvrzení.

ad (1) Ze vztahu pro r v předpokladu tvrzení platí

$$\begin{aligned} r^r &= \left(2dN^{\frac{2}{d+1}}M^{d+1}\right)^{2dN^{\frac{2}{d+1}}M^{d+1}} = \\ &= N^{\frac{2d}{d+1}\cdot\left(2N^{\frac{2}{d+1}}M^{d+1}\right)} \cdot \left(2dM^{d+1}\right)^{2dN^{\frac{2}{d+1}}M^{d+1}} \end{aligned} \tag{4.11}$$

Předpoklad $N > d^{2d^2} > 1$ implikuje $d > 1$ a spolu s předpokladem $M > 1$ dávají vztahy

$$\begin{gathered} 2dN^{\frac{2}{d+1}}M^{d+1} > 2N^{\frac{2}{d+1}}M^{d+1} > 1, \\ 2dM^{d+1} > 1, \\ \frac{2d}{d+1} > 1. \end{gathered}$$

Dosazením těchto vztahů do rovnosti (4.11) získáváme $r^r > N$, ale $N \to \infty$, takže i $r^r \to \infty$ a proto také $r \to \infty$.

ad (2) Upravením předpokladu tvrzení $\frac{M^2\psi(r,B)}{r} \geq h(B)$ a využitím tvrzení věty 3 dostáváme nerovnost

$$M^2 \geq \frac{r\cdot h(B)}{\psi(r,B)} \geq L_r\left[\frac{1}{2}, \sqrt{2}+o(1)\right]. \tag{4.12}$$

Předpoklady věty 3 jsou splněny z předpokladů tohoto lemmatu a předchozího bodu.

ad (3) Využitím definice 9 lze odhad (4.12) upravit na

$$M^2 \geq \exp\left[(\sqrt{2}+o(1))(\ln r)^{\frac{1}{2}}(\ln\ln r)^{\frac{1}{2}}\right].$$

Zlogaritmováním, umocněním obou stran a drobnými úpravami dostáváme

$$\begin{aligned} 2\ln M &\geq (\sqrt{2}+o(1))(\ln r)^{\frac{1}{2}}(\ln\ln r)^{\frac{1}{2}} \\ 4\ln^2 M &\geq (\sqrt{2}+o(1))^2 \ln r \ln\ln r \\ \ln^2 M &\geq \frac{1}{4}(\sqrt{2}+o(1))^2 \ln r \ln\ln r \geq (1+o(1))\ln r \ln\ln r. \end{aligned} \tag{4.13}$$

Obecně platí, že pro $t \geq \mathrm{e}$ je $\frac{t}{\ln t}$ rostoucí funkcí v t. Označme nyní nerovnost (4.13) jako $f(M) \geq g(r)$. Bude-li $g(r) \geq \mathrm{e}$, pak $f(M) \geq g(r)$ právě tehdy, když $\frac{f(M)}{\ln(f(M))} \geq \frac{g(r)}{\ln(g(r))}$. Podmínka $g(r) = \ln r \ln\ln r \geq \mathrm{e}$ platí pro $r \geq \mathrm{e}^{\mathrm{e}}$, tedy zřejmě i v našem případě $r \to \infty$. Nerovnost (4.13) proto můžeme dále upravit

$$\begin{aligned} \frac{\ln^2 M}{\ln\ln^2 M} &\geq (1+o(1))\frac{\ln r \ln\ln r}{\ln(\ln r \ln\ln r)} \\ \frac{\ln^2 M}{\ln\ln M} &\geq (1+o(1))\frac{2\cdot\ln r \ln\ln r}{\ln(\ln r \ln\ln r)}. \end{aligned}$$

Pro $r \geq 6$ navíc platí

$$2\cdot\frac{\ln r \ln\ln r}{\ln(\ln r \ln\ln r)} \geq \ln r,$$

tudíž při $r \to \infty$ lze psát

$$\frac{\ln^2 M}{\ln\ln M} \geq (1+o(1))\ln r. \tag{4.14}$$

ad (4) Dosazením za r do (4.14) dostáváme

$$\begin{aligned} \frac{\ln^2 M}{\ln\ln M} &\geq (1+o(1))\ln\left(2dN^{\frac{2}{d+1}}M^{d+1}\right) = \\ &= (1+o(1))\left(\ln 2d + 2\ln N^{\frac{1}{d+1}} + (d+1)\ln M\right) \geq \\ &\geq (1+o(1))\left(2\ln N^{\frac{1}{d+1}} + (d+1)\ln M\right), \end{aligned} \tag{4.15}$$

protože $\ln 2d$ je jistě nezáporné.

ad (5) Využijeme lemma 8 pro

$$\begin{aligned} k &\geq (1+o(1))(d+1) \geq \mathrm{e}, \\ l &= (1+o(1))\left(2\ln N^{\frac{1}{d+1}}\right) \geq 1, \\ v &= \ln M, \end{aligned}$$

přičemž v definujeme tak, aby $v \geq \mathrm{e}$. Teoreticky nám z toho plyne omezení na mez prosívací oblasti

$$\begin{aligned} \ln M &\geq \mathrm{e} \\ M &\geq \mathrm{e}^{\mathrm{e}} \doteq 15.15, \end{aligned}$$

nicméně v praxi se mez prosívací oblasti uvažuje nejméně v řádu tisíců.

Před dosazením do nerovnosti (4.15) ještě výrazy zjednodušíme. Označme

$$A := k \ln k$$
$$B := 2l \ln l$$

a odhadněme A, B zdola pomocí

$$\begin{aligned} A &\geq (1+o(1))(d+1)\ln\left((1+o(1))(d+1)\right) \geq \\ &\geq (1+o(1))d\ln d \qquad (4.16) \\ B &\geq 2\cdot(1+o(1))\left(2\ln N^{\frac{1}{d+1}}\right)\cdot\ln\left((1+o(1))\left(2\ln N^{\frac{1}{d+1}}\right)\right) \geq \\ &\geq (1+o(1))4\ln N^{\frac{1}{d+1}}\ln\ln N^{\frac{1}{d+1}} \qquad (4.17) \end{aligned}$$

Po dosazení výrazů (4.16) a (4.17) do tvrzení lemmatu 8 a drobném zjednodušení dostáváme

$$\begin{aligned} 2\ln M &= (1+o(1))\left(A+\sqrt{A^2+B}\right) \geq \\ &\geq (1+o(1))\left(d\ln d+\sqrt{(d\ln d)^2+4\ln N^{\frac{1}{d+1}}\ln\ln N^{\frac{1}{d+1}}}\right), \end{aligned}$$

což jsme chtěli dokázat. □

Analogickým postupem lze pak odvodit o něco lepší odhad doby běhu algoritmu, než je uveden v hypotéze 21, konkrétně vzorec (4.4)

$$\exp\left[(1+o(1))\left(d\ln d+\sqrt{(d\ln d)^2+4\ln N^{\frac{1}{d+1}}\ln\ln N^{\frac{1}{d+1}}}\right)\right].$$

Murphy v [21] uvádí (ovšem bez podrobnějšího vysvětlení), že minimalizací tohoto výrazu lze získat asymptotický vzorec pro stupeň nelineárního polynomu generovaného base-(m,p) metodou

$$d \approx \sqrt[3]{\frac{\delta\ln N}{\ln\ln N}}$$

pro $\delta = 3+o(1)$ a $N\to\infty$.

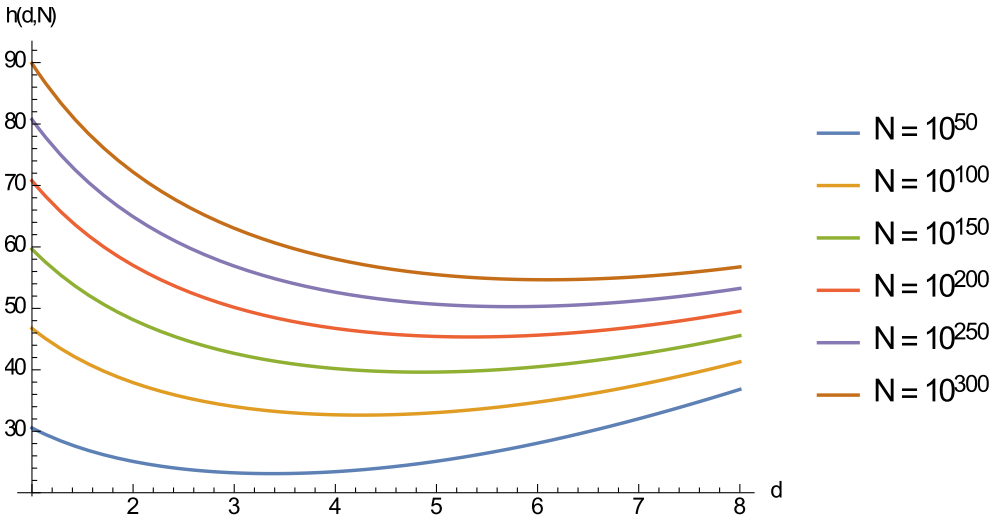
Úvahy skryté za odvozením vzorce obsahují pravděpodobně řadu heuristických zjednodušení, která se bohužel nepovedlo reprodukovat. Pokusili jsme se alespoň o následující numerickou analýzu.

Předpokládejme, že Murphyho odhad pro dobu běhu algoritmu číselného síta (4.4) platí rigorózně. Optimální volbu stupně d získáme minimalizací výrazu (4.4) pro N pevné a d rostoucí.

$$\arg\min_d\left(\exp\left[d\ln d+\sqrt{(d\ln d)^2+4\ln N^{\frac{1}{d+1}}\ln\ln N^{\frac{1}{d+1}}}\right]\right)$$

Jelikož exponenciální funkce je rostoucí, lze počítat pouze

$$\arg\min_d\left(d\ln d+\sqrt{(d\ln d)^2+4\ln N^{\frac{1}{d+1}}\ln\ln N^{\frac{1}{d+1}}}\right).$$



Obrázek 4.4: Průběh funkce (4.18) pro N pevné a d rostoucí

Označme

$$\begin{aligned} h(d,N) &= d\ln d + \sqrt{(d\ln d)^2 + 4\ln N^{\frac{1}{d+1}}\ln\ln N^{\frac{1}{d+1}}} = \\ &= d\ln d\sqrt{\bar{h}(d,N)} \end{aligned} \tag{4.18}$$

pro $\bar{h}(d,N) = (d\ln d)^2 + 4\ln N^{\frac{1}{d+1}}\ln\ln N^{\frac{1}{d+1}}$. Na obr. 4.4 je průběh této funkce zachycen pro různé volby N.

Funkce (4.18) je definovaná pro všechna $d > 0$ a $N > 1$, v naší situaci však uvažujeme ještě předpoklad $N > d^{2d^2} \geq 2$. V krajních bodech definičního oboru jsou limity

$$\lim_{d\to 0} h(d,N) = 2\sqrt{\ln N\ln\ln N} \qquad \text{a} \qquad \lim_{d\to\infty} h(d,N) = \infty.$$

Parciální derivací podle d dostáváme

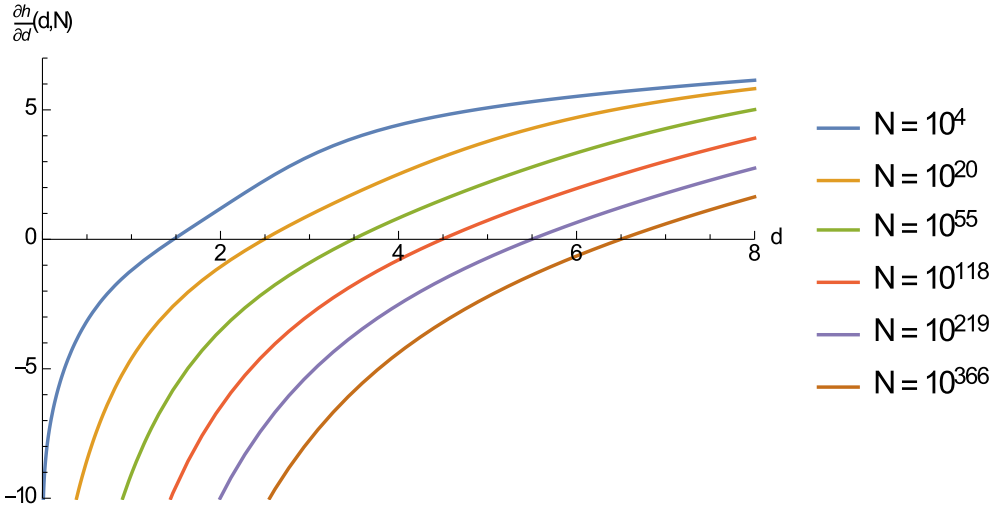
$$\begin{aligned} \frac{\partial h}{\partial d}(d,N) &= 1 + \ln d + \frac{1}{2}\frac{\frac{\partial\bar{h}}{\partial d}(d,N)}{\sqrt{\bar{h}(d,N)}} = \\ &= 1 + \ln d + \frac{2d\ln d + 2d\ln^2 d - \frac{4\ln N}{(d+1)^2} - \frac{4\ln N\ln\ln N^{\frac{1}{d+1}}}{(d+1)^2}}{2\sqrt{(d\ln d)^2 + 4\ln N^{\frac{1}{d+1}}\ln\ln N^{\frac{1}{d+1}}}}. \end{aligned}$$

Limity v krajních bodech jsou

$$\lim_{d\to 0}\frac{\partial h}{\partial d}(d,N) = -\infty \qquad \text{a} \qquad \lim_{d\to\infty}\frac{\partial h}{\partial d}(d,N) = \infty.$$

Po dosazení libovolného N_0 nalezneme řešením $\frac{\partial h}{\partial d}(d,N_0) = 0$ optimální hodnotu stupně d pro příslušné N_0. Nalezli jsme proto hraniční hodnoty N, v kterých je optimální stupeň na rozhraní d a $d+1$, viz. obr. 4.5.

Analogický postup jsme pro srovnání provedli i s výsledky pro base-m metodu. Srovnání intervalů, kdy je konkrétní stupeň d optimální pro tu kterou metodu uvádíme v tabulce 4.1.



Obrázek 4.5: Průběh derivace funkce (4.18) podle d pro N pevné a d rostoucí

stupeň d nelineárního polynomu	3	4	5	6	7
počet decimálních cifer N u base-m metody	< 30	$30 - 70$	$80 - 140$	$150 - 260$	> 260
počet decimálních cifer N u base-(m, p) metody	< 50	$60 - 110$	$120 - 210$	$220 - 360$	> 360

Tabulka 4.1: Optimální volba stupně d pro různé velikosti N

Z této tabulky také vidíme, že v obou případech uvažujeme zdaleka větší faktorizovaná čísla N, než jsou minima splňující předpoklad $N > d^{2d^2}$ z předpokladů hypotézy 21 i lemmatu 19 (viz tab. 4.2).

Vidíme, že pro danou hodnotu N lze stupeň polynomu v případě base-(m, p) metody volit nižší než pro base-m metodu. Také experimentální výsledky získané z implementace číselného síta ukazují, že srovnatelné hodnocení (jak jej popíšeme v podkapitole 4.2.1) v případě pevného N získávají polynomy s různým stupněm v závislosti na typu metody použité při generování polynomu. Podíl $\frac{\ln N}{\ln \ln N}$ považujeme ve vzorci pro stupeň d za klíčový, nicméně různé metody generování polynomu mohou pravděpodobně přispět ke zmenšení koeficientu δ.

Pro base-(m, p) metodu jsme se proto pokusili o zjištění přesné hodnoty δ tak, aby křivka $\sqrt[3]{\frac{\delta \ln N}{\ln \ln N}}$ co nejvěrohodněji odpovídala křivce

$$\min_{d \in \{3,\ldots,7\}} \left(\exp \left[d \ln d + \sqrt{(d \ln d)^2 + 4 \ln N^{\frac{1}{d+1}} \ln \ln N^{\frac{1}{d+1}}} \right] \right)$$

pro $N \in [10^{10}, 10^{400}]$. Pro účely hledání parametrů rovnic nelineárních křivek je možné využít nelineární metodou nejmenších čtverců. Konkrétně jsme použili metodu Levenberg-Marquardt implementovanou v softwaru Wolfram Mathematica (verze 10.3.1.0). Analogicky jsme postupovali i v případě base-m metody.

stupeň d nelineárního polynomu	3	4	5	6	7
minimální počet decimálních cifer N	9	20	35	57	83

Tabulka 4.2: Minimální velikost N pro různé stupně d

Hypotéza 23. *Buď $d = \sqrt[3]{\frac{\delta \ln N}{\ln \ln N}}$ asymptotický odhad stupně nelineárního polynomu.*

- *Pro base-m metodu je optimální volba stupně s parametrem $\delta = 2.56$.*
- *Pro base-(m, p) metodu je optimální volba stupně s parametrem $\delta = 2.30$.*

4.1.3 Kořenové vlastnosti

Druhým aspektem ovlivňujícím výtěžnost polynomu jsou takzvané kořenové vlastnosti. Pro posouzení vlivu kořenových vlastností na výtěžnost polynomu f na množině S zavedl Murphy parametr $\alpha(S)$, který efektivnost kvantifikuje. Pro odvození tohoto parametru bude nutné pracovat s hodnotou „průměrné valuace“ na množině. Ve snaze uchopit tento pojem co nejvíce rigorózně, zavedeme níže novou strukturu tzv. *přínosového schématu*, která nám umožňuje hodnotu průměrné valuace efektivně aproximovat.

Připomeňme, že p-valuací hodnoty v, značíme $\mathrm{val}_p(v)$, máme na mysli největší mocninu p dělící v (definice 6).

Definice 11 (Přínos)**.** *Mějme $S \subseteq \mathbb{Z} \setminus \{0\}$ a p prvočíslo. Přínos p v množině S definujeme jako očekávanou (střední) hodnotu p-valuace v pro $v \in S$, tedy*

$$\mathrm{cont}_p(S) = \mathsf{E}_{v \in S} \left[\mathrm{val}_p(v) \right].$$

Zaměřme se nejprve na výpočet hodnoty $\mathrm{cont}_p(S)$. Pro dostatečně velké rovnoměrně náhodně volené $S' \subset S$ konečné lze odhadnout očekávanou hodnotu přímo z definice jako

$$\mathrm{cont}_p(S) \approx \frac{\sum\limits_{v \in S'} \mathrm{val}_p(v)}{|S'|}. \tag{4.19}$$

Z praktického hlediska se však jedná o velice neefektivní přístup, proto nyní odvodíme jiný, výpočetně méně náročný, aproximativní vzorec. Pro přehlednost zavedeme nejdříve *přínosové schéma*, které využijeme pro odvození obecného vzorce v tvrzení 24. Následně obecný vzorec aplikujeme v tvrzení 25 a ukážeme konkrétní hodnoty přínosu pro vybrané struktury.

Definice 12 (Hustota množiny)**.** *Buď $D_n^l = \{-n, \ldots, -1, 0, 1, \ldots, n\}^l \subset \mathbb{Z}^l$ pro $l \in \mathbb{N}$ a množina $M \subseteq \mathbb{Z}^l$ libovolná. Řekneme, že množina M má (dobře definovanou) hustotu η, pokud platí*

$$\lim_{n \to \infty} \frac{\left| M \cap D_n^l \right|}{|D_n^l|} = \eta.$$

Poznámka. Množinu D_n^l lze definovat i obecnějším způsobem. Čtvercovou oblast jsme volili pro jednoduchost a přehlednost zápisu.

Buď $q \in \mathbb{N}$ libovolné, pak pro všechna $i, j \in \{0, 1, \ldots, q-1\}$ označíme

$$M_{i,j,q} = \{(a,b) \in \mathbb{Z} \times \mathbb{Z} \mid \mathrm{NSD}(a,b) = 1, a \bmod q \equiv i, b \bmod q \equiv j\}.$$

Zajímá nás, zda má $M_{i,j,q}$ dobře definovanou hustotu. Lze ukázat, že hustota $M_{i,j,q}$ existuje a pro (i,j) různé od $(0,0)$ je rovna $\frac{1}{q^2-1} \cdot \frac{6}{\pi^2}$. Za rigorózní důkaz tohoto vztahu děkujeme Víťovi Kalovi.

Příklad. Je-li $q = 2$, pak má $M_{0,0,2}$ hustotu 0 a $M_{i,j,2}$ hustotu $\eta = \frac{2}{\pi^2}$ pro $(i,j) \in \{(0,1),(1,0),(1,1)\}$.

Dvě nesoudělná sudá čísla neexistují, proto je hustota $M_{0,0,2}$ rovna 0. Relace největšího společného dělitele je symetrická, tedy platí rovnost hustot $M_{0,1,2}$ a $M_{1,0,2}$. Stačí proto ukázat rovnost hustot množin $M_{0,1,2}$ a $M_{1,1,2}$, tedy zda

$$\lim_{n\to\infty} \frac{|M_{0,1,2} \cap D_n^2|}{|D_n^2|} = \lim_{n\to\infty} \frac{|M_{1,1,2} \cap D_n^2|}{|D_n^2|}. \tag{4.20}$$

Pro pevné b liché, $|b| \leq n$, stačí ukázat, že když $n \to \infty$, pak sudých čísel menších nebo rovných n a nesoudělných s b je asymptoticky stejně jako lichých čísel menších nebo rovných n a nesoudělných s b. Označme $p_1, \ldots, p_k$ všechny prvočíselné dělitele b a $A_i = p_i\mathbb{Z} \cap D_n^1$. Množinu všech násobků b můžeme pak vyjádřit jako $\bigcup_{i=1}^k A_i$.

Indukcí nyní ukažme, že v $\bigcup_{i=1}^k A_i$ je zhruba polovina sudých a polovina lichých čísel. V každém A_i je přibližně stejně sudých a lichých čísel, případ $k = 1$ je tedy zřejmý. Buď nyní $k = 2$. V $A_1 \cap A_2$ je počet sudých a lichých čísel také asymptoticky stejný, protože se jedná právě o p_1-násobky p_2, kterých je asymptoticky polovina sudých a polovina lichých. Z toho a z indukčního předpokladu plyne, že i v

$$|A_1 \cup A_2| = |A_1| + |A_2| - |A_1 \cap A_2|$$

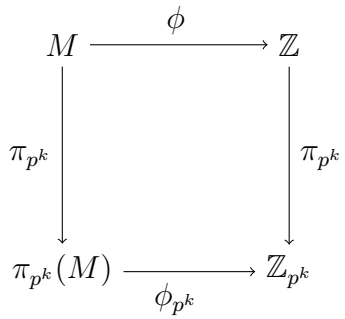
je počet sudých a lichých čísel asymptoticky shodný. Podle principu inkluze a exkluze lze tuto úvahu rozšířit na libovolný počet prvočíselných dělitelů b a stále dostáváme asymptotickou shodu počtu sudých a lichých čísel v $\bigcup_{i=1}^k A_i$.

Také v množině D_n^1 je přibližně polovina čísel sudých a polovina lichých. Když z této množiny vyjmeme násobky b, dostaneme zde právě všechna a nesoudělná s b. Z toho již plyne požadovaná limitní rovnost (4.20).

Označme $\eta(M_{i,j,q})$ hustotu množiny $M_{i,j,q}$. S využitím výsledku, že hustota množiny všech nesoudělných čísel je $\frac{6}{\pi^2}$, lze odvodit

$$\begin{aligned}
\frac{6}{\pi^2} &= \eta(M_{0,0,2}) + \eta(M_{0,1,2}) + \eta(M_{1,0,2}) + \eta(M_{1,1,2}) \\
\frac{6}{\pi^2} &= 0 + 3 \cdot \eta(M_{1,1,2}) \\
\frac{2}{\pi^2} &= \eta(M_{1,1,2}) = \eta
\end{aligned}$$

Definice 13 (Přínosové schéma)**.** *Dvojici (M, ϕ) označíme jako* přínosové schéma*, pokud množina $M \subseteq \mathbb{Z}^l$ pro $l \in \mathbb{N}$ má (dobře definovanou) hustotu, $0 \notin \phi(M)$ a pro každé prvočíslo p a přirozené $k \geq 1$ existuje jednoznačně určené zobrazení $\phi_{p^k} \colon \pi_{p^k}(M) \to \mathbb{Z}_{p^k}$ takové, že pro libovolné $v \in M$ platí $\phi_{p^k}(\pi_{p^k}(v)) = \pi_{p^k}(\phi(v))$ (zobrazení π_{p^k} zde znamená projekci modulo p^k).*



Obrázek 4.6: Znázornění přínosového schématu

Představu přínosového schématu znázorňuje obr. 4.6.

Tvrzení 24. *Mějme přínosové schéma* (M, ϕ) *a* p *prvočíslo. Pak přínos prvočísla* p *v množině* $\mathrm{Im}(\phi)$ *je*

$$\mathrm{cont}_p(\mathrm{Im}(\phi)) = \sum_{k=1}^{\infty} \frac{\left|\phi_{p^k}^{-1}(0)\right|}{\left|\pi_{p^k}(M)\right|}.$$

Důkaz. Na fakt, že p^k dělí v pro nějaké prvočíslo p a $k \geq 1$, lze také pohlížet jako na kongruenci $v \equiv 0 \bmod p^k$. V notaci přínosového schématu lze tudíž p-valuaci $v \in \mathrm{Im}(\phi)$ zapsat také jako

$$\begin{aligned} \mathrm{val}_p(v) &= \sum_{k=1}^{\infty} \left|\left\{v \in \mathrm{Im}(\phi) \colon v \equiv 0 \bmod p^k\right\}\right| = \\ &= \sum_{k=1}^{\infty} \left|\left\{v \in \mathrm{Im}(\phi) \colon \pi_{p^k}(v) = 0\right\}\right| = \\ &= \sum_{k=1}^{\infty} \left|\pi_{p^k}^{-1}(0)\right|. \end{aligned} \tag{4.21}$$

Pracujeme ovšem s přínosovým schématem, a to nám zaručuje, že existuje právě jedno zobrazení $\phi_{p^k} \colon \pi_{p^k}(M) \to \mathbb{Z}_{p^k}$ takové, že pro všechna $u \in M$ platí rovnost $\pi_{p^k}(\phi(u)) = \phi_{p^k}(\pi_{p^k}(u))$. To umožňuje výraz (4.21) ekvivalentně vyjádřit jako

$$\sum_{k=1}^{\infty} \left|\left\{v \in \mathrm{Im}(\phi) \colon \pi_{p^k}(v) = 0\right\}\right| = \sum_{k=1}^{\infty} \left|\left\{w \in \pi_{p^k}(M) \colon \phi_{p^k}(w) = 0\right\}\right| = \sum_{k=1}^{\infty} \left|\phi_{p^k}^{-1}(0)\right|.$$

Z definice přínosu 11 je pak

$$\mathrm{cont}_p(\mathrm{Im}(\phi)) = \mathsf{E}_{\,v \in \mathrm{Im}(\phi)} \left[\mathrm{val}_p(v)\right] = \sum_{k=1}^{\infty} \frac{\left|\phi_{p^k}^{-1}(0)\right|}{\left|\pi_{p^k}(M)\right|},$$

což jsme chtěli ukázat. □

Jednoznačná existence zobrazení ϕ_{p^k} z definice 13 plyne ze surjektivity zobrazení π_{p^k}, která je zajištěna dobře definovanou hustotou M. Konečná množina však z definice dobře definovanou hustotu mít nemůže. V případě číselného síta (resp.

praktických algoritmů obecně) však z principu není možné pracovat s nekonečnými množinami. Jak se s tímto omezením vypořádat a přitom zůstat konzistentní s definicí přínosového schématu?

Připomeňme, že výtěžnost polynomu na nějaké pevně dané konečné prosívací oblasti D hodnotíme podle počtu získaných B-hladkých relací pro předem zvolené B. V případě využití definice přínosového schématu v konkrétním případě číselného síta proto stačí, když bude surjektivita π_{p^k} (a tedy jednoznačná existence zobrazení ϕ_{p^k}) zajištěna jen pro všechna $p^k \leq B$. To lze jistě zajistit i pro konečné podmnožiny hustých množin. Pro účely následujícího tvrzení budeme takové konečné množiny označovat jako *konečně husté*. Ukážeme aproximativní vyjádření $\mathrm{cont}_p(S) = \mathrm{cont}_p(\mathrm{Im}(\phi))$ pro tři konkrétní ϕ s konečně hustými definičními obory $D \subset M$. Ve všech případech pak dostaneme konkrétní přínosové schéma (viz obr. 4.7), na které aplikujeme tvrzení 24.

Tvrzení 25. *Mějme (M, ϕ) konkrétní reprezentaci přínosového schématu a buď $D \subset M$ konečně hustá. Odvodíme tato vyjádření přínosu prvočísla p v množině $S = \phi(D)$:*

(i) Buď $M = \mathbb{Z} \setminus 0$ a $\phi\colon D \to D$ dané identitou, pak

$$\mathrm{cont}_p(S) = \mathrm{cont}_p(D) = \frac{1}{p-1}.$$

(ii) Buď $M = \mathbb{Z}$ a $\phi\colon D \to \mathrm{Im}(f)$ definované ireducibilním polynomem f, pak

$$\mathrm{cont}_p(S) = \mathrm{cont}_p(\mathrm{Im}(f)) = \frac{q_p}{p-1},$$

kde q_p značí počet kořenů f modulo p pro p nespeciální.

(iii) Buď $M = \{(a,b) \subset \mathbb{Z}^2\colon \mathrm{NSD}(a,b) = 1\}$ a $\phi\colon D \to \mathrm{Im}(F)$ definované homogenním polynomem F, pak

$$\mathrm{cont}_p(S) = \mathrm{cont}_p(\mathrm{Im}(F)) = \frac{p \cdot q_p}{p^2-1},$$

kde q_p značí počet kořenů F modulo p pro p nespeciální.

Poznámka. Existence konečně hustého $D \subset M$ je v bodech (i) a (ii) intuitivní. V případě číselného síta obvykle volíme jako interval $[-M, M] \subset \mathbb{Z}$, kde $M \gg B$. V bodě (iii) plyne z toho, že množina $M_{i,j,q}$ má dobře definovanou hustotu pro všechna přirozená q a $i, j \in \{1, \ldots, q-1\}$. V praxi volíme jako prosívací oblast množinu nesoudělných dvojic z $[-M_1, M_1] \times [-M_2, M_2]$ s $M_1, M_2 \gg B$, tedy obsahuje $\bigcup_{p^k \leq B} M_{i,j,p^k}$.

Důkaz.

ad (i) Pro D konečně husté musí být zobrazení ϕ_{p^k} také identita, což je prosté zobrazení a tedy nulový prvek má pouze jeden vzor. Dostáváme

$$\mathrm{cont}_p(S) = \mathrm{cont}_p(D) = \sum_{k=1}^{\infty} \frac{1}{p^k} = \frac{1}{p-1}.$$

ad (ii) Pro $v \in D$ je $\phi(v) = f(v)$. Označme $w = \pi_{p^k}(v)$, zobrazení ϕ_{p^k} lze definovat jediným způsobem jako $\phi_{p^k}(w) = f(v) \bmod p^k$, protože $\pi_{p^k}(D) = \mathbb{Z}_{p^k}$. Zajímá nás tudíž počet kořenů $f \bmod p^k$. Pro p nespeciální jsou tyto kořeny podle Henselova lemmatu (viz věta 9) jednoznačně určeny kořeny f modulo p. Je-li q_p počet kořenů f modulo p, pak lze celkový přínos odhadnout výrazem

$$\mathrm{cont}_p(S) = \mathrm{cont}_p(\mathrm{Im}(f)) = \sum_{k=1}^{\infty} \frac{q_p}{p^k} = \frac{q_p}{p-1}.$$

ad (iii) Analogickým argumentem jako v předchozím případě lze odvodit, že $\phi_{p^k}(u, v) = F(a, b) \bmod p^k$ pro $u = \pi_{p^k}(a)$ a $v = \pi_{p^k}(b)$, tudíž se budeme zajímat o počet kořenů $F \bmod p^k$. Navíc si uvědomme, že na $\pi_k(D)$ lze pohlížet také jako na projektivní přímku $\mathbf{P}^1\left(\mathbb{Z}_{p^k}\right)$. Projektivní přímkou nad okruhem $\mathbb{Z}_{p^k}$ rozumíme rozšíření definice projektivní přímky nad tělesem, kde ztotožníme všechny násobky invertibilním prvkem ze $\mathbb{Z}_{p^k}$.

Pro $p^k \mid F(a,b)$, kde $(a, b) \in D$, nastávají dva vzájemně se vylučující případy:

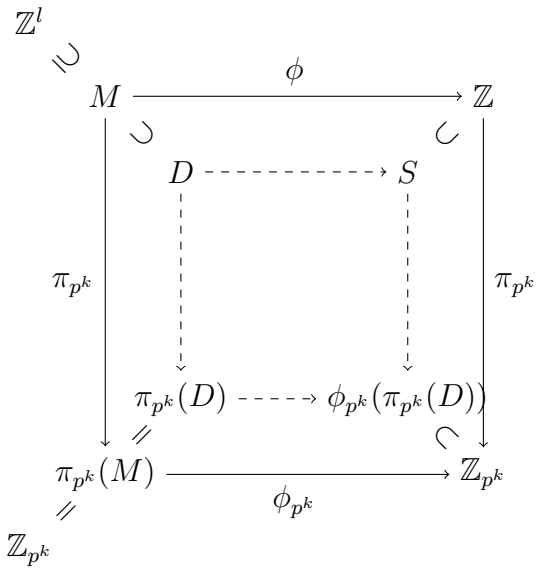
a) Prvočíslo p nedělí b, pak existuje $c \equiv \frac{a}{b} \bmod p^k$, pro které lze psát $F(c, 1) = f(c) \equiv 0 \bmod p^k$. Takové dvojice definičního oboru $\pi_k(D)$ tvoří přesně p^k tříd ekvivalence odpovídající bodům $(c : 1)$ na projektivní přímce $\mathbf{P}^1\left(\mathbb{Z}_{p^k}\right)$, které korespondují s možnými afinními kořeny F. V každé třídě je právě $\varphi(p^k) = p^{k-1}(p-1) = p^k - p^{k-1}$ prvků, kde $\varphi(x)$ značí Eulerovu funkci, tedy počet všech nesoudělných čísel, která jsou menší či rovna x.

b) Platí $p \mid b$ (tedy $p \nmid a$), musí proto existovat $c \equiv \frac{b}{a} \bmod p^k$ takové, že platí $F(1, c) = c^d f\left(\frac{1}{c}\right) \equiv 0 \bmod p^k$, kde $d = \deg(f)$. Příslušné dvojice definičního oboru $\pi_k(D)$ tvoří $p^k - \varphi(p^k) = p^{k-1}$ tříd ekvivalence, které odpovídají bodům $(1 : c)$ projektivní přímky $\mathbf{P}^1\left(\mathbb{Z}_{p^k}\right)$ a korespondují s možnými projektivními kořeny F. V každé třídě je $\varphi(p^k)$ prvků.

Sečtením počtu tříd ekvivalence v obou skupinách získáme počet prvků definičního oboru $|\pi_k(D)| = p^k + p^{k-1}$. Ukázali jsme také vztah mezi kořeny homogenního polynomu F a příslušného (dehomogenizovaného) polynomu f modulo p^k. Pro nespeciální p proto můžeme znovu použít Henselovo lemma (věta 9) a omezit se na výpočet q_p – počtu kořenů F modulo p. Všechny třídy mají stejný počet prvků, proto pravděpodobnost, že náhodná dvojice (a, b) bude konkrétním kořenem F, je $\frac{1}{p^k + p^{k-1}}$. Přínos p lze spočítat jako

$$\mathrm{cont}_p(S) = \mathrm{cont}_p(\mathrm{Im}(F)) = \sum_{k=1}^{\infty} \frac{q_p}{p^k + p^{k-1}} = \frac{p \cdot q_p}{p^2 - 1}. \qquad \square$$

V případě speciálních p v částech (i) resp. (ii) nelze zaměnit q_p (počet kořenů polynomu modulo p) a q_{p^k} (počet kořenů modulo p^k). Nezbývá než počet zdvižených kořenů modulo p^k explicitně spočítat a pro odhad přínosu využít omezený součet $\sum_{k=1}^{e} \frac{q_{p^k}}{p^k}$ resp. $\sum_{k=1}^{e} \frac{q_{p^k}}{p^k + p^{k-1}}$ nebo využít vzorec (4.19).



Obrázek 4.7: Znázornění přínosového schématu s konečnými podmnožinami

Příspěvkem fáze generování polynomů k efektivitě číselného síta má být především vysoká výtěžnost polynomů. Proto budeme chtít nějakým způsobem kvantifikovat výhodu volby ϕ jako polynomiálního zobrazení oproti identitě a využijeme proto právě zavedené pojmy založené na kořenových vlastnostech polynomu.

Pojem přínosu (tedy očekávané valuace) p na konečně husté množině S nám umožňuje definovat tzv. *typickou B-hladkou hodnotu* S jako

$$\prod_{p \leq B} p^{\mathrm{cont}_p(S)},$$

častěji budeme užívat logaritmický tvar

$$\sum_{p \leq B} \mathrm{cont}_p(S) \ln p.$$

Uvažujeme-li náhodné $v \in S$, pak se bude po prosívání v průměru jevit jako

$$\ln v - \sum_{p \leq B} \mathrm{cont}_p(S) \ln p.$$

Dosazením za S pro $S = D$ konečně husté a $S = \mathrm{Im}(F)$ (případy (i) a (iii) tvrzení 25) a odečtením těchto hodnot dostaneme právě hledaný parametr kvantifikující výtěžnost polynomiálních hodnot oproti hodnotám náhodným. Tento parametr budeme dále značit jako $\alpha(S)$

$$\begin{aligned}\alpha(S) &= \sum_{p \leq B} \frac{1}{p-1} \ln p - \sum_{p \leq B} \frac{pq_p}{p^2-1} \ln p = \\ &= \sum_{p \leq B} \left(\frac{p+1-pq_p}{p^2-1} \right) \ln p,\end{aligned}$$

kde q_p značí počet kořenů F modulo p.

Předpokládejme nyní, že logaritmy náhodné hodnoty $v \in D$ a F-hodnoty $F(x,y)$ pro nějaké $x, y \in D$ se po prosívání rovnají, neboli platí následující rovnosti

$$\begin{aligned} \ln v - \sum_{p \le B} \frac{\ln p}{p-1} &= \ln F(x,y) - \sum_{p \le B} \mathrm{cont}_p(\mathrm{Im}(F)) \cdot \ln p \\ \ln v &= \ln F(x,y) + \alpha(\mathrm{Im}(F)) \\ v &= F(x,y) \cdot \mathrm{e}^{\alpha(\mathrm{Im}(F))}. \end{aligned}$$

Z poslední rovnosti je zřejmé, že hodnota $F(x,y)$ pro náhodné x, y bude mít (co do výtěžnosti příslušného polynomu) stejné vlastnosti jako náhodná hodnota velikosti $F(x,y) \cdot \mathrm{e}^{\alpha(\mathrm{Im}(F))}$. Vidíme (Murphy to i empiricky dokazuje [21, kap. 4]), že parametr $\alpha(\mathrm{Im}(F))$ lze tudíž použít jako měřítko výtěžnosti závislé na kořenových vlastnostech příslušného polynomu. Proto budeme volit polynomy s co nejmenší $\alpha(\mathrm{Im}(F))$.

4.2 Kombinované hodnocení

Vlastnost velikosti a kořenové vlastnosti spolu interagují. Proto bychom také rádi nalezli nějakou kombinovanou metodu hodnocení, která by spojila hodnocení polynomů na základě obou vlastností. Při odhadu počtu hladkých relací v kapitole 4.1.1 jsme pro všechny dvojice (a, b) z prosívací oblasti sčítali pravděpodobnost, že náhodné celé číslo z intervalu $[0, |F(a,b)|]$ je B-hladké. Jak jsme ukázali v předchozím odstavci, vezmeme-li v potaz vliv kořenových vlastností, stačí pracovat na intervalu $[0, |F(a,b)| \cdot \mathrm{e}^{\alpha(\mathrm{Im}(F))}]$. Na základě toho redefinujeme vzorec (4.1) pro odhad počtu hladkých relací pro účely kombinovaného hodnocení jako

$$\sum_{(a,b) \in \mathcal{M}'} \rho\left(\frac{\ln|F(a,b)| + \alpha(\mathrm{Im}(F))}{\ln B}\right) \rho\left(\frac{\ln|G(a,b)| + \alpha(\mathrm{Im}(G))}{\ln B}\right). \qquad (4.22)$$

Poznamenejme, že také zde může mít každý polynom svoji vlastní mez hladkosti – B_F a B_G.

V oblasti $\mathcal{M}'$ je však standardně velké množství bodů a počítání (4.22) přes všechny tyto body by bylo neúměrnou výpočetní zátěží pro celý algoritmus číselného síta (uvědomme si, že navíc chceme hodnotit desítky či stovky polynomů). Murphy [21] proto navrhuje celou oblast $\mathcal{M}'$ rozdělit na menší podoblasti, v kterých se hodnota $F(x,y)$ (a pravděpodobně ani $\rho\left(\frac{\ln|F(x,y)| + \alpha(\mathrm{Im}(F))}{\ln B}\right)$) nebude příliš měnit. Pak stačí z každé této podoblasti vzít jen jednoho zástupce, jakousi střední hodnotu $F(x,y)$ v dané podoblasti.

Přejděme nyní od kartézské souřadnicové reprezentace k souřadnicím polárním, tedy

$$F(x,y) \rightsquigarrow F(r \cdot \cos\theta, r \cdot \sin\theta) = r^{\deg(F)} \cdot F(\cos\theta, \sin\theta).$$

Vidíme, že pro každé dva polynomy stejného stupně, bude i koeficient $r^{\deg(F)}$ stejný, a proto stačí pro účely hodnocení výtěžnosti uvažovat jen hodnotu $F(\cos\theta, \sin\theta)$. Tudíž místo všech bodů v oblasti $\mathcal{M}'$ budeme testovat jen body na

jednotkové půlkružnici. Interval $[0,\pi]$ rozdělíme rovnoměrně do K podintervalů (pro nějaké vhodně zvolené $K \in \mathbb{N}$) a volíme $\theta_i = \frac{\pi}{K}\left(i - \frac{1}{2}\right)$ pro $i = 1, \ldots, K$. Hodnotu $F(\cos\theta_i, \sin\theta_i)$ prohlásíme za reprezentanta na dané podoblasti. Pro všechna $i = 1, \ldots, K$ označme

$$\tilde{F}(\cos\theta_i, \sin\theta_i) = \begin{cases} F(\cos\theta_i, \sin\theta_i) & \text{pokud } F(\cos\theta_i, \sin\theta_i) \notin (-1, 1), \\ 1 & \text{jinak} \end{cases}$$

a analogicky $\tilde{G}(\cos\theta_i, \sin\theta_i)$. Pak

$$\sum_{i=1}^{K} \rho\left(\frac{\ln\left|\tilde{F}(\cos\theta_i, \sin\theta_i)\right| + \alpha(\mathrm{Im}(F))}{\ln B}\right) \rho\left(\frac{\ln\left|\tilde{G}(\cos\theta_i, \sin\theta_i)\right| + \alpha(\mathrm{Im}(G))}{\ln B}\right)$$

je funkce pro odhad kombinovaného hodnocení polynomů F a G dle jejich předpokládané výtěžnosti na prosívací oblasti $\mathcal{M}'$. Někdy bývá tato funkce označována jako $\mathbb{E}(F, G)$.

Co se týče optimální volby počtu intervalů K, v průběhu implementace jsme empiricky vyzkoušeli, že zcela postačuje volba K v řádu stovek.

4.2.1 Výtěžnost zkosených polynomů

Jak jsme diskutovali výše (část 4.1.1), prosívání s polynomy z Kleinjungovy metody provádíme na zkreslené oblasti $\mathcal{S}' = [-M\sqrt{s}, M\sqrt{s}] \times [1, \frac{M}{\sqrt{s}}]$, proto chceme definovat i kombinované hodnocení pro zkosené polynomy na zkreslené oblasti. Připomeňme, že v předchozím případě jsme kvůli efektivnímu výpočtu přecházeli od kartézské k polární souřadnicové reprezentaci, což učiníme i nyní, jen s tím rozdílem, že se nebudeme pohybovat na jednotkové půlkružnici, nicméně na její *zkreslené* (anglicky *skewed*) variantě – „jednotkové“ půlelipse. „Jednotkové“ ve smyslu, že se budeme zajímat pouze o poměr mezi její hlavní a vedlejší půlosou, což je právě parametr zkreslení s.

$$F(x,y) \rightsquigarrow F\left(r\sqrt{s}\cdot\cos\theta, r\frac{1}{\sqrt{s}}\cdot\sin\theta\right) = r^{\deg(F)}\cdot F\left(\sqrt{s}\cos\theta, \frac{1}{\sqrt{s}}\sin\theta\right).$$

Pro přehlednost označme pro všechna $i = 1, \ldots, K$

$$u_F(\theta_i) = \frac{\ln\left|\tilde{F}\left(\sqrt{s}\cos\theta_i, \frac{1}{\sqrt{s}}\sin\theta_i\right)\right| + \alpha(\mathrm{Im}(F))}{\ln B},$$

analogicky $u_G(\theta_i)$. Pak lze funkci $\mathbb{E}(F, G)$ předefinovat pro výpočet na zkosené prosívací oblasti jako

$$\sum_{i=1}^{K} \rho\left(u_F(\theta_i)\right) \rho\left(u_G(\theta_i)\right) \tag{4.23}$$

Nyní máme metodu, jak hodnotit polynomy získané přímo z Kleinjungova algoritmu na základě kombinace jejich velikostních i kořenových vlastností. Existují způsoby, jak polynomy před hodnocením vylepšit, abychom získali polynomy s lepšími vlastnostmi. V následující části princip vylepšení popíšeme. Vyjdeme z postupů, které navrhuje Murphy [21], a navrhneme rozšíření pro případ polynomů z base-(m, p) rozkladu. Popsané řešení je také součástí implementace.

4.3 Optimalizace výtěžnosti

Jak jsme předestřeli v úvodu této kapitoly, mezi generování velkého množství polynomů a výběr nejlepšího kandidáta na základě právě popsaného kombinovaného hodnocení vkládáme proces optimalizace polynomů, který popíšeme v této části. Optimalizací rozumíme souhrnný název pro dva procesy, které původní polynom modifikují. Nejprve optimalizujeme polynomy vzhledem k vlastnosti velikosti a provádíme hodnocení na základě L^2-normy, které jsme odvodili v kapitole 4.1.1. Polynomy, které splní stanovenou mez pro L^2-normu poté optimalizujeme vzhledem ke kořenovým vlastnostem tak, abychom příliš nezasahovali do vlastnosti velikosti. Kořenové vlastnosti kvantifikujeme parametrem α zavedeným v kapitole 4.1.3. Užitečnými způsoby, jak ovlivňovat vlastnosti polynomu, jsou tzv. *translace* a *rotace*, které zavedeme níže. Jedná se o parametrizovaná zobrazení ovlivňující vlastnosti polynomu. Popíšeme také metody, které jsme při implementaci využili k nalezení optimálních parametrů těchto zobrazení.

Uvažujeme polynom $f(x)$ stupně d generovaný pomocí base-(m, p) rozkladu a lineární polynom $g(x) = px - m$. Translace i rotace jsou zobrazení ze $\mathbb{Z}[x]$ do $\mathbb{Z}[x]$:

- Buď $t \in \mathbb{Z}$, pak *translace* $f(x)$ *o* t je zobrazení $\tau_t \colon \mathbb{Z}[x] \to \mathbb{Z}[x]$ definované předpisem $f(x) \mapsto f(x + t)$.
- Buď $\lambda(x) \in \mathbb{Z}[x]$, pak *rotace* $f(x)$ *o* $\lambda(x)$ je zobrazení $\tau_{\lambda(x)} \colon \mathbb{Z}[x] \to \mathbb{Z}[x]$ definované předpisem $f(x) \mapsto f(x) + \lambda(x) \cdot g(x)$.

Kroky optimalizace vlastností polynomu musí probíhat s ohledem na zachování faktu, že polynomy f a g mají sdílet stejný kořen modulo rozkládané číslo N. Po translaci polynomu $f(x)$ o t je tudíž nutné upravit i polynom $g(x)$ na $g(x) + t \cdot p$ a společný kořen $\frac{m}{p} \bmod N$ na $\frac{m}{p} - t \bmod N$. Při translaci se sice mění hodnoty kořenů $\tau_t(f)$ oproti f, nicméně množina $\mathrm{Im}(\tau_t(f)) = \mathrm{Im}(f)$, proto také translace nebude ovlivňovat kořenové vlastnosti. V případě rotace zjevně zůstává společný kořen zachován, proto není třeba nijak upravovat ani lineární polynom $g(x)$. Zřejmě ale $\mathrm{Im}(\tau_{\lambda(x)}(f)) \neq \mathrm{Im}(f)$, tudíž rotace ovlivňuje kořenové vlastnosti. Vlastnost velikosti koeficientů je ovlivněna jak translací, tak rotací.

Je-li translační parametr t výrazně menší než parametr zkreslení s, při rozepsání koeficientů $\tau_t(f)$ vidíme, že translace nemá až takový vliv na vysoce zkosené polynomy. Proto pro optimalizaci vlastnosti velikosti koeficientů zkosených polynomů využijeme vhodnou kombinaci translace i rotace. Úskalím rotace je však volba vhodného rotačního polynomu $\lambda(x)$. Z Kleinjungova algoritmu máme polynom, jehož koeficienty a_d, a_{d-1} i a_{d-2} jsou vhodně malé, proto je v rámci optimalizace už měnit nechceme. U polynomů stupně 4 nemá tedy rotace valného smyslu. Je-li $\deg(f) = d = 5$, chceme rotací ovlivnit nejvýše koeficient u kvadratického členu, proto volíme rotační polynom lineární. Pro $d > 5$ je možné volit rotační polynom i kvadratický.

Formálně lze při zavedeném značení optimalizaci velikosti polynomu popsat takto, označme $\bar{f} = \tau_{\lambda(x)}(\tau_t(f))$, $\bar{F}$ buď homogenizace $\bar{f}$ a mějme referenční „jednotkovou“ zkreslenou prosívací oblast $\mathcal{S}'' = [-\sqrt{s}, \sqrt{s}] \times [-\frac{1}{\sqrt{s}}, \frac{1}{\sqrt{s}}]$. Pak (dle

úvah kap. 4.1.1) chceme minimalizovat hodnotu logaritmu integrálu

$$\ln\left(\iint_{\mathcal{S}''} \bar{F}^2(x,y)\,\mathrm{d}x\,\mathrm{d}y\right). \tag{4.24}$$

Murphy [21] navrhuje použít pro řešení jakoukoliv minimalizační metodu pro více proměnných, např. metodu největšího spádu. Ukazuje se však (např. v implementaci číselného síta GGNFS [19]), že přímočaré použití metody největšího spádu není možné, protože vzhledem k velkým koeficientům polynomů metoda nemusí konvergovat vůbec nebo jen velmi pomalu. Pro implementaci jsme tudíž zvolili upravenou variantu metody největšího spádu, jak ji navrhuje P. Jedlička v článku [15]. Podrobněji tuto metodu popisujeme v kapitole 6.1.3.

Po skončení minimalizačního procesu již průběžně filtrujeme polynomy, které splňují nějakou zvolenou mez pro hodnotu (4.24). Mez pro takové filtrování nemusí být nutně statická po celou dobu běhu generování polynomů, průběžnou úpravou lze například dosáhnout odfiltrování polynomů, u kterých je již v této chvíli zřejmé, že neaspirují na celkové kvalitní hodnocení. Podrobným studiem tohoto filtru jsme se však v naší práci nezabývali.

Buď nyní $f(x)$ polynom po optimalizaci vzhledem k vlastnosti velikosti. Dalším krokem je optimalizace kořenových vlastností. Ty ovlivňuje pouze proces rotace, proto budeme hledat takový rotační polynom $\lambda(x)$, pro který má $\tau_{\lambda(x)}(f)$ nejlepší kořenové vlastnosti. Nechceme ovšem nijak výrazně měnit vlastnost velikosti polynomu, proto uvažujeme již jen lineární rotační polynom. Označme $f_{c_1c_0} \coloneqq \tau_{\lambda(x)}(f) = f(x) + (c_1x + c_0)\cdot g(x)$, hledáme takovou volbu $c_0, c_1 \in \mathbb{Z}$, pro kterou je $\alpha(\mathrm{Im}(F_{c_1c_0}))$ nejmenší možná na prosívací oblasti. Různým metodám pro řešení tohoto problému se věnujeme podrobně v kapitole 5.

Díky volbě nízkého stupně rotačního polynomu $\lambda(x)$ můžeme předpokládat rovnost logaritmů integrálů

$$\ln\left(\iint_{\mathcal{S}''} \bar{F}^2(x,y)\,\mathrm{d}x\,\mathrm{d}y\right) \approx \ln\left(\iint_{\mathcal{S}''} F^2_{c_1c_0}(x,y)\,\mathrm{d}x\,\mathrm{d}y\right).$$

Nové koeficienty proto dopočítáme jen pro polynomy, jejichž úvodní hodnocení spočítané jako

$$\ln\left(\iint_{\mathcal{S}''} F^2_{c_1c_0}(x,y)\,\mathrm{d}x\,\mathrm{d}y\right) + \alpha(\mathrm{Im}(F)),$$

je dostatečně nízké.

Získali jsme vylepšený polynom f s optimalizovanými hodnotami obou vlastností. Na závěr ještě podle nového společného kořene, který jsme získali translací při optimalizaci velikosti, vyjádříme i nový lineární polynom g.

Průběh optimalizace vlastností provede filtrování polynomů na základě stanovených mezí pro průběžné hodnocení jednotlivých vlastností. Na zbytek polynomů aplikujeme kombinované hodnocení, které jsme popsali v kapitole 4.2 (případně zkušební prosívání) a vybereme nejlepší dvojici, která je předána do další fáze algoritmu číselného síta.

Kapitola 5

Optimalizace kořenových vlastností

Nejdříve jsme pomocí Kleinjungova algoritmu vygenerovali velké množství polynomů, které jsme následně pomocí translace a rotace optimalizovali na velikost. Nyní budeme polynomy se srovnatelnou velikostí optimalizovat vzhledem k jejich kořenovým vlastnostem tak, abychom velikost již výrazně neměnili. Vysvětlíme nejprve princip tzv. *kořenového síta*, jak jej zavádí Murphy v práci [21], a následně ukážeme optimalizaci kořenového síta použitou v naší implementaci. Závěr kapitoly věnujeme popisu nejefektivnější známé modifikace kořenového síta, Baiově *dvoufázovému kořenovému sítu* [3].

5.1 Kořenové síto

Mějme $f(x) \in \mathbb{Z}[x]$ polynom po optimalizaci vzhledem k vlastnosti velikosti a $\lambda(x) = c_1x+c_0$ rotační polynom. Uvažujeme již jen lineární $\lambda(x)$, abychom rotací výrazněji neovlivňovali vlastnost velikosti polynomu $f(x)$. Označme $F_{c_1c_0}(x, y)$ homogenizaci zrotovaného $\tau_{\lambda(x)}(f)$. Jak jsme ukázali v podkapitole 4.1.3, cílem kořenového síta je najít taková $c_0, c_1 \in \mathbb{Z}$, která zajišťují nejmenší $\alpha(\mathrm{Im}(F_{c_1c_0}))$ na dané prosívací oblasti.

Zamysleme se obecně, kdy bude hodnota $\alpha(S)$ nejmenší? Analyzujme vzorec pro výpočet $\alpha(S) = \sum_{p \leq B} \left(\frac{p+1-q_pp}{p^2-1}\right) \ln p$. Jelikož $p \geq 2$, je $\ln p$ rostoucí kladná funkce a zároveň $p^2 - 1 > 0$. Proto bude-li $q_p > 1$, pak bude $p + 1 - q_pp < 0$. Zpět v našem konkrétním případě pro $\alpha(\mathrm{Im}(F_{c_1c_0}))$ značí q_p počet kořenů F modulo p. Tyto kořeny (jak jsme ukázali v kapitole 4.1.3) korespondují s kořeny f modulo p^k pro nespeciální p a k taková, že $p^k < B$ pro mez hladkosti B. Budeme tedy upřednostňovat polynomy $f(x)$ s co nejvíce kořeny modulo malé mocniny p^k.

Hlavní myšlenku kořenového prosívání ukazuje následující lemma:

Lemma 26. *Buď $f_{c_1,c_0}(x) = \tau_{\lambda(x)}(f)$ zrotovaný polynom lineárním $\lambda(x) = c_1x+c_0$ a p^k mocnina prvočísla. Je-li r kořenem $f_{c_1,c_0}(x) \bmod p^k$, pak je také kořenem $f_{c_1+ip^k,c_0+jp^k}(x) \bmod p^k$ pro všechna $i, j \in \mathbb{Z}$.*

Důkaz. Z definice rotace o $\lambda(x)$ je $f_{c_1,c_0}(x) = f(x) + (c_1x + c_0)g(x)$ pro $g(x)$ lineární. Pak tedy

$$\begin{aligned} f_{c_1+ip^k,c_0+jp^k}(x) &= f(x) + \Big((c_1 + ip^k)x + (c_2 + jp^k)\Big)\, g(x) = \\ &= f(x) + (c_1x + c_0)g(x) + (ip^kx + jp^k)g(x) = \\ &= f_{c_1,c_0}(x) + p^k \cdot (ix + j)g(x) \end{aligned}$$

Je tedy $f_{c1,c0}(x) \equiv f_{c1+ip^k,c_0+jp^k}(x) \bmod p^k$, z čehož plyne tvrzení věty. □

Postup kořenového prosívání popisuje algoritmus 5.1. Vysvětleme nyní podrobněji jednotlivé kroky. Řekněme, že rotační koeficienty mohou nabývat hodnot z oblasti $[-C_0, C_0] \times [-C_1, C_1] \subseteq \mathbb{Z}^2$. Pak lemma 26 umožňuje místo výpočtu $(2C_0 + 1)(2C_1 + 1)$ hodnot $\alpha(\mathrm{Im}(F_{c_1c_0}))$ počítat pro pevné p^k dílčí přínosy $\mathrm{cont}_p(\mathrm{Im}(F_{c_1c_0}))$ pouze pro $c_0, c_1 \in \{0, \ldots, p^k\}$ a tyto výsledky replikovat přes celou rotační oblast $[-C_0, C_0] \times [-C_1, C_1]$. Sečtením přes všechna k (řádek 2) dostáváme $\mathrm{cont}_p(\mathrm{Im}(F_{c_1c_0}))$ pro všechny možné polynomy $F_{c_1c_0}$. Cyklem přes všechna malá prvočísla p (řádek 1) dopočítáme $\alpha(\mathrm{Im}(F_{c_0c_1}))$. Výhodné je navíc procházet přes množinu všech potenciálních kořenů $f_{c_1c_0}(x) \bmod p^k$ (řádek 3) a průběžně aktualizovat hodnotu přínosu $\mathrm{cont}_p(\mathrm{Im}(F_{c_1+ip^k,c_0+jp^k}))$, přínos každého kořene je $\frac{1}{p^{k-1}(p+1)}$, což plyne z tvrzení 25. Po skončení algoritmu vybereme ty koeficienty c_0 a c_1, pro které je spočítaná $\alpha(\mathrm{Im}(F_{c_0c_1}))$ v celé rotační oblasti $[-C_0, C_0] \times [-C_1, C_1]$ nejmenší.

Algoritmus 5.1 Kořenové síto

input : polynomy $f(x), g(x)$, meze rotační oblasti C_0, C_1, mez hladkosti B

output : aproximované hodnoty $\alpha(\mathrm{Im}(F_{c_1c_0}))$ v rotační oblasti $[-C_0, C_0] \times [-C_1, C_1]$

```
1 : for (p ≤ B, p prvočíslo)
2 :   for (k: p^k ≤ B)
3 :     for (r = 0 ... p^k − 1)
4 :       for (c_1 = 0 ... p^k − 1)
5 :         spočítej c_0 z f(r) + (c_1 r + c_0) g(r) ≡ 0   (mod p^k)
6 :         aktualizuj cont_p(Im(F_{c_1+ip^k, c_0+jp^k}))
7 :   aktualizuj α(Im(F_{c_1 c_0}))
```

Lemma 27. *Pro C_0, C_1 meze rotační oblasti a B mez hladkosti lze výpočetní náročnost algoritmu kořenového síta (alg. 5.1) asymptoticky odhadnout jako*

$$4C_0C_1 \sum_{\substack{p \le B \\ p\ prvočíslo}} \sum_{k:\, p^k \le B} \mathcal{O}(1).$$

Důkaz. Výpočet hodnoty c_0 lze provést v konstantním čase, stejně jako aktualizaci přínosu, tu je však nutné aktualizovat pro všechny polynomy $F_{c_1+ip^k,c_0+jp^k}$.

Z algoritmu 5.1 pak plyne přímo odhad

$$\sum_{p\le B}\sum_{k:\ p^k\le B}\sum_{r=0}^{p^k-1}\sum_{c_1=0}^{p^k-1}\left(\mathcal{O}(1)+\frac{2C_1+1}{p^k}\frac{2C_0+1}{p^k}\mathcal{O}(1)\right)=$$
$$=\sum_{p\le B}\sum_{k:\ p^k\le B}\left(p^k\cdot p^k\cdot\mathcal{O}\left(\frac{4C_1C_0}{p^{2k}}\right)\right)=$$
$$=4C_0C_1\sum_{p\le B}\sum_{k:\ p^k\le B}\mathcal{O}(1).$$

□

5.2 Optimalizace kořenového síta

První bod Henselova lemmatu 9 nám dává návod, jak kořenové síto dále optimalizovat. Narazíme-li totiž na jednoduchý kořen, můžeme přínos v tomto bodě spočítat rovnou, aniž bychom museli kořen zdvihat, protože každý jednoduchý kořen modulo p jednoznačně určuje právě jeden kořen modulo p^k pro všechna k. Spíše pak implementační záležitostí je možnost nahrazení výpočtu $\alpha(\mathrm{Im}(F_{c_0c_1}))$ pouze za součty $\sum_{p\le B}\mathrm{cont}_p(\mathrm{Im}(F_{c_0c_1}))$.

Jak efektivně rozeznat násobné kořeny zrotovaných polynomů $f_{c_1c_0}(x)$ modulo p ukazuje následující lemma.

Lemma 28. *Buď f, g polynomy generované base-(m,p) metodou, označme rotaci polynomu f s rotačními koeficienty z oblasti $[-C_0,C_0]\times[-C_1,C_1]\subset\mathbb{Z}^2$ jako $f_{c_1c_0}(x)=f(x)+(c_1x+c_0)g(x)$ a buď r kořenem $f_{c_1c_0}(x)$ mod p pro p prvočíslo. Pak kongruence*

$$c_1g^2(r)\equiv f(r)g'(r)-f'(r)g(r)\bmod p$$

má právě jedno řešení a r je násobný kořen $f_{c_1c_0}(x)$ mod p.

Důkaz. Předpokládejme nejprve, že $g(x)\not\equiv 0 \bmod p$. Pak lze jediné řešení kongruence z tvrzení vyjádřit jako

$$\bar{c_1}\equiv\frac{f(r)g'(r)-f'(r)g(r)}{g^2(r)}\mod p.$$

Navíc můžeme psát

$$\begin{aligned}c_1g^2(r)&\equiv f(r)g'(r)-f'(r)g(r)\mod p\\ c_1g(r)&\equiv\frac{f(r)}{g(r)}g'(r)-f'(r)\mod p.\end{aligned}\tag{5.1}$$

Z definice zrotovaného polynomu $f_{c_1c_0}(x)$ a faktu, že r je kořenem $f_{c_1c_0}(x)$ mod p, máme vztah:

$$\begin{aligned}0&\equiv f(r)+(c_1r+c_0)g(r)\mod p\\ \frac{f(r)}{g(r)}&\equiv-(c_1r+c_0)\mod p.\end{aligned}$$

Dosazením tohoto vztahu do (5.1) dostáváme

$$\begin{aligned} c_1 g(r) &\equiv -(c_1 r + c_0) g'(r) - f'(r) \mod p \\ f'(r) + c_1 g(r) + (c_1 r + c_0) g'(r) &\equiv 0 \mod p \\ f'_{c_1 c_0}(r) &\equiv 0 \mod p, \end{aligned}$$

z čehož plyne, že r je násobným kořenem $f_{c_1 c_0}(x) \bmod p$.

Kdyby $g(r) \equiv 0 \bmod p$, pak by muselo p dělit oba koeficienty $g(r) = pr - m$, protože $r < p$. Jenže v base-(m, p) metodě předpokládáme $\mathrm{NSD}(m, p) = 1$, jak jsme ukázali v kapitole 3.2. Případ $g(r) \equiv 0 \bmod p$ je tedy vyloučen. □

Algoritmus 5.2 popisuje kořenové síto s navrženými vylepšeními. Jako v případě obecného kořenového síta procházíme přes všechny možné kořeny tentokrát však pouze modulo p (řádek 2) a aktualizujeme přínos najednou pro všechny příslušné polynomy. Stěžejní je však rozlišení mezi jednoduchými a násobnými kořeny v kroku 3, jež zdůvodňujeme v lemmatu 28. V případě jednoduchého kořene můžeme hodnotou $\frac{p}{p^2-1}$ aktualizovat přínos u všech F_{c_1+ip, c_0+jp}. Pokud narazíme na kořen násobný, lze aktualizovat přínos jen hodnotou $\frac{1}{p^{k-1}(p+1)} = \frac{1}{p+1}$ pro $k = 1$, poté musíme provést zdvihnutí, pro násobné kořeny $f_{c_1 c_0}(x)$ modulo p^2 aktualizovat přínos u všech $F_{c_1+ip^2, c_0+jp^2}$ a tentýž postup zopakovat pro všechna p^k menší než mez hladkosti B. Po zopakování celého postupu pro všechna B-hladká p dostáváme aproximované hodnoty přínosu pro polynomy v celé rotační oblasti a vybíráme polynom s největším přínosem.

Algoritmus 5.2 Optimalizované kořenové síto

input : polynomy $f(x), g(x)$, meze rotační oblasti C_0, C_1, mez hladkosti B

output : aproximace $\mathrm{cont}_p(\mathrm{Im}(F_{c_1 c_0}))$ v rotační oblasti $[-C_0, C_0] \times [-C_1, C_1]$

```
1 :  for (p ≤ B, p prvočíslo)
2 :    for (r = 0 ... p − 1)
3 :      spočítej c̄1 : c̄1 g²(r) ≡ f(r)g'(r) − f'(r)g(r)  (mod p)
4 :      for (c1 = 0 ... p − 1)
5 :        spočítej c0 z f(r) + (c1 r + c0)g(r) ≡ 0  (mod p)
6 :        if (c̄1 ≠ c1)   // r je jednoduchý kořen
7 :          aktualizuj cont_p(Im(F_{c1+ip, c0+jp}))
8 :        else   // r je násobný kořen
9 :          for (k : p^k ≤ B)
10 :           aktualizuj cont_p(Im(F_{c1+ip^k, c0+jp^k}))
11 :           zdvihni násobný kořen
```

Lemma 29. *Pro C_0, C_1 meze rotační oblasti a B mez hladkosti lze výpočetní náročnost optimalizovaného algoritmu kořenového síta (alg. 5.2) asymptoticky odhadnout jako*

$$4 C_0 C_1 \sum_{\substack{p \le B \\ p \text{ prvočíslo}}} \left(\mathcal{O}(1) + \sum_{k:\, p^k \le B} \frac{1}{p^{2k-1}} \right).$$

Důkaz. Výpočet hodnot $\bar{c_1}$ a c_0 stejně jako aktualizaci přínosu a zdvihání považujeme za konstantní operaci. V $p-1$ případech jednoduchého kořenu aktualizujeme přínos pro všechna F_{c_1+ip,c_0+jp}. V jednom případě násobného kořenu je však nutné provádět zdvihání a aktualizovat přínos pro všechny polynomy $F_{c_1+ip^k,c_0+jp^k}$. Z algoritmu 5.2 pak plyne odhad

$$\begin{aligned}
&\sum_{p\leq B}\sum_{r=0}^{p-1}\left(\mathcal{O}(1)+\mathcal{O}(1)(p-1)\frac{(2C_0+1)(2C_1+1)}{p^2}+\right.\\
&\qquad\left.+\sum_{k\colon p^k\leq B}\left(\frac{2C_1+1}{p^k}\frac{2C_0+1}{p^k}+\mathcal{O}(1)\right)\right)=\\
&=\sum_{p\leq B}\left(p\cdot(p-1)\mathcal{O}\left(\frac{4C_0C_1}{p^2}\right)+\sum_{k\colon p^k\leq B}\mathcal{O}\left(\frac{4C_1C_0}{p^{2k}}\right)\right)=\\
&=4C_0C_1\sum_{p\leq B}\left(\mathcal{O}(1)+\sum_{k\colon p^k\leq B}\frac{1}{p^{2k-1}}\right).
\end{aligned}$$

□

Pro lepší představu o vylepšení, které kořenové síto přináší oproti přímému výpočtu hodnot $\alpha(\mathrm{Im}(F_{c_1c_0}))$ pro všechny možné rotace, formulujme ještě následující lemma.

Lemma 30. *Pro C_0, C_1 meze rotační oblasti a B mez hladkosti lze výpočetní náročnost přímého výpočtu hodnot $\alpha(\mathrm{Im}(F_{c_1c_0}))$ asymptoticky odhadnout jako*

$$(2C_0+1)(2C_1+1)\sum_{\substack{p\leq B\\ p\ prvočíslo}}\sum_{k\colon p^k\leq B}p^k.$$

Důkaz. Výpočet $\alpha(\mathrm{Im}(F_{c_1c_0}))$ obnáší nalezení všech kořenů modulo p^k pro všechna možná p^k. Pokud uvažujeme dopočítání přínosu a výpočet α již jako konstantní operaci, dostáváme odhad

$$\begin{aligned}
&\sum_{c_0=-C_0}^{C_0}\sum_{c_1=-C_1}^{C_1}\sum_{p\leq B}\sum_{k\colon p^k\leq B}\sum_{r=0}^{p^k-1}\mathcal{O}(1)=\\
&(2C_0+1)(2C_1+1)\sum_{\substack{p\leq B\\ p\ \mathrm{prvočíslo}}}\sum_{k\colon p^k\leq B}p^k.
\end{aligned}$$

□

Na základě vzorců odvozených v lemmatech 27, 29 a 30 jsme sestavili tabulku 5.1, kde v druhém sloupci uvádíme procentuální zlepšení kořenového síta oproti přímému výpočtu všech $\alpha(\mathrm{Im}(F_{c_1c_0}))$ a ve třetím sloupci procentuální vylepšení kořenového síta při použití optimalizace popsané v této podkapitole.

velikost meze hladkosti B	vylepšení při kořenovém sítu	vylepšení při optimalizaci
200	98.8201%	19.7171%
300	99.2037%	18.6914%
400	99.4026%	17.2372%
500	99.5227%	14.6334%
600	99.6023%	14.3513%
700	99.6631%	13.355%
800	99.7045%	12.7241%
900	99.7397%	12.1322%
1000	99.7666%	11.6998%
1100	99.7911%	11.2216%
1200	99.8058%	10.6103%
1300	99.8219%	9.93371%
1400	99.8339%	10.2139%
1500	99.8477%	9.55917%
1600	99.8562%	9.14526%
1700	99.8661%	8.98513%
1800	99.8728%	8.63169%
1900	99.8794%	8.5915%
2000	99.8855%	8.2541%

Tabulka 5.1: Procentuální vylepšení při použití kořenového síta oproti přímému výpočtu $\alpha(\mathrm{Im}(F_{c_1c_0}))$ a vylepšení při optimalizaci oproti klasickému kořenovému sítu.

5.3 Dvoufázové kořenové síto

Klasické i optimalizované kořenové síto operují se všemi zrotovanými polynomy, tudíž (jak jsme mohli nahlédnout z lemmat 27 a 29) vysoké meze rotačních koeficientů přispívají k celkovému času kořenového síta dominantním způsobem. Bai v článku [3] proto navrhuje komplexnější konstrukci – dvoufázové kořenové síto. V první fázi nejprve vybere polynomiálního kandidáta (případně více), u kterého se očekává, že bude mít dobré kořenové vlastnosti, a poté v druhé fázi aplikuje kořenové prosívání, ovšem pouze na mřížce definované tímto kandidátem nikoli v celé rotační oblasti, čímž dosahuje znatelného vylepšení.

5.3.1 První fáze – prohledávání stromu do hloubky

Zvolme P celočíselnou mez, buď $s = \pi(P)$ a mějme s malých prvočíselných mocnin $p_1^{e_1}, \ldots, p_s^{e_s}$ pro různá p_i. Pro pevný polynom $f(x)$ generovaný base-(m, p) metodou hledáme lineární rotaci $f_{c_1c_0}(x)$ s mnoha kořeny modulo zvolené malé prvočíselné mocniny $p_1^{e_1}, \ldots, p_s^{e_s}$. Pro malý součin $\prod_{i=1}^{s} p_i^{e_i}$ je možné využít optimalizované kořenové prosívání přes matici o velikosti $\left(\prod_{i=1}^{s} p_i^{e_i}\right)^2$. V rozsáhlejších případech najdeme nejprve s polynomů, které označíme $f_{c_1^{(i)}c_0^{(i)}p_i}(x)$ pro $1 \leq i \leq s$, kde každý bude mít mnoho kořenů modulo $p_i^{e_i}$ a následně pomocí Čínské zbytkové věty (ČZV) tyto polynomy zkombinujeme na výsledný $f_{c_1c_0}(x) \bmod \prod_{i=1}^{s} p_i^{e_i}$. Pro libovolné $1 \leq k \leq e_i$ bude mít $f_{c_1c_0}(x) \bmod p_i^k$ stejný počet kořenů jako jed-

notlivé $f_{c_1^{(i)}c_0^{(i)}p_i}(x) \bmod p_i^k$, proto lze předpokládat, že výsledný polynom $f_{c_1c_0}(x)$ bude mít mnoho kořenů modulo malé prvočíselné mocniny $p_1^{e_1}, \ldots, p_s^{e_s}$, jak jsme zamýšleli.

Pro nalezení každého z s polynomů $f_{c_1^{(i)}c_0^{(i)}p_i}(x)$ sestrojíme zvláštní p_i^2-ární strom. Kořenem bude vždy polynom $f(x)$. Mějme pevné i a pro přehlednost položme $p = p_i$ a $e = e_i$. V každé z k úrovní ($1 \leq k \leq e$) p^2-árního stromu budou všechny možné rotace $f(x)$ modulo p^k. Průchod tímto stromem s nalezením nejlepší rotace demonstruje algoritmus 5.3. Nejprve (řádky 1-9) nalezneme kořeny všech rotací $f_{c_1c_0}(x) \bmod p$ a rozdělíme je na jednoduché a násobné pomocí stejného principu, který jsme již popsali u optimalizovaného kořenového síta (lemma 28). Dále budeme pracovat jen s těmi rotacemi, které mají mnoho kořenů. Pro každou z těchto rotací chceme spočítat přínos p. Přínos jednoduchých kořenů známe hned (díky Henselovu lemmatu 9); aktualizujeme o hodnotu $\frac{\left|R_S^{c_1c_0}\right|p}{p^2-1}$ (řádek 11). U násobných kořenů však musíme provádět zdvihání. To provedeme pomocí rekurzivního průchodu p_i^2-árním stromem do hloubky, jak popisuje metoda LiftMult. Kroky 5 a 6 plynou z Henselova lemmatu 9 (2a) a principu kořenového prosívání (lemma 26). V každém listu pak známe počet kořenů a můžeme hodnotu přínosu aktualizovat o $\frac{\left|R_M^{c_1c_0}\right|p}{p^2-1}$.

Pro každé z s prvočísel p_i máme nyní nějakou rotaci $f_{c_1^{(i)}c_0^{(i)}p_i}(x)$ s mnoha kořeny modulo $p_i^{e_i}$. Pomocí některého z algoritmů využívajících ČZV vyřešíme soustavu

$$\begin{aligned} c_0 &\equiv c_0^{(1)} \mod p_1^{e_1} & \qquad c_1 &\equiv c_1^{(1)} \mod p_1^{e_1} \\ c_0 &\equiv c_0^{(2)} \mod p_2^{e_2} & \qquad c_1 &\equiv c_1^{(2)} \mod p_2^{e_2} \\ &\cdots & \qquad &\cdots \\ c_0 &\equiv c_0^{(s)} \mod p_s^{e_s} & \qquad c_1 &\equiv c_1^{(s)} \mod p_s^{e_s} \end{aligned}$$

a tím získáme hledanou rotaci $f_{c_1c_0}(x)$ s mnoha kořeny modulo $\prod_{i=1}^s p_i^{e_i}$, kterou předáme do druhé fáze.

5.3.2 Druhá fáze – prosívání na mříži

Z první fáze do druhé vstupuje rotace polynomu $f(x)$ definovaná dvojicí (c_0, c_1) a modul $M = \prod_{i=1}^s p_i^{e_i}$. V druhé fázi budeme kořenově prosívat, nikoliv však na celé oblasti $[-C_0, C_0] \times [-C_1, C_1] \subset \mathbb{Z}^2$, ale pouze na mřížce definované

$$\{(c_0 + \gamma M, c_1 + \beta M) \colon \gamma, \beta \in \mathbb{Z}\} \subset [-C_0, C_0] \times [-C_1, C_1].$$

Rotace z této mřížky jsou konstruovány tak, aby měly mnoho kořenů modulo M, proto očekáváme, že budou mít i dobré kořenové vlastnosti. Princip kořenového prosívání na mřížce je analogický myšlence lemmatu 26.

Lemma 31. *Buď $M = \prod_{i=1}^s p_i^{e_i}$, rotační koeficienty $c_0, c_1 \in \mathbb{Z}$ a uvažujme mřížku $\{(c_0+\gamma M, c_1+\beta M) \colon \gamma, \beta \in \mathbb{Z}\}$ definující množinu všech možných rotací polynomu $f(x)$, tedy $\tau_{\lambda(x)}(f) = f_{c_1+\beta M, c_0+\gamma M}(x)$ pro $\lambda(x) = (c_1 + \beta M)x + (c_0 + \gamma M)$. Pokud r je kořenem $f_{c_1+\beta M, c_0+\gamma M}(x) \bmod p^k$, kde p^k je mocnina prvočísla, pak r je také kořenem $f_{c_1+M(\beta+ip^k), c_0+M(\gamma+jp^k)}(x) \bmod p^k$ pro všechna $i, j \in \mathbb{Z}$.*

Algoritmus 5.3 Průchod p^2-árním stromem s rekurzí

input : polynomy f, g, aktuální prvočíslo p

output : rotace s maximálním přínosem p

1 : **for** $(r = 0 \dots p-1)$

2 : spočítej $\bar{c_1}$: $\bar{c_1} g^2(r) \equiv f(r)g'(r) - f'(r)g(r) \pmod{p}$

3 : **for** $(c_1 = 0 \dots p-1)$

4 : spočítej c_0: $f(r) + (c_1 t + c_0)g(r) \equiv 0 \pmod{p}$

5 : **if** $(c_1 \neq \bar{c_1})$ // r je jednoduchý kořen

6 : přidej r do $R_S^{c_1 c_0}$ // aktualizuj jednoduché kořeny rotace $f_{c_1 c_0}$

7 : **else** // r je násobný kořen

8 : přidej r do $R_M^{c_1 c_0}$ // aktualizuj násobné kořeny rotace $f_{c_1 c_0}$

9 : $A \leftarrow \{$vybrané rotace s mnoha kořeny$\}$

10 : **for** $((c_0,c_1) \in A)$

11 : aktualizuj $\text{cont}_p(\text{Im}(F_{c_1+ip, c_0+jp}))$

12 : $best_{c_1 c_0} \leftarrow$ LiftMult$(k, f_{c_1 c_0}, R_M^{c_1 c_0})$

1 : **if** $(k = e)$ // jsme v listu

2 : **return** aktualizované $\text{cont}_p(\text{Im}(F_{c_1 c_0}))$

3 : **else**

4 : **for** $(r \in R_M^{c_1 c_0})$

5 : **for** $((i,j)$: $(ir+j)g(r) + \dfrac{f_{c_1 c_0}(r)}{p^k} \equiv 0 \bmod p)$

6 : přidej $\{r + lp^k : l \in \mathbb{Z}_p\}$ do $R_M^{c_1 + ip^k, c_0 + jp^k}$

7 : **for** $(i = 0 \dots p-1)$

8 : **for** $(j = 0 \dots p-1)$

9 : $best_{i,j} \leftarrow$ LiftMult$(k+1, f_{c_1+ip^k, c_0+jp^k}, R_M^{c_1+ip^k, c_0+jp^k})$

10 : **return** $\max\limits_{i,j}(best_{ij})$

13 : **return** $\arg\max_{c_1 c_0}(best_{c_1 c_0})$

Důkaz. Dle definice rotace o $\lambda(x)$ je

$$f_{c_1+\beta M,c_0+\gamma M}(x) = f(x) + ((c_1+\beta M)x + (c_0+\gamma M))\, g(x)$$

pro příslušný lineární polynom $g(x)$. Pro

$$\lambda_{ij}(x) = \left(c_1 + M(\beta + ip^k)\right) x + \left(c_0 + M(\gamma + jp^k)\right)$$

z toho plyne

$$\begin{aligned}\tau_{\lambda_{ij}(x)}(f) &= f(x) + \left(\left(c_1 + M(\beta + ip^k)\right) x + \left(c_0 + M(\gamma + jp^k)\right)\right) g(x) = \\ &= f(x) + ((c_1+\gamma M)x + (c_0+\beta M))\, g(x) + (Mip^k x + Mjp^k)g(x) = \\ &= f_{c_1+\gamma M,c_0+\beta M}(x) + p^k \cdot (Mix + Mj)g(x).\end{aligned}$$

Dostáváme tvrzení věty, protože zřejmě

$$f_{c_1+\beta M,c_0+\gamma M}(x) \equiv f_{c_1+M(\beta+ip^k),c_0+M(\gamma+jp^k)}(x) \bmod p^k.$$

□

Kořenové síto na mřížce (alg. 5.4) pracuje na obdobném principu jako optimalizované kořenové síto (alg. 5.2). Pro pevné prvočíslo p a kořen r nalezneme příslušnou rotaci (u, v) a rotaci na mřížce dopočítáme vyřešením soustavy kongruencí

$$\begin{aligned} u &\equiv c_1 + \beta M \pmod p \\ v &\equiv c_0 + \gamma M \pmod p \end{aligned}$$

pro (γ, β). Podmínkou pro jednoznačné řešení je $p \nmid M$. Uvědomme si, že případ $p \mid M$ znamená, že příslušný kořen přispívá všem zrotovaným polynomům na mřížce. Pro účely porovnání polynomů lze tedy tento přínos vynechat a tyto případy přeskočit. Jelikož jsme v první fázi uvažovali všechna prvočísla menší než mez P, pak se ve druhé fázi můžeme omezit pouze na prvočísla ležící v intervalu $P \leq p \leq B$.

Pokud bychom na rekurzi popsanou v metodě LiftMult neaplikovali žádnou heuristiku ořezávající větve, u nichž očekáváme, že nepřinesou dobré výsledky, pak bude tato metoda v nejhorším případě stejně efektivní jako přímé zdvihání násobných kořenů, jak je využíváno v optimalizovaném kořenovém sítu (alg. 5.2). Proto Bai navrhuje pro zdvihání násobných kořenů při prosívání na mřížce recyklovat rekurzi použitou v první fázi, kdy hodnoty přínosu nebudeme přímo vracet, ale pouze aktualizovat. Po skončení celého algoritmu vybereme nejlepší rotaci z mřížky, případně na několik nejlepších polynomů aplikujeme zkušební prosívání.

Algoritmus 5.4 Kořenové síto na mřížce

input : polynomy $f(x), g(x)$, nejlepší rotace (c_0, c_1), meze rotační oblasti C_0, C_1, mez hladkosti B, $M = \prod_{i=1}^{s} p_i^{e_i}$

output : aproximace $\mathrm{cont}_p(\mathrm{Im}(F_{c_1 c_0}))$ v M-mříži rotační oblasti $[-C_0, C_0] \times [-C_1, C_1]$

1 : **for** $(p \leq B, p$ prvočíslo$, p \nmid M)$
2 : **for** $(r = 0 \ldots p-1)$
3 : spočítej $\bar{u}$: $\bar{u}g^2(r) \equiv f(r)g'(r) - f'(r)g(r) \pmod{p}$
4 : **for** $(u = 0 \ldots p-1)$
5 : spočítej v z $f(r) + (ur + v)g(r) \equiv 0 \pmod{p}$
6 : najdi (γ, β) z $u \equiv c_1 + \beta M \bmod p$ a zároveň $v \equiv c_0 + \gamma M \bmod p$
7 : **if** $(\bar{u} \neq u)$ // r je jednoduchý kořen
8 : aktualizuj $\mathrm{cont}_p(\mathrm{Im}(F_{c_1+M(\beta+ip), c_0+M(\gamma+jp)}))$
9 : **else** // r je násobný kořen
10 : LiftMult$(2, f_{c_1+\beta M, c_0+\gamma M}, \{r\})$
11 : // stejná rekurze jako v alg. 5.3, místo vracení nejlepších hodnot pouze aktualizujeme

Kapitola 6

Implementační aspekty generování polynomů

V rámci projektu NFS vedeného na Katedře algebry MFF UK jsme spolupracovali na implementaci Kleinjungova algoritmu pro generování polynomů (popsaného v podkapitole 3.2). V následujících podkapitolách nejprve stručně popíšeme celou implementaci, podrobněji se budeme věnovat konkrétnímu řešení některých aspektů, které se v rámci vývoje ukázaly jako ne přímo odvoditelné z návrhu algoritmu dle T. Kleinjunga v článku [16]. Závěr kapitoly bude patřit experimentu, který jsme provedli na reálných datech získaných z implementace a který nám pomohl ověřit některé heuristické úvahy z kapitoly 4.1.1.

6.1 Implementace Kleinjungova algoritmu

Kleinjungův algoritmus (kap. 3.2) jsme naimplementovali jako samostatnou třídu `CKleinjungPolySelPhase` projektu NFS, vylepšení a hodnocení polynomů bylo odděleno do třídy `CPolynomialImprovement`. Zdrojový kód je psán v jazyce C++, kompilace probíhala překladačem GCC (verze 4.9.2 a 4.8.2) v linuxovém prostředí Debian 8.x (Jessie), verze jádra 3.16; a Kubuntu 14.04 (TrustyTahr), verze jádra 4.2.

Implementaci Kleinjungova algoritmu se podařilo vyladit do té míry, že pro rozkládaná čísla větší než 130 decimálních cifer generuje výrazně lepší polynomy než dříve používané klasické generování polynomů base-m metodou. Hodnocení vygenerovaných polynomů je navíc srovnatelné s hodnocením polynomů z implementace CADO-NFS [25], která je současnou nejlepší implementací faktorizace pomocí obecného číselného síta.

6.1.1 Výpočet base-(m,p) rozvoje

Při výpočtu base-(m, p) rozkladu čísla N rekurzivním způsobem popsaným v algoritmu 3.1 počítáme v každém kroku pro i sestupně od $d - 1$ dvě hodnoty

$r_i = \frac{r_{i+1} - a_{i+1} m^{i+1}}{p}$ a

$a_i = \frac{r_i}{m^i} + \delta_i$ pro $0 \leq \delta_i < p$ takové, že platí $r_i \equiv a_i m^i \pmod{p}$.

Existenci celočíselného podílu pro r_i v každém kroku jsme již okomentovali v důkazu věty 13. Podíl $\frac{r_i}{m^i}$ ovšem celočíselný být nemusí, takže hledáme $\delta_i \in \mathbb{Q}$. Celočíselné koeficienty a_i máme tedy teoreticky vyjádřeny jako součet dvou racionálních čísel, což znemožňuje přímočarou implementaci kvůli nedostatku přesnosti.

V naší implementaci jsme zvolili následující postup: Spočítejme nejprve hodnotu $\bar{a}_i = \lceil \frac{r_i}{m^i} \rceil$, hledaný koeficient a_i musí ležet v množině $\{\bar{a}_i, \bar{a}_i{+}1, \ldots, \bar{a}_i{+}p{-}1\}$ z omezení pro δ_i. Dále spočítejme $c \equiv \frac{r_i}{m^i} \bmod p$. Takové c musí existovat, protože pracujeme s nesoudělnými m a p, tudíž m^i má inverz modulo p. Hledaný koeficient pak lze spočítat jednoznačně jako

$$a_i = \bar{a}_i + ((c - \bar{a}_i) \bmod p).$$

6.1.2 Výpočet Dickmanovy funkce

Dickmanova funkce (definice 8) se ukázala být stěžejním kvantifikátorem pro hodnocení polynomů. Ačkoli v průběhu generování polynomů jsme výpočet Dickmanovy funkce na některých místech z výpočetních důvodů obešli různými zjednodušeními (viz kap. 4.1.1), několik nejlepších kandidátů je dobré na závěr porovnat plným kombinovaným hodnocením (4.23). Proto musíme umět spočítat hodnotu Dickmanovy funkce ρ prakticky.

Existují numerické metody pro výpočet Dickmanovy funkce, nicméně není znám žádný vzorec v uzavřeném tvaru využívající pouze elementární funkce. V naší implementaci jsme však pro výpočet $\rho(u)$ nezvolili numerické integrování, ale efektivnější metodu Pattersona a Rumseye popsanou až v článku Bacha a Peralty [1], která využívá Taylorovy řady.

K důkazu následujícího lemmatu využijeme vlastnost funkce $\rho(u)$, kterou ukazuje Dickman [10].

$$\rho(u) = \begin{cases} 1 & \text{pro } 0 \leq u \leq 1; \\ \frac{1}{u} \int\limits_{u-1}^{u} \rho(t)\,\mathrm{d}t & \text{jinak.} \end{cases}$$

Lemma 32. *Buď $u > 1$, pak pro Dickmanovu funkci $\rho(u)$ platí vztah*

$$\rho'(u) = \frac{-\rho(u-1)}{u}.$$

Důkaz. Pro $u > 1$ platí druhá výše uvedená vlastnost

$$\rho(u) = \frac{1}{u} \int\limits_{u-1}^{u} \rho(t)\,\mathrm{d}t.$$

Zderivujeme obě strany rovnosti

$$\begin{aligned}\rho'(u) &= \frac{\partial}{\partial u}\left(\frac{\int_{u-1}^{u}\rho(t)\,\mathrm{d}t)}{u}\right) = \\ &= -\frac{1}{u^2}\int_{u-1}^{u}\rho(t)\,\mathrm{d}t + \frac{1}{u}\left(\rho(u)-\rho(u-1)\right) = \\ &= -\frac{1}{u}\cdot\rho(u) + \frac{1}{u}\cdot\rho(u) - \frac{\rho(u-1)}{u} = \\ &= -\frac{\rho(u-1)}{u}.\end{aligned}$$

□

Dokázaný vztah je diferenciální rovnicí se zpožděním a implikuje, že $\rho(u)$ náleží to třídy C^k na intervalu $[k,\infty)$ pro $k \geq 1$ (neboli všechny parciální derivace $\rho(u)$ až do řádu k jsou na uvedeném intervalu spojité) a je navíc po částech analytická (viz [1]).

Věta 33 (Taylorova věta s Peanovým tvarem zbytku)**.** *Buď $a \in \mathbb{R}$ a mějme reálnou funkci f, která má v bodě a vlastní k-tou derivaci pro $k \in \mathbb{N}$. Pak existuje právě jeden polynom $T_k(x)$ stupně nejvýše k takový, že*

$$f(x) - T_k(x) = o\left((x-a)^k\right) \text{ pro } x \to a.$$

Z této věty plyne, že pro $u > 1$ lze hodnotu $\rho(u)$ aproximovat polynomem, který budeme dále pro $0 < \epsilon \leq 1$ značit

$$\rho(u) \approx \sum_{i=0}^{k} c_i^{(k)}\epsilon^i,$$

kde $u = k + \epsilon$, a tudíž $c_0^{(k)} = \rho(k)$. Nyní ve stručnosti shrneme, jak lze takové koeficienty $c_i^{(k)}$ spočítat.

Na $1 < u \leq 2$ se podle Bacha a Peralty [1] $\rho(u)$ chová jako $1 - \ln(u)$. Je tedy

$$\begin{aligned}\rho(2-\epsilon) &= 1 - \ln(2-\epsilon) = 1 - \ln(2) - \ln\left(1 - \frac{\epsilon}{2}\right) = \\ &= 1 - \ln(2) + \sum_{i=1}^{\infty}\frac{\epsilon^i}{i2^i},\end{aligned}$$

přičemž poslední rovnost plyne z rozvoje řady

$$\ln(1+x) = \sum_{i=1}^{\infty}\frac{(-1)^{i+1}x^i}{i}.$$

Buď $u > 2$ a předpokládejme, že známe hodnotu $\rho((k-1)-\epsilon) = \sum_{j=0}^{\infty} c_j^{(k-1)}\epsilon^j$. Derivaci $\rho'(k-\epsilon)$ umíme vyjádřit dvěma způsoby

$$\rho'(k-\epsilon) = \frac{-\rho(k-1-\epsilon)}{k-\epsilon} = -\frac{1}{k}\cdot\frac{\sum_{j=0}^{\infty} c_j^{(k-1)}\epsilon^j}{1-\frac{\epsilon}{k}} = -\frac{1}{k}\cdot\sum_{j=0}^{\infty} c_j^{(k-1)}\epsilon^j\sum_{i=0}^{\infty}\left(\frac{\epsilon}{k}\right)^i,$$

$$\rho'(k-\epsilon) = \left(\sum_{i=0}^{\infty} c_i^{(k)}\epsilon^i\right)' = \sum_{i=1}^{\infty} i c_i^{(k)}\epsilon^{i-1}.$$

Porovnáním těchto dvou výrazů a srovnáním sčítanců se stejnou mocninou ϵ lze nalézt vyjádření pro koeficienty Taylorova polynomu $c_i^{(k)}$ pro $i > 0$. V případě $c_0^{(k)}$ platí z definice

$$kc_0^{(k)} = k\rho(k) = \int_0^1 \rho(k-\epsilon)\,\mathrm{d}\epsilon = \int_0^1 \sum_{j=0}^{\infty} c_j^{(k)}\epsilon^j\,\mathrm{d}\epsilon = \sum_{j=0}^{\infty} c_j^{(k)}\frac{1}{j+1}.$$

Odvodili jsme vzorce pro výpočet všech potřebných koeficientů k aproximaci $\rho(u)$ pro $u > 1$. Připomeňme, že pro $u \in [0,1]$ je $\rho(u) = 1$ z definice.

$$\begin{aligned}
k = 0 \qquad & c_i^{(0)} = 1 \quad \text{pro } i \geq 0 \\
k = 1 \qquad & c_0^{(1)} = 1 - \log 2 \\
& c_i^{(1)} = \frac{1}{i2^i} \quad \text{pro } i > 0 \\
k > 1 \qquad & c_i^{(k)} = \sum_{j=0}^{i-1} \frac{c_j^{(k-1)}}{ik^{i-j}} \quad \text{pro } i > 0 \\
& c_0^{(k)} = \frac{1}{k-1}\sum_{j=1}^{\infty}\frac{c_j^{(k)}}{j+1}
\end{aligned}$$

Tato reprezentace je z implementačního hlediska velice efektivní, jelikož koeficienty $c_i^{(k)}$ lze předpočítat dopředu, přičemž je možné využít jejich rekurentní závislosti. V momentě výpočtu $\rho(u)$ je nejdříve určeno k, pro které $u \in [k-1, k)$, a následně vyhodnocena příslušná Taylorova řada s využitím předpočítaných koeficientů $c_i^{(k)}$.

Lze navíc ukázat, že platí $0 \leq c_i^{(k)} \leq \frac{1}{2^i}$, tudíž $m+1$ členů posloupnosti aproximuje $\rho(u)$ na intervalu $[k-1, k)$ s absolutní chybou menší než 2^{-m}. Empiricky zjistili autoři článku [1], že v oblasti $0 \leq u \leq 20$ stačí spočítání 55 koeficientů pro vyčíslení $\rho(u)$ s relativní chybou menší než 10^{-17}, což odpovídá přesnosti typu `double` dle standardu IEEE [14].

6.1.3 Vylepšení metody největšího spádu

V první části optimalizace výtěžnosti se snažíme optimalizovat vlastnost velikosti polynomu $f = \sum_{i=0}^{d} a_i x^i$ pomocí úprav translace a rotace. Připomeňme formální značení, buď $\tilde{f} = \tau_{\lambda(x)}(\tau_t(f))$ polynom upravený pomocí zobrazení translace o t a rotace o $\lambda(x) = \sum_{i=0}^{2} c_i x^i$. Homogenizaci $\tilde{f}$ značíme $\tilde{F}$. Koeficienty homogenního polynomu $\tilde{F}$ jsou parametrizovány proměnnými t, c_0, c_1 a c_2.

Chceme minimalizovat hodnotu integrálu

$$\iint_{\mathcal{S}''} \tilde{F}^2(x,y)\,\mathrm{d}x\,\mathrm{d}y$$

přes $\mathcal{S}'' = [-\sqrt{s}, \sqrt{s}] \times [-\frac{1}{\sqrt{s}}, \frac{1}{\sqrt{s}}]$, referenční „jednotkovou“ zkreslenou prosívací

oblast. Snažíme se tedy minimalizovat hodnotu

$$F_s = \int\limits_{-\frac{1}{\sqrt{s}}}^{\frac{1}{\sqrt{s}}} \int\limits_{-\sqrt{s}}^{\sqrt{s}} \tilde{F}^2(x,y)\,\mathrm{d}x\,\mathrm{d}y$$

vzhledem k proměnným s, t a c_i pro všechna $0 \leq i \leq 2$. Murphy [21] navrhuje využití metody největšího spádu, Jedlička v článku [15] ukazuje, že přímá implementace metody největšího spádu je pro tento případ nevhodná a navrhuje následující vylepšení. Uvědomme si, že F_s je polynomem v proměnných s, t a c_i a navíc lze tyto proměnné separovat na dvě v principu nezávislé skupiny: c_i pro $i \in \{0, 1, 2\}$ a s, t. Polynom F_s lze proto zapsat ve tvaru

$$F_s = \int\limits_{-\frac{1}{\sqrt{s}}}^{\frac{1}{\sqrt{s}}} \int\limits_{-\sqrt{s}}^{\sqrt{s}} \tilde{F}^2(x,y)\,\mathrm{d}x\,\mathrm{d}y = B(t,s) + \sum_{i=0}^{2} B_i(t,s)c_i + \sum_{i=0}^{2}\sum_{j=i}^{2} B_{ij}(t,s)c_ic_j,$$

kde B, B_i a B_{ij} jsou funkce pouze v proměnných s, t. Nyní je možné minimalizaci provádět ve dvou krocích pro oddělené skupiny koeficientů.

Nejdříve zvolíme libovolná ale pevná s a t a spočítáme příslušné funkční hodnoty $B(s,t) = b$, $B_i(s,t) = b_i$ a $B_{ij}(s,t) = b_{ij}$. Dle článku [15] je grafem této funkce kvadrika v $\mathbb{R}^4$, konkrétně se zřejmě jedná o paraboloid, a minimum v proměnných c_i lze tudíž spočítat pomocí parciálních derivací a Gaussovou eliminací. V druhém kroku fixujeme nalezené minimum v proměnných c_i a pomocí klasické metody největšího spádu hledáme minimum pro t a s. Střídáním uvedených dvou kroků postupně konvergujeme k hodnotě minima.

6.2 Zjednodušení hodnocení polynomu

V kapitole 4.1.1 jsme popsali heuristické argumenty vedoucí ke zjednodušení teoretického vzorce pro hodnocení vlastnosti velikosti polynomů využívajícího Dickmanovu funkci pro výpočet pravděpodobnosti hladkosti polynomiálních hodnot $|F(x,y)|$ a $|G(x,y)|$

$$\frac{6}{\pi^2} \cdot \iint\limits_{\mathcal{M}'} \rho\left(\frac{\ln\left|\tilde{F}(x,y)\right|}{\ln B}\right) \rho\left(\frac{\ln\left|\tilde{G}(x,y)\right|}{\ln B}\right) \mathrm{d}x\,\mathrm{d}y$$

na výpočetně efektivnější vzorec založený na L^2-normě nelineárního polynomu $F(x,y)$

$$\ln\left(\iint\limits_{\mathcal{M}'} F^2(x,y)\,\mathrm{d}x\,\mathrm{d}y\right).$$

Za účelem ověření, zda výše popsané zjednodušení nevnáší do hodnocení vlastnosti velikosti příliš velké chyby, jsme provedli následující empirický test. Pro pět vstupních N různé délky, jak je uvedeno v tabulce 6.1, jsme z běhu implementovaného algoritmu číselného síta náhodně zvolili jeden z hodnocených polynomů. Velikost N je volena tak, aby stupeň polynomu byl různý. (Nejmenší N o 36 desetinných cifrách jsme zvolili kvůli zjištění, zda budou naše pozorování konzistentní i v extrémním případě.)

stupeň f	počet decimálních cifer N
4	36
4	80
5	124
6	145
7	173

Tabulka 6.1: Volba různých N pro test zjednodušení hodnocení vlastností polynomů.

Na tento polynom jsme aplikovali úpravy pomocí translace a rotace, spočítali jsme 100 posunutých a zrotovaných polynomů pomocí náhodných hodnot t (translace) a c_0, c_1, c_2 (kvadratická rotace) z různých intervalů (viz tab. 6.2).

pořadí polynomu	translace	rotace
1-10	$[-20, 20]$	$[-10, 10]$
11-20	$[-20, 20]$	$[-10^2, 10^2]$
21-30	$[-20, 20]$	$[-10^3, 10^3]$
31-40	$[-20, 20]$	$[-10^4, 10^4]$
41-50	$[-20, 20]$	$[-10^5, 10^5]$
51-60	$[-10^5, 10^5]$	$[-10, 10]$
61-70	$[-10^5, 10^5]$	$[-10^2, 10^2]$
71-80	$[-10^5, 10^5]$	$[-10^3, 10^3]$
81-90	$[-10^5, 10^5]$	$[-10^4, 10^4]$
91-100	$[-10^5, 10^5]$	$[-10^5, 10^5]$

Tabulka 6.2: Intervaly pro volbu parametrů translace a rotace pro test zjednodušení hodnocení vlastností polynomů.

Každý vzorek 100 polynomů jsme poté seřadili podle hodnocení na prosívací oblasti $\mathcal{M}' = [-10^5, 10^5] \times [1, 10^5]$ těmito kritérii:

- „plná Dickmanova norma“ $\iint\limits_{\mathcal{M}'} \rho\left(\frac{\ln\left|\tilde{F}(x,y)\right|}{\ln 1000}\right) \rho\left(\frac{\ln\left|\tilde{G}(x,y)\right|}{\ln 1000}\right) \mathrm{d}x\,\mathrm{d}y$ (od největšího);
- „Dickmanova norma“ $\iint\limits_{\mathcal{M}'} \rho\left(\frac{\ln\left|\tilde{F}(x,y)\right|}{\ln 1000}\right) \mathrm{d}x\,\mathrm{d}y$ (od největšího);
- L^2-norma $\frac{1}{2}\ln\left(\iint\limits_{\mathcal{M}'} F^2(x,y)\,\mathrm{d}x\,\mathrm{d}y\right)$ (od nejmenšího);
- `Sort-Abs-Koef` – řazením podle velikosti koeficientů s upřednostněním střídání znamének (od nejmenšího).

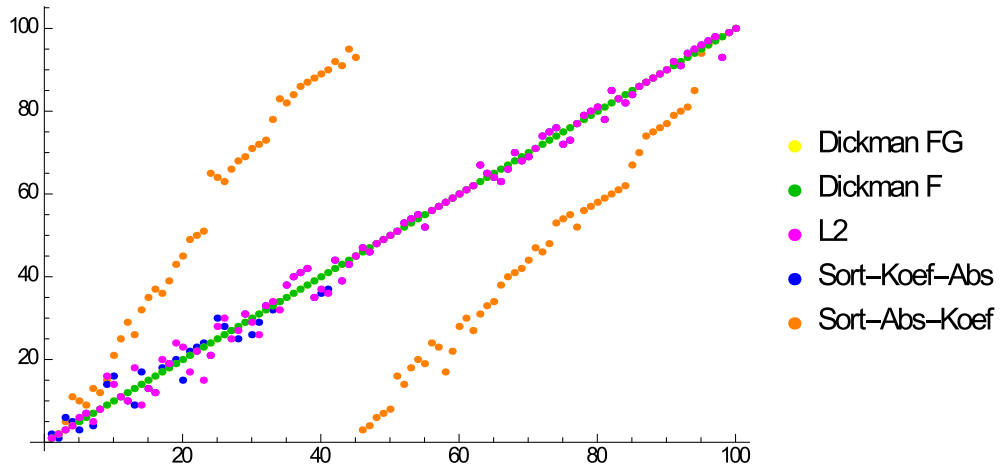
Poslední uvedené řazení se ukázalo jako velice nepřesné, proto jsme zařadili ještě jeho adaptaci

- `Sort-Koef-Abs` – řazení podle velikosti koeficientů, při shodě upřednostňuje střídání znamének (od nejmenšího).

Celý test jsme opakovali desetkrát pro každou volbu N. Získali jsme tedy celkem 50 sad o 100 polynomech, na které jsme aplikovali všech pět výše uvedených kritérií pro řazení.

Nejprve jsme zjišťovali chybu, kterou přináší vynechání lineárního polynomu $g(x)$ z výpočtu. Ukázalo se, že pro $\deg(f) \geq 5$ došlo pouze v jednom případě k prohození dvou sousedních polynomů (nejednalo se o polynomy z desíti nejlepších), jinak byly všechny polynomy řazeny totožně. V případě polynomů stupně 4 byla situace výrazně jiná, totožně byla řazena jen přibližně polovina polynomů v případě N o 80 decimálních cifrách a 35-40% pro N o 36 decimálních cifrách. Záměna dvou sousedních polynomů proběhla v obou případech u zhruba čtvrtiny polynomů. Co se však týče odhalení několika nejlepších polynomů, můžeme konstatovat, že zanedbání lineárního polynomu z výpočtu má vliv zcela minimální (viz tab. 6.3).

Výsledky řazení dalšími kritérii již nevykazovaly žádné abnormality vzhledem ke stupni nelineárního polynomu a (až na řazení s upřednostněním znaménka) výrazně korelují s řazením dle definice. Výsledky jsme vizualizovali jako bodový graf s referenčním řazením podle plné Dickmanovy normy. Jednotlivé barvy jsou podle legendy přiřazeny příslušným kritériím. Bod se souřadnicemi (i, j) značí, že i-tý polynom podle plné Dickmanovy normy byl podle barvy příslušným kritériem zařazen jako j-tý v pořadí od nejlepšího. Body na diagonále tedy přísluší řazení dle plné Dickmanovy normy. Pro všechny sady polynomů stupně $d > 4$ vypadaly výsledky zcela analogicky jako pro příklad na obr. 6.1. Procentuální srovnání výsledků vzhledem k řazení podle plné Dickmanovy normy přes všech 50 testovacích sad uvádíme v tabulce 6.3.



Obrázek 6.1: Výsledky řazení různými kritérii hodnocení velikosti polynomů.

Náš test tedy uzavíráme konstatováním, že popsané zjednodušení výpočtu hodnocení vlastnosti velikosti z Dickmanovy na L^2-normu nevnáší do hodnocení žádnou zásadní chybu. Dokonce se ukazuje, že by postačovalo pouhé řazení porovnáváním velikosti koeficientů s upřednostněním střídání znamének v případě rovnosti koeficientů. Výhodou řazení by jistě bylo znatelné zrychlení výpočtu při vhodně optimalizované implementaci řazení. Nevýhodou však je absence globálního kvantifikátoru a tudíž nemožnost porovnat například nejlepší polynomy z různých běhů a také nemožnost kombinace hodnocení s kořenovými vlastnostmi, jak ji popisujeme v kap. 4.2. Možná by však jistě byla alternativa kombinující

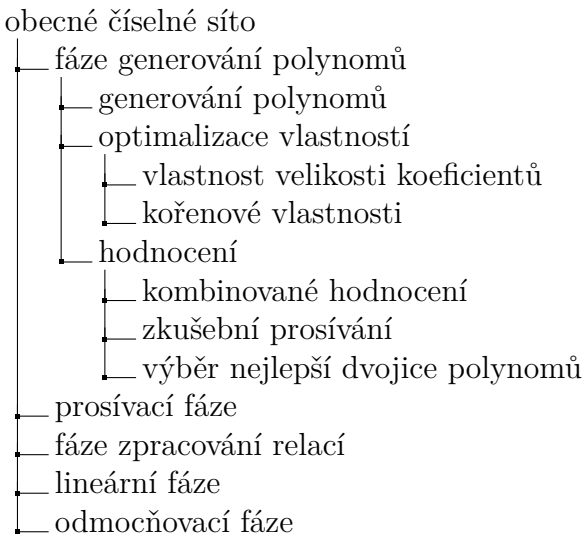
nejdříve rychlé řazení podle velikosti koeficientů a následné dopočítání L^2-normy pro několik nejlepších polynomů.

	Dickman F	L^2-norma	`Sort-Koef-Abs`	`Sort-Abs-Koef`
stupeň $d = 4$				
shoda v nejlepším	65%	70%	72%	34%
nejlepší	100%	96%	98%	60%
v prvních pěti shodná nejlepší trojice	70%	92%	84%	40%
stupeň $d = 5$				
shoda v nejlepším	100%	74%	70%	40%
nejlepší	100%	98%	96%	62%
v prvních pěti shodná nejlepší trojice	100%	94%	88%	46%
stupeň $d = 6$				
shoda v nejlepším	100%	70%	70%	36%
nejlepší	100%	100%	98%	70%
v prvních pěti shodná nejlepší trojice	100%	92%	90%	50%
stupeň $d = 7$				
shoda v nejlepším	100%	72%	68%	42%
nejlepší	100%	98%	100%	64%
v prvních pěti shodná nejlepší trojice	100%	90%	90%	48%

Tabulka 6.3: Procentuální srovnání řazení různými kritérii hodnocení velikosti polynomu vzhledem k Dickmanově normě.

Závěr

Práce přináší podrobný přehled první fáze algoritmu číselného síta, kterou lze rozdělit do několika kroků. Posloupnost těchto kroků v kontextu celého číselného síta znázorňuje následující diagram.



Jednotlivé kroky lze provádět celou řadou různých metod, které jsou předmětem mnoha akademických prací a článků. Vybrané postupy jsme v práci popsali a nyní je shrneme.

Pro samotné generování polynomů jsme představili dvě metody: base-m metodu (kap. 3.1) a Kleinjungův algoritmus (kap. 3.2).

Vlastnostem generovaných polynomů a jejich optimalizacím jsme věnovali velkou část práce. V kapitole 4.1.1 jsme ukázali, proč vlastnost velikosti hodnotíme pomocí L^2-normy, v kapitole 4.1.3 jsme představili kořenové vlastnosti a jejich kvantifikaci pomocí parametru α. Při optimalizaci (kap. 4.3) nejprve pomocí translace a rotace minimalizujeme (např. vylepšenou metodou největšího spádu, viz. 6.1.3) L^2-normu, čímž dostáváme polynom s dobrými velikostními vlastnostmi. Poté již pouze pomocí rotace minimalizujeme některým z kořenových sít (kap. 5) hodnotu α tak, abychom vylepšili kořenové vlastnosti, ale neporušili již optimalizovanou velikost.

Optimalizované polynomy hodnotíme Murphyho kombinovaným hodnocením popsaným v kapitole 4.2 a vybíráme nejlepší polynom. Volitelně lze vybrat několik polynomů s nejvyšším hodnocením a nejlepšího kandidáta určit pomocí zkušebního prosívání, jehož princip jsme stručně nastínili v úvodu kapitoly 4.

Počet polynomů vygenerovaných Kleinjungovým algoritmem závisí na počátečním nastavení vstupních parametrů, především na horních mezích prvních tří koeficientů a počtu prvočinitelů parametru p. Je třeba najít rovnováhu mezi časem

nutným pro generování a kvalitou nalezených polynomů, neboť rychlejší průběh fáze bývá obvykle na úkor kvality nalezených polynomů. Relativně dobrých výsledků jsme na osobním počítači většinou dosáhli, pokud se počet vygenerovaných polynomů pohyboval v řádu tisíců, maximálně desetitisíců. Klíčovým faktorem se ukázalo být nastavení mezí pro optimalizaci. Z výpočetního hlediska je žádoucí, aby optimalizace provedly redukci na maximálně několik desítek polynomů, které jsme schopni zpracovat v kombinovaném hodnocení. Případné zkušební prosívání je možné použít pouze na několik nejlepších polynomů.

Mnoho myšlenek a tvrzení skrývajících se za postupy generování polynomů má svůj původ v empirické zkušenosti a často bývají dalšími autory bez podrobnějšího vysvětlení přebírány i pro nové metody, které však již disponují jinými parametry než metody původní. Přínosem práce je doplnění důkazů k tvrzením, jež jsou založena na rigorózní teorii především v kapitole 3 o Kleinjungově algoritmu a v kapitole 5 o optimalizaci kořenových vlastností. Původní je také pojem přínosového schématu, který umožnil zjednodušit řadu úvah o přínosu prvočísel a podrobně odvodit aproximační vzorce v lemmatu 25. Někdy však snaha o vysvětlení jediné volby parametru může vyústit v několik stran úvah vysvětlujících téma, které s danou problematikou na první pohled téměř nesouvisí, jak tomu bylo u problému volby stupně nelineárního polynomu v kapitole 4.1.2. Poznatky získané při studiu teoretických principů jsme také aplikovali prakticky při programování Kleinjungova algoritmu s optimalizací a hodnocením polynomů. Naši implementaci jsme pak využili pro empirické ověření některých heuristik, které jsme v teoretické části popsali. Především jsme se zaměřili na experimentální ověření, že zjednodušení při hodnocení vlastnosti velikosti popsaná v kapitole 4.1.1 vedou k uspokojivým výsledkům (viz kap. 6.2).

Ukazuje se, že ač některé postupy využívané v číselném sítu nestojí na pevném teoretickém základu, přesto dávají velice dobré praktické výsledky. V tomto směru musíme plně souhlasit s Carlem Pomerancem [9]: „*The numbers we are trying to factor don't seem to mind our lack of rigor, they get factored anyway.*“

Literatura

[1] Bach, E. a Peralta, R. (1992). Asymptotic semi-smoothness probabilities. Technical Report TR 1115, University of Wisconsin, Computer Sciences Department (Madison, WI US). URL http://research.cs.wisc.edu/techreports/1992/TR1115.pdf.

[2] Bai, S. (2011). *Polynomial Selection for the Number Field Sieve*. PhD thesis, Australian National University. URL http://maths-people.anu.edu.au/~brent/pd/Bai-thesis.pdf.

[3] Bai, S., Brent, R. a Thomé, E. (2015). Root optimization of polynomials in the number field sieve. *Mathematics of Computation*, **84**(295), 2447–2457. URL http://maths-people.anu.edu.au/~bai/paper/rs.pdf.

[4] Bai, S., Bouvier, C., Kruppa, A. a Zimmermann, P. (2016). Better polynomials for GNFS. *Mathematics of Computation*, **85**(298), 861–873. URL http://maths-people.anu.edu.au/~bai/paper/sopt.pdf.

[5] Bruijn, N. D. (1951). On The Number of Positive Integers $\leq$ x and Free of Prime Factors $>$ y. *Indagationes Mathematicae (Proceedings)*, **54**, 50 – 60. ISSN 1385-7258.

[6] Buhler, J., Lenstra, H.W., J. a Pomerance, C. (1993). Factoring integers with the number field sieve. In Lenstra, A. K. a Lenstra, Hendrik W., J., editors, *The development of the number field sieve*, volume 1554 of *Lecture Notes in Mathematics*, pages 50–94. Springer Berlin Heidelberg. ISBN 978-3-540-57013-4. doi: 10.1007/BFb0091539.

[7] Canfield, E. R., Erdös, P. a Pomerance, C. (1983). On a problem of Oppenheim concerning „factorisatio numerorum“. *Journal of Number Theory*, **17**(1), 1–28. URL http://ad.bolyai.hu/~p_erdos/1983-11.pdf.

[8] Coppersmith, D. (1994). Solving homogeneous linear equations over GF(2) via block Wiedemann algorithm. *Mathematics of Computation*, **62** (205), 333–350. URL http://www.ams.org/journals/mcom/1994-62-205/S0025-5718-1994-1192970-7/S0025-5718-1994-1192970-7.pdf.

[9] Crandall, R. a Pomerance, C. (2006). *Prime Numbers: A Computational Perspective*. Lecture notes in statistics. Springer New York. ISBN 9780387289793.

[10] Dickman, K. (1930). *On the Frequency of Numbers Containing Prime Factors of a Certain Relative Magnitude*. Arkiv för matematik, astronomi och fysik. Almqvist & Wiksell.

[11] Gathen, J. v. z. a Gerhard, J. (2013). *Modern Computer Algebra*. Cambridge University Press, New York, NY, USA, 3rd edition. ISBN 1107039037, 9781107039032.

[12] Gower, J. E. (2003). Rotations and translations of number field sieve polynomials. In *Advances in Cryptology-ASIACRYPT 2003*, pages 302–310. Springer. URL `http://iacr.org/archive/asiacrypt2003/07_Session07/04_021/28940055.pdf`.

[13] Hensel, K. (1918). Eine neue Theorie der algebraischen Zahlen. *Mathematische Zeitschrift*, **2**(3-4), 433–452. ISSN 0025-5874.

[14] IEEE Task P754 (1985). *ANSI/IEEE 754-1985, Standard for Binary Floating-Point Arithmetic*. Institute of Electrical and Electronics Engineers, New York, NY, USA. ISBN 1-55937-653-8.

[15] Jedlička, P. (2010). Integral Minimisation Improvement for Murphy's Polynomial Selection Algorithm. *Analele Stiint. Univ. Ovidius C.*, **18**(2), 125–130. URL `http://www.emis.ams.org/journals/ASUO/mathematics/pdf21/12.pdf`.

[16] Kleinjung, T. (2006). On polynomial selection for the general number field sieve. *Mathematics of Computation*, **75**(256), 2037–2047. ISSN 0025-5718; 1088-6842/e. URL `http://www.ams.org/journals/mcom/2006-75-256/S0025-5718-06-01870-9/S0025-5718-06-01870-9.pdf`.

[17] Lanczos, C. (1950). An Iteration Method for the Solution of the Eigenvalue Problem of Linear Differential and Integral Operators. *Journal of Research of the National Bureau of Standards*, **45**(4). URL `http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.691.9952&rep=rep1&type=pdf`.

[18] Lenstra, A. K. a Hendrik W. Lenstra, J., editors (1993). *The development of the number field sieve*, volume 1554 of *Lecture Notes in Mathematics*. Springer-Verlag, Berlin. ISBN 3–540–57013–6.

[19] Monico, C. (2005). GGNFS – A Number Field Sieve implementation. URL `http://www.math.ttu.edu/~cmonico/software/ggnfs/index.html`.

[20] Montgomery, P. L. (1994). Square roots of products of algebraic numbers. In *Mathematics of Computation 1943–1993: a half-century of computational mathematics*, pages 567–571. American Mathematical Society.

[21] Murphy, B. A. (1999). *Polynomial Selection for the Number Field Sieve Integer Factorisation Algorithm*. PhD thesis, Australian National University. URL `http://maths.anu.edu.au/~brent/ftp/Murphy-thesis.ps.gz`.

[22] Perůtka, L. (2009). Hledání optimálních strategií číselného síta. Master's thesis, Univerzita Karlova v Praze.

[23] Pollard, J. (1993). Factoring with cubic integers. In Lenstra, A. K. a Lenstra, Hendrik W., J., editors, *The development of the number field sieve*, volume 1554 of *Lecture Notes in Mathematics*, pages 4–10. Springer Berlin Heidelberg. ISBN 978-3-540-57013-4. doi: 10.1007/BFb0091536.

[24] Skoková, A. (2015). Podpůrné algoritmy číselného síta. Master's thesis, Univerzita Karlova v Praze.

[25] The CADO-NFS Development Team (2016). CADO-NFS, An Implementation of the Number Field Sieve Algorithm. URL http://cado-nfs.gforge.inria.fr/. Release 2.2.0.

[26] Wiedemann, D. H. (1986). Solving sparse linear equations over finite fields. *Information Theory, IEEE Transactions on*, **32**(1), 54–62. URL http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.128.3466&rep=rep1&type=pdf.

[27] Yang, M., Meng, Q., Wang, Z., Wang, L. a Zhang, H. (2013). Polynomial Selection for the Number Field Sieve in Geometric View. Cryptology ePrint Archive, Report 2013/583. URL https://eprint.iacr.org/2013/583.pdf.

Seznam algoritmů

3.1 Base-(m, p) rozklad čísla N 24
3.2 Kleinjungův algoritmus 31
5.1 Kořenové síto 59
5.2 Optimalizované kořenové síto 61
5.3 Průchod p^2-árním stromem s rekurzí 65
5.4 Kořenové síto na mřížce 67

Seznam obrázků

4.1 Příklad nezkoseného a zkoseného polynomu pro N se 124 decimálními ciframi. 36
4.2 Vliv koeficientů na graf funkce polynomu. 39
4.3 Odhad funkce $\pi(B)$ nerovností (4.7). 42
4.4 Průběh funkce (4.18) pro N pevné a d rostoucí 46
4.5 Průběh derivace funkce (4.18) podle d pro N pevné a d rostoucí . 47
4.6 Znázornění přínosového schématu 50
4.7 Znázornění přínosového schématu s konečnými podmnožinami . . 53

6.1 Výsledky řazení různými kritérii hodnocení velikosti polynomů. . . 74

Seznam tabulek

4.1 Optimální volba stupně d pro různé velikosti N 47
4.2 Minimální velikost N pro různé stupně d 48

5.1 Procentuální vylepšení při použití kořenového síta oproti přímému výpočtu $\alpha(\mathrm{Im}(F_{c_1c_0}))$ a vylepšení při optimalizaci oproti klasickému kořenovému sítu. 63

6.1 Volba různých N pro test zjednodušení hodnocení vlastností polynomů. 73
6.2 Intervaly pro volbu parametrů translace a rotace pro test zjednodušení hodnocení vlastností polynomů. 73
6.3 Procentuální srovnání řazení různými kritérii hodnocení velikosti polynomu vzhledem k Dickmanově normě. 75