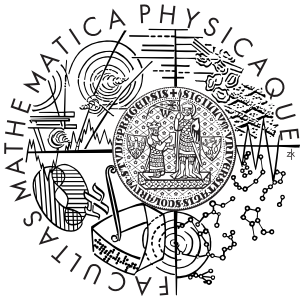
Univerzita Karlova v Praze
Matematicko-fyzikální fakulta

BAKALÁŘSKÁ PRÁCE



Miloslav Sobotka

Faktorizace polynomů a problém batohu

Katedra algebry

Vedoucí bakalářské práce: Mgr. Libor Barto, Ph.D.

Studijní program: Matematika, Matematické metody informační bezpečnosti

2010

Děkuji vedoucímu své práce Mgr. Liboru Bartovi, Ph.D. za odborné vedení a cenné připomínky k obsahu práce.

Dále bych rád poděkoval rodině za podporu a trpělivost.

Prohlašuji, že jsem svou bakalářskou práci napsal(a) samostatně a výhradně s použitím citovaných pramenů. Souhlasím se zapůjčováním práce a jejím zveřejňováním.

V Praze dne 6.8.2010 Miloslav Sobotka

Obsah

Název práce: Faktorizace polynomů a problém batohu
Autor: Miloslav Sobotka
Katedra (ústav): Katedra algebry
Vedoucí bakalářské práce: Mgr. Libor Barto, Ph.D.
e-mail vedoucího: Libor.Barto@mff.cuni.cz

Abstrakt: Cílem práce je popsat algoritmus na rozklad celočíselného polynomu na ireducibilní faktory, který ke hledání správné kombinace faktorů využívá mřížky a LLL algoritmu.

Hlavní část práce se zabývá algoritmem, pod kterým je podepsána čtveřice autorů Belabas, van Hoeij, Klüners a Steel. V práci vybudujeme potřebnou teorii pro dokázání korektnosti, konečnosti a časové složitosti algoritmu. Algoritmus pracuje v polynomiálním čase vzhledem ke stupni faktorizovaného polynomu.

Ke konci je také zmíněn původní van Hoeijův algoritmus založený na stejném principu.

V příloze jsou uvedeny ukázky běhu algoritmu na konkrétních příkladech a na přiloženém CD je implementace algoritmu BvKHS.

Klíčová slova: faktorizace, polynomy, algoritmus, mřížka, LLL algoritmus

Title: Factoring polynomials and the knapsack problem
Author: Miloslav Sobotka
Department: Department of Algebra
Supervisor: Mgr. Libor Barto, Ph.D.
Supervisor's e-mail address: Libor.Barto@mff.cuni.cz

Abstract: The aim of this thesis is to describe an algorithm for factoring a polynomial with integer coefficients into irreducible factors that uses lattices and LLL algorithm for the combination of the factors.

The main part of the work deals with the algorithm by four authors; Belabas, van Hoeij, Klüners and Steel. The theory needed for the proof of corectness, finiteness and time complexity of the algorithm will be presented. The algorithm works in polynomial time with respect to the degree of the polynomial.

The original van Hoeij's algorithm based on the same principle is mentioned in the end of the work.

In the appendix the algorithm is demonstrated on concrete examples, the implementation of the algorithm BvKHS can be found on the enclosed CD.

Keywords: factoring, polynomials, algorithm, lattice, LLL algorithm

Kapitola 1

Úvod

Tato práce se zabývá problémem rozkladu polynomů na nerozložitelné faktory. Po dobu čtyř desetiletí se v praxi užíval na faktorizaci Berlekamp-Zassenhausův algoritmus, který využíval pro získání rozkladu polynomu nad konečným tělesem $\mathbb{F}_p$ Berlekampův algoritmus, faktory posléze naliftoval pomocí algoritmu Henselovo liftování na rozklad, který je s faktorizovaným polynomem kongruentní modulo p^k, a tyto faktory pak Zassenhausovou metodou nakombinoval na správný rozklad na racionální faktory. V nejhorším případě však musel vyzkoušet všechny možnosti, což znamená, že pracoval v exponenciálním čase v závislosti na n, stupni rozkládaného polynomu.

Na začátku 80. let publikovala trojice Lenstra, Lenstra, Lovász článek [7], který byl z hlediska problému faktorizace průlomový, neboť jejich algoritmus měl dokazatelnou polynomiální složitost. V praxi se však neujal, protože byl pomalejší, než Zassenhausova metoda.

V roce 2002 představil Mark van Hoeij z Florida State University algoritmus (dále jen MvH algoritmus), který byl rychlejší než Zassenhausova metoda a dokázal rozložit i polynomy, na které tehdejší algoritmy nestačily. Jedinou vadou na kráse byl chybějící důkaz polynomiální složitosti. MvH algoritmus nejprve najde rozklad v tělese $\mathbb{F}_p$ pro malé prvočíslo p, který potom Henselovým liftováním navýší na rozklad modulo p^k. Narozdíl od Berlekamp-Zassenhausova algoritmu, který posléze hledá správnou kombinaci tak, že naliftované faktory zkouší roznásobovat a testuje, zda nedostane racionální faktory, MvH algoritmus najde správnou kombinaci přímo, převedením problému z multiplikativní do aditivní struktury, ve které potom hledá správnou kombinaci pomocí polynomiálního LLL algoritmu. K převodu využívá stopy polynomu, viz Kapitola 4. Tento algoritmus velmi záhy

začala implementovat většina výpočtových softwarů. Poměrně dlouhou dobu se však nedařilo dokázat jeho polynomialitu. Hlavním problémem bylo zjistit počet průchodů algoritmem. Tímto problémem a vylepšováním algoritmu se zabývá ve své disertační práci jeden z van Hoeijových studentů Andy Novocin.

V mezičase, kdy ještě nebyla dokázána polynomialita, se Mark van Hoeij pokusil společně s Karimem Belabasem, Jürgenem Klünersem a Allanem Steelem jít jinou cestou. Drželi se původní myšlenky, tedy převést problém do aditivní struktury a řešit ho pak rychle pomocí mřížek, k převodu však využili funkce nazývané logaritmická derivace. Podařila se jim najít podmínka na ukončení algoritmu a tím dokázat i jeho polynomialitu. Dále budeme tento algoritmus značit BvHKS.

Cílem této práce bylo popsat některý z algoritmů využívající myšlenky převedení problému kombinace faktorů do aditivní struktury, na kterou se posléze aplikuje LLL algoritmus na redukci mřížek. Podrobněji je zde uveden algoritmus BvHKS. V kapitole o MvH algoritmu se věnujeme krátce jeho představení a poukázání na odlišnosti mezi oběma algoritmy.

V úvodních kapitolách se věnuji poznatkům, které budeme dále užívat bez důkazu, ať už se týkají základů teorie resultantu, případně algoritmu LLL. Poměrně velký prostor jsem vyhradil seznámení se s p-adickými čísly, resp. p-adickými celými čísly. Ta mají obecně nekonečný rozvoj. Setkáváme se s nimi při Henselově liftování, které nám při navýšení rozkladu na modulo p^k odkrývá část rozvoje koeficientů p-adických faktorů, přesněji řečeno jeho prvních k cifer. Pokud bychom liftovali donekonečna, dostali bychom p-adické faktory. Pro pochopení algoritmu není znalost p-adických čísel nezbytně nutná, protože algoritmus pracuje pouze s jejich konečným rozvojem. Přesto jsem tuto kapitolu uvedl, neboť jsem při studiu nacházel spoustu materiálů týkajících se problému, kde se předpokládala základní znalost $p-$adických čísel a jejich vlastností.

Podněty k přístupu k zavedení p-adických čísel, jsem našel v [6], některé příklady jsou převzaté. Korektní důkaz, že p-adická čísla tvoří těleso je možné nalézt v [4], popřípadě [1]. Teorie resultantu je převzatá z [2]. Při zavádění mřížky a popisu algoritmu LLL jsem vycházel převážně z [7] a také z [2]. Důkaz Lemmatu 2.3.1 je pak inspirován [3]. Sekce o problému batohu vychází z popisu algoritmů MvH a BvHKS, jak v [3], tak v [5].

Návody k důkazům ze sekce 3.2 jsem získal v [3]. V důkaze Lemmatu 3.2.2 používám nerovnost dokázanou v [8]. K důkazu rovnosti 3.1 jsem využil textu [9]. Lemma 3.2.4 je převzaté z [3]. Důkazy v sekcích 3.3 a 3.4 jsou převzaty

z [3], Lemma 3.4.1 je pak z [5]. Důkaz složitosti je podpořen odhadem z [10].

Kapitola 4 je opět ovlivněna texty [5] a [3].

V textu budeme předpokládat znalost kapitoly týkající se faktorizace ze skript [2].

Kapitola 2

Příprava

2.1 Stručný úvod do p-adických čísel

V celé této sekci bude p značit prvočíslo. Hlavním úkolem této sekce je vytvořit těleso p-adických racionálních čísel $\mathbb{Q}_p$, které je zúplněním prostoru $\mathbb{Q}$ vzhledem k metrice $|\cdot|_p$. To znamená, že v prostoru $\mathbb{Q}_p$ je každá Cauchyovská posloupnost konvergentní v metrice $|\cdot|_p$. Pro konkrétnější představu bude teorie často prokládána příklady.

Definice (valuace)**.** Nechť $z \in \mathbb{Z}$ nenulové, p prvočíslo, pak p-valuací čísla z rozumíme

$$\mathrm{ord}_p z = \max\{k : p^k | z\}.$$

Nechť $r/s \in \mathbb{Q}$, pak p-valuací čísla $q = r/s$ rozumíme

$$\mathrm{ord}_p q = \mathrm{ord}_p r - \mathrm{ord}_p s.$$

Dále definujme $\mathrm{ord}_p 0 = \infty$.

Definice (p-adická norma)**.** Nechť $x \in \mathbb{Q}$. Pak definujeme normu $|\cdot|_p$

$$|x|_p = p^{-\mathrm{ord}_p x}.$$

Lemma 2.1.1. *Nechť $x, y \in \mathbb{Q}$. Norma $|x|_p : \mathbb{Q} \longrightarrow \mathbb{R}^+$ má následující vlastnosti:*

(1) $|x|_p = 0$*, právě když* $x = 0$

(2) $|xy|_p = |x|_p |y|_p$

(3) $|x+y|_p \leq \max\{|x|_p, |y|_p\}$

Definice (p-adická metrika)**.** Nechť $x, y \in \mathbb{Q}$. Pak definujeme metriku d

$$d : \mathbb{Q} \times \mathbb{Q} \to \mathbb{R}$$

$$d(x,y) = |x-y|_p \quad .$$

Příklad (p-adická norma)**.** Nechť $a = 2^3 \cdot 3 \cdot 7^2 \cdot 11 \cdot 13^{-1} \cdot 5^{-3}$.

$\lvert a\rvert_2 = \frac{1}{2^3} = 0.125$	$\lvert a\rvert_3 = \frac{1}{3} \doteq 0.333$	$\lvert a\rvert_5 = 5^3 = 125$	$\lvert a\rvert_7 = \frac{1}{7^2} \doteq 0.020$
$\lvert a\rvert_{11} = \frac{1}{11} \doteq 0.090$	$\lvert a\rvert_{13} = 13$	$\lvert a\rvert_{17} = 0$	$\lvert a\rvert_{19} = 0$

Definice. Definujme p-adické číslo a jako formální řadu

$$a = \sum_{i=n}^{\infty} a_i p^i,$$

kde $a_i \in \mathbb{F}_p = \mathbb{Z}/p\mathbb{Z}$, $n \in \mathbb{Z}$.

Tento jednoznačný zápis nazýváme kanonickou reprezentací čísla a. Často se používá zkrácený zápis p-adických čísel v následujícím tvaru.

$$\begin{array}{rl} a_n a_{n+1} \ldots a_{-2} a_{-1} . a_0 a_1 a_2 \ldots & \text{pro } n < 0 \\ . a_0 a_1 a_2 \ldots & \text{pro } n = 0 \\ . \underbrace{0 \ldots 0}_{n} a_0 a_1 a_2 \ldots & \text{pro } n > 0 \end{array}$$

Pokud od nějakého $j > n$ počínaje jsou $a_i = 0$, $i > j$, nulové koeficienty v zápise budeme vynechávat.

Příklad (Zápis p-adických čísel.)**.** Nechť $p = 5$.

$$\begin{array}{rcll} 13.41 & = & 1 \cdot 5^{-2} + 3 \cdot 5^{-1} + 4 \cdot 5^0 + 1 \cdot 5^1 & = \quad 241/25 \\ .1341 & = & 1 \cdot 5^0 + 3 \cdot 5^1 + 4 \cdot 5^2 + 1 \cdot 5^3 & = \quad 241 \\ .01341 & = & 0 \cdot 5^0 + 1 \cdot 5^1 + 3 \cdot 5^2 + 4 \cdot 5^3 + 1 \cdot 5^4 & = \quad 1205 \end{array}$$

Nyní postupně zavedeme na množině p-adických čísel sčítání, opačný prvek, násobení a inverzní prvek. V závěru sekce ukážeme, že množina p-adických čísel s takto nadefinovanými operacemi tvoří těleso, které budeme značit $\mathbb{Q}_p$. Sčítání a násobení v tělese $\mathbb{Q}_p$ probíha stejně jako násobení a sčítání v přirozených číslech jen s tím rozdílem, že místo dekadické báze máme p-bázi a čísla mohou mít nekonečně mnoho číslic.

Operace budeme pro větší přehlednost popisovat algoritmicky, často budeme používat nekonečných cyklů.

Ještě poznamenejme, že máme-li $a = \sum_{i=n}^{\infty} a_i x^i$, jsou $a_i = 0$ pro $i < n$.

Sčítání p-adických čísel

Algoritmus 1 (Sčítání)**.** Sčítání je podobné klasickému školskému algoritmu pro sčítání desetinných čísel pod sebou s tím rozdílem, že před zápisem spočtené cifry nemodulíme $\bmod 10$, ale $\bmod p$ a do vyššího řádu přičteme $\operatorname{div} p$.

VSTUP: $a = \sum_{i=n}^{\infty} a_i p^i$, $b = \sum_{i=m}^{\infty} b_i p^i$, $n < m$
VÝSTUP: $a + b$
1. $r := 0$, $b_i := 0$ pro $n \le i < m$
2. `for` $i = n$ `to` ∞ `do`
   $C_i := a_i + b_i + r$
   $c_i := C_i \bmod p$, $r := C_i \operatorname{div} p$
3. `return` $\sum_{i=n}^{\infty} c_i p^i$

Příklad. Nechť $p = 5$.

$$\begin{array}{rl} 2/3 = .413131313\ldots & .413131313\ldots \\ 5/6 = .014040404\ldots & +.014040404\ldots \\ & \overline{.422222222\ldots} \end{array}$$

Součet $.4\overline{13}$ a $.01\overline{40}$ je $.4\bar{2}$.

Příklad. Nechť $p = 5$.

$$\begin{array}{rl} 1/25 = 10. & 10. \\ 24/25 = 44. & 44. \\ & \overline{00.1} \end{array}$$

Součet 10. a 44. je .1.

Násobení p-adických čísel

Definice. Mějme p-adické číslo $a = \sum_{i=n}^{\infty} a_i p^i$. Řekneme, že a je základní, pokud $n = 0$ a $a_0 \neq 0$, nebo $a = .0$ (a je nulový prvek).

Každé p-adické číslo lze zapsat ve tvaru $a = Ap^n$, kde A je základní. Mějme p-adická čísla $a = Ap^n$, $b = Bp^m$, kde A, B jsou základní. Pak součin $ab = ABp^{m+n}$.

Stačí se tedy zaměřit na součin základních p-adických čísel, ostatní součiny se řeší přenásobením odpovídající mocninou p, což odpovídá posunu tečky.

Algoritmus 2 (Násobení)**.**

VSTUP: $a = \sum_{i=0}^{\infty} a_i p^i$, $b = \sum_{i=0}^{\infty} b_i p^i$, a, b jsou základní
VÝSTUP: $a \cdot b$
0. $r := 0$
1. for $i = 0$ to ∞ do
$C_i := r + \sum_{k=0}^{i} C(k)_{i-k}$, kde $C(k)_i = a_i b_k + (C(k)_{i-1} \text{ div } p)$
$c_i := C_i \bmod p$, $r := C_i \text{ div } p$
2. return $\sum_{i=0}^{\infty} c_i p^i$

Násobení je opět podobné školskému, jen nepočítáme nejdříve celé řádky, které nakonec sečteme, ale počítáme po sloupcích, tedy při každém průchodu cyklem spočteme jeden koeficient.

Příklad (Násobení základních p-adických čísel)**.** Uvažme 5-adická čísla
$\frac{1}{6} = .140\overline{40}\ldots$
$\frac{2}{3} = .413\overline{13}\ldots$

i	$C_i := r + \sum_{k=0}^{i} C(k)_{i-k}$	c_i	r
$-$			0
0	$r + C(0)_0 = 0 + 1 \cdot 4 = 4$	4	0
1	$r + C(0)_1 + C(1)_0 = 0 + 16 + 1 = 17$	2	3
2	$r + C(0)_2 + C(1)_1 + C(2)_0 = 3 + 0 + 4 + 3 = 10$	0	2
3	$r + C(0)_3 + C(1)_2 + C(2)_1 + C(3)_0 = 2 + 16 + 0 + 12 + 1 = 31$	1	6

Dostáváme výsledek $.4201\ldots$.

Pro zajímavost se můžeme podívat, jak by vypadalo násobení po řádcích:

$$\begin{array}{rr} & .14040\ldots \\ & .41313\ldots \\ \hline j=0 & 41313\ldots \\ j=1 & 1404\ldots \\ j=2 & 322\ldots \\ j=3 & 14\ldots \\ \vdots & 3\ldots \\ \hline & .42012\ldots \end{array}$$

Přirozená čísla v p-adickém zápisu

Každé přirozené číslo lze jednoznačně zapsat v p bázi. Tento zápis tvoří jeho p-adickou reprezentaci.

Příklad.

$$119 = 4 \cdot 5^0 + 3 \cdot 5^1 + 4 \cdot 5^2 = .434$$

Celá čísla v p-adickém zápisu

Přirozená čísla a $0 = .0$ převést na p-adická umíme, proto nám zbývá ukázat převod čísel záporných. To odpovídá hledání opačného čísla k přirozenému:

Algoritmus 3 (Opačný prvek)**.**
VSTUP: $a = \sum_{i=n}^{\infty} a_i p^i$
VÝSTUP: $-a$
1. $b_n := p - a_n$
2. `for` $i = n+1$ `to` ∞ `do`
   $b_i := p - a_i - 1$
3. `return` $\sum_{i=0}^{\infty} b_i p^i$

Příklad (Hledání opačného čísla)**.** Najděte číslo opačné k 1 v reprezentaci 5-adických čísel.
Snadno spočteme, že $-1 = .\overline{4}$. Obecně $-1 = .(p-1)(p-1)(p-1)\ldots = (p-1)+(p-1)\cdot p^1+(p-1)\cdot p^2+(p-1)\cdot p^3+\ldots$, po roznásobení dostaneme teleskopickou sumu, zbyde nám -1, ostatní členy se odečtou.

Racionální čísla v p-adickém zápisu

p-adický zápis racionálního čísla $\frac{a}{b}$, $\mathrm{NSD}(a,b) = 1$ získáme tak, že nalezneme inverz k b, ten pak přenásobíme číslem a.

Algoritmus 4 (Hledání inverzu)**.**
VSTUP: $a = \sum_{i=0}^{\infty} a_i p^i$, a je základní
VÝSTUP: a^{-1}
1. $c_0 := a_0^{-1} \bmod p$, $r = a_0 c_0 \operatorname{div} p$
2. `for` $i = 1$ `to` ∞ `do`
   $C_i := r + \sum_{j=1}^{i} C(j)_{i-j}$, kde $C(j)_i = a_j c_i + (C(j)_{i-1} \operatorname{div} p)$
   $c_i := -c_0(C_i) \bmod p$
   $r := r + a_0 c_i \operatorname{div} p$
3. `return` $a = \sum_{i=n}^{\infty} a_i p^i$

Příklad (Hledání inverzu základního p-adického čísla)**.** Uvažme 5-adické číslo $\frac{2}{3} = .413131313\ldots$

i	$C_i := r + \sum_{j=1}^{i} C(j)_{i-j}$	c_i	r
0		4	3
1	$r + C(1)_0 = 3 + 1 \cdot 4 = 7$	2	1
2	$r + C(1)_1 + C(2)_0 = 1 + 1 \cdot 2 + 1 + 12 = 16$	1	1
3	$r + C(1)_2 + C(2)_1 + C(3)_0 = 1 + 1 \cdot 1 + 3 \cdot 2 + 2 + 1 \cdot 4 = 14$	4	3

Pro kontrolu můžeme roznásobit:

a^{-1}	.4	2	1	4	...
a	.4	1	3	1	...
	1	1	1	2	...
		4	2	1	...
			2	3	...
	+			4	...
	.1				

Dělení základních p-adických čísel

Opět se omezíme na základní p-adická čísla a opět se necháme inspirovat školským algoritmem pro dělení. Chceme najít podíl

$$\frac{a_0 + a_1 p + a_2 p^2 + \dots}{b_0 + b_1 p + b_2 p^2 + \dots}.$$

Algoritmus 5 (Dělení)**.**
VSTUP: $a = \sum_{i=0}^{\infty} a_i p^i$, $b = \sum_{i=0}^{\infty} b_i p^i$
VÝSTUP: a/b
1. $c_0 := b_0^{-1} \cdot a_0$
2. `for` $i = 1$ `to` ∞ `do`
   $c_i := b_0^{-1} \cdot a_i \pmod{v}$
   $a := a - c_i \cdot b \cdot p^i$
3. `return` $\sum_{i=0}^{\infty} c_i p^i$

Příklad (Dělení základních p-adických čísel)**.** Uvažme 5-adická čísla
$\frac{2}{3} = .413131313\dots$
$\frac{1}{12} = .342424242\dots$

i	a	c_i	$-c_i \cdot b$	$a - c_i \cdot b \cdot p^i$
0	.41313131...	$2 \cdot 4 = 3 \pmod 5$	.11111111...	.34242424...
1	.34242424...	$2 \cdot 3 = 1 \pmod 5$	.20202020...	.0

V tuto chvíli můžeme výpočet ukončit, neboť $c_i = 0$, pro $i > 1$. Výsledek je $.31 = 3 + 1 \cdot 5^1 = 8$.

Nyní nahlédneme, že množina takto definovaných p-adických čísel společně s operacemi sčítání a násobení tvoří těleso.

Asociativita a komutativita u sčítání jsou zřejmé. Máme nulový prvek .0, k libovolnému prvku umíme najít prvek opačný.

Komutativita u násobení je vidět, asociativitu lze dokázat indukcí, existenci inverzního prvku jsme ukázali pro základní p-adická čísla, u ostatních je převedeme na hledání inverzu příslušného základního a provedeme korekci posunem tečky. Máme jednotkový prvek .1.

Distributivita je opět zřejmá, korektní důkaz lze provést indukcí.

Definice (p-adická celá čísla)**.** Definujme množinu p-adických celých čísel

$$\mathbb{Z}_p = \{a \in \mathbb{Q}_p : i < 0 \implies a_i = 0\}.$$

$\mathbb{Z}_p$ tvoří okruh v tělese $\mathbb{Q}_p$.

Stačí ověřit uzavřenost na sčítání a násobení z definice $\mathbb{Z}_p$.

Příklad. Nechť $a \in \mathbb{Z}$, pak i $a \in \mathbb{Z}_p$.

Nechť $a \in \mathbb{Q}$, $|x|_p < 1$, pak $a \in \mathbb{Z}_p$.

Příklad. Jak vypadá $\bmod p^k$ v okruhu p-adických celých čísel?
Mějme $\mathbb{Z}_5$. V tabulce budeme sledovat, jak se promítne $\bmod p^k$ v reprezentaci celých čísel

a $(\in \mathbb{Q})$	a $(\in \mathbb{Z}_p)$	$a_1 := a \bmod 5^1$ $(\in \mathbb{Z}_p)$	a_1 $(\in \mathbb{Q})$	$a_2 := \bmod 5^2$ $(\in \mathbb{Z}_p)$	a_2 $(\in \mathbb{Q})$
49	.441	.4	4	.44	24
13	.32	.3	3	.32	13
2/3	.413131...	.4	4	.41	9
3/4	.211111...	.2	2	.21	7
4	.4	.4	4	.4	4

Z definice $\mathbb{Q}_p$ je patrné, že modulo odpovídá "ořezávání" p-adického čísla. Povšimněme si ještě, že platí $ab \bmod p^k = (a \bmod p^k)(b \bmod p^k) \bmod p^k$. Např.: $3/4 \times 4$

$$\begin{array}{rl} & .21111\ldots \\ \times & .4 \\ \hline & .3 \end{array}$$

Po ořezání $\bmod 5^2$:

$$\begin{array}{rl} & .21 \\ \times & .4 \\ \hline & .301 \end{array}$$

Po oříznutí součinu $\bmod 5^2$ dostaneme opět výsledek .3.

2.2 Resultant

Definice (resultant)**.** Nechť $\mathbb{F}$ je těleso a $f = \sum_{i=0}^{n} a_i x^i$, $h = \sum_{i=0}^{m} b_i x^i$, jsou polynomy stupně n, resp. m nad tělesem $\mathbb{F}$. Nechť $\alpha_1, \ldots, \alpha_n$, resp. $\beta_1, \ldots, \beta_m$ jsou jejich kořeny v algebraickém uzávěru $\overline{\mathbb{F}}$. Pak definujeme resultant

$$\mathrm{Res}_{\mathbb{F}}(f, g) = a_n^m b_m^n \prod_{i=1}^{n} \prod_{j=1}^{m} (\alpha_i - \beta_j).$$

Často se resultant zavádí ekvivalentní definicí jako determinant čtvercové matice stupně $m + n$

$$\mathrm{Res}_{\mathbb{F}}(f, g) = det(A),$$

kde $k-$tý řádek matice A je tvaru:

$(\underbrace{0, \ldots, 0}_{k-1}, a_n, a_{n-1}, \ldots, a_0, \underbrace{0, \ldots, 0}_{m-k})$ pro $1 \leq k \leq m$

$(\underbrace{0, \ldots, 0}_{k-m-1}, b_m, b_{m-1}, \ldots, b_0, \underbrace{0, \ldots, 0}_{m+n-k})$ pro $m + 1 \leq k \leq m + n$.

Tvrzení 2.2.1 (Sylvestrovo kritérium)**.** *Nechť f, g jsou polynomy nad $\mathbb{Z}[x]$. Potom* $\mathrm{NSD}_{\mathbb{Q}[x]}(f, g) = 1$ *právě tehdy, když jejich resultant* $\mathrm{Res}_{\mathbb{Q}[x]}(f, g) \neq 0$. *Analogie tvrzení platí pro okruh $\mathbb{Z}_p[x]$ a těleso $\mathbb{Q}_p[x]$.*

Důkaz Sylvestrova kritéria je možné najít v [2].

Pro důkaz konečnosti algoritmu BvHKS se nám bude hodit horní odhad na resultant, využívající Hadamardovy nerovnosti.

Tvrzení 2.2.2 (Hadamardova nerovnost.)**.** *Nechť $A = (a_{ij})$ je matice $t \times t$ komplexních čísel. Pak platí*

$$|det(A)| \leq \prod_{i=1}^{t} \sqrt{\sum_{j=1}^{t} |a_{ij}|^2} \quad .$$

Důkaz je možné najít v [2].

Aplikací tvrzení na resultant snadno dostáváme $|\mathrm{Res}(f, g)| \leq \|f\|^m \|g\|^n$, kde $n = deg(f)$, $m = deg(g)$ a $\|\cdot\|$ je norma polynomu definovaná vztahem $\| \sum_{i=0}^{k} a_i x^i \| = \sqrt{\sum_{i=0}^{k} |a_i|^2}$

2.3 Mřížka, LLL algoritmus

V dalším textu zopakujeme základní definice týkající se mřížek a tvrzení, která budeme dále využívat.

Definice (mřížka)**.** Nechť $L \subseteq \mathbb{R}^n$. L nazveme mřížkou, pokud existuje báze $b_1, \ldots, b_n$ vektorového prostoru $\mathbb{R}^n$ taková, že

$$L = \sum_{i=1}^{n} \mathbb{Z} b_i = \left\{ \sum_{i=1}^{n} r_i b_i : r_i \in \mathbb{Z},\ 1 \leq i \leq n) \right\}.$$

Bázi $b_1, \ldots, b_n$ nazýváme bází mřížky L.

Připomeňme dále Gram-Schmidtův ortogonalizační proces známý z lineární algebry.
Nechť $b_1, \ldots, b_n$ jsou lineárně nezávislé vektory z $\mathbb{R}^n$. Gram-Schmidtův ortogonalizační proces nalezne vektory $b_1^*, \ldots, b_n^*$ splňující

(i) $\langle b_i^*, b_j^* \rangle = 0$, pro libovolné $1 \leq i < j \leq n$;

(ii) pro libovolné $1 \leq i \leq n$ je $b_i^* = b_i - x_i$, kde x_i je projekcí vektoru b_i na podprostor generovaný vektory $b_1, \ldots, b_{i-1}$, popř. $b_1^*, \ldots, b_{i-1}^*$.

Vektory x_i můžeme dopočítat induktivně

$$x_i = \sum_{j=1}^{i-1} \mu_{ij} b_j^*, \text{ kde } \mu_{ij} = \frac{\langle b_i, b_j^* \rangle}{\langle b_j^*, b_j^* \rangle}.$$

Definice. Bázi $b_1, \ldots, b_n$ mřížky L nazveme redukovanou, pokud splňuje následující podmínky:

(i) $|\mu_{ij}| \leq \frac{1}{2}$, pro $1 \leq j < i \leq n$;

(ii) $\|b_i^*\|^2 \geq (\gamma - \mu_{i\,i-1}^2)\|b_{i-1}^*\|^2$ pro $1 < i \leq n$, kde $\gamma \in (\frac{1}{4}, 1)$.

LLL algoritmus najde k zadané bázi mřížky redukovanou bázi a to v polynomiálním čase.

Lemma 2.3.1. *Nechť $K \in \mathbb{R}$ kladné. Nechť $L \subset \mathbb{R}^n$ je mřížka zadaná bází $b_1, b_2, \ldots, b_m$ a nechť $b_1^*, b_2^*, \ldots, b_m^*$ je odpovídající báze vzniklá Gram-Schmidtovou ortogonalizací . Vezměme $t = \min\{i : \|b_j^*\| > K\ \forall j, i < j \leq m\}$.*

Pak každý vektor $b \in L$, $\|b\| \leq K$ lze vyjádřit jako celočíselnou lineární kombinaci vektorů $b_1, b_2, \ldots, b_t$.

Důkaz. Nechť t jako v lemmatu. Vezměme libovolné b, $\|b\| \leq K$. Nechť $b = \sum_{i=1}^{m} r_i b_i = \sum_{i=1}^{m} r'_i b^*_i$, kde $r_i \in \mathbb{Z}$ a $r'_i \in \mathbb{R}$ pro $\forall i \in \{1, 2, \ldots, m\}$. Označme j nejmenší index splňující $r_i = 0$ pro $\forall i,\ j < i \leq m$.

Nejprve ukážeme, že $r_j = r'_j$. Vezměme bázový vektor b_j a podívejme se na jeho zápis v ohvězdičkované bázi. Označíme-li $b_j^{(i)}$ projekci b_j do směru b^*_i, odpovídající koeficient u b^*_i, $1 \leq i \leq j$, bude roven $\frac{\|b_j^{(i)}\|}{\|b^*_i\|}$. V případě $i = j$ pak $\frac{\|b_j^{(j)}\|}{\|b^*_j\|} = 1$, neboť vektor b^*_j vznikl jako projekce b_j do doplňku prostoru generovaného $b^*_1, \ldots, b^*_{j-1}$, tedy $b_j^{(j)} = b^*_j$. V zápisu b_j v ohvězdičkované bázi je tedy koeficient u b^*_j roven 1. Nyní si stačí rozmyslet, že rovnost $r_j = r'_j$ nám plyne z toho, že vektory b^*_i, $j < i \leq m$, které jediné mohly ovlivnit koeficient r'_j, se nepodílí na zápise b. Koeficient r'_j je tedy nenulové celé číslo, proto je splněna nerovnost

$$K \geq \|b\| \geq |r'_j| \cdot \|b^*_j\| \geq \|b^*_j\|.$$

Platí $j \leq t$, jinak $\|b^*_j\| \geq K$, spor s nerovností výše. Důkaz hotov. □

Tvrzení 2.3.2. *Nechť $b_1, \ldots, b_n$ je redukovaná báze mřížky L, $b^*_1, \ldots, b^*_n$ odpovídající báze vzniklá Gram-Schmidtovou ortogonalizací.*

Potom platí nerovnost

$$\|b_i\| \leq \left(\frac{4}{4\gamma - 1}\right)^{\frac{j-1}{2}} \|b^*_j\|, \text{ pro } 1 \leq i \leq j \leq m.$$

Důkaz tvrzení lze nalézt v [7], Proposition 1.6. Z tvrzení přímo plyne následující nerovnost

$$\|b_i\| \leq C^n \|b^*_j\|, \text{ pro } 1 \leq i \leq j \leq m, \tag{2.1}$$

kde C je vhodně zvolená konstanta. Volíme-li parametr γ dostatečně blízký 1, stačí použít $C > \sqrt{\frac{4}{3}}$. Tohoto poznatku využijeme v důkazu konečnosti BvKHS algoritmu.

2.4 Problém batohu

Problém batohu

Problém batohu je úloha, kdy máme z množiny předmětů různých hmotností vybrat předměty do batohu, abychom co nejlépe využili jeho nosnost, kterou ovšem nesmíme překročit.

Řešíme tedy úlohu

$$x_1 m_1 + x_2 m_2 + \ldots + x_r m_r \leq m,$$

kde m_i jsou hmotnosti předmětů, m je nosnost batohu a $x_i \in \{0, 1\}$. K řešení této úlohy nám může pomoci LLL algoritmus. Bázi mřížky budou tvořit sloupcové vektory matice

$$\begin{pmatrix} 1 & 0 & \ldots & 0 & 0 \\ 0 & 1 & \ldots & 0 & 0 \\ \vdots & & \ddots & \vdots & \vdots \\ 0 & 0 & \ldots & 1 & 0 \\ -m_1 & -m_2 & \ldots & -m_r & m \end{pmatrix}.$$

Řešením úlohy je (v případě rovnosti) nejkratší nenulový vektor mřížky s normou menší nebo rovnou $\sqrt{r}$. LLL algoritmus najde vektor, který bude v jistém smyslu dobrou aproximací tohoto řešení.

Jak problém batohu souvisí s problémem faktorizace?

Pokud chceme podobným způsobem řešit faktorizaci polynomů, je potřeba v první řadě najít způsob, jak převést úlohu z multiplikativní do aditivní struktury. K tomu využíváme v případě algoritmu BvHKS homomorfismu logaritmické derivace, MvH v původní realizaci tohoto nápadu používá stopu polynomu. Oba navíc zobrazují racionální faktory do celočíselných vektorů, p-adické faktory s koeficienty ze $\mathbb{Z}_p$ zobrazují do $\mathbb{Z}_p$.

Podívejme se, jak budou vypadat vektory mřížky odpovídající faktorům. Mějme r p-adických faktorů, že $f = lc(f) f_1 \cdots f_r$. Každý faktor bude reprezentován jedním bázovým vektorem. Máme-li i-tý faktor, bude mít jeho bázový vektor na prvních r souřadnicích 0 vyjma i-té, kde bude mít 1. Těchto r souřadnic nám u konkrétního vektoru mřížky kóduje, které p-adické faktory se podílejí na výstavbě polynomu příslušného vektoru. Jinými slovy, máme-li vektor mřížky b, jehož prvních r souřadnic nabývá hodnot $u_1, \ldots, u_r$, odpovídá vektor b polynomu $f_1^{u_1} \cdots f_r^{u_r}$.

Pro lepší představu malý příklad. Máme monický ireducibilní polynom $f \in \mathbb{Z}[x]$, v prvočíslo, v-adické faktory f_1, f_2, které nejsou ze $\mathbb{Z}[x]$. Chceme najít celočíselný rozklad, který v tomto případě bude tvořit samotný f. Mějme vhodný homomorfismus γ, tedy platí $\gamma(f_1 f_2) = \gamma(f_1) + \gamma(f_2)$. Nechť dále pro dostatečně velká l zobrazuje γ neracionální faktory h na velká v-adická čísla ve smyslu, že $|\gamma(h) \bmod v^l|$ je velké oproti $|\gamma(g) \bmod v^l|$, kde

g je racionální faktor. Tato vlastnost je klíčová pro to, aby dokázal LLL algoritmus oba typy faktorů od sebe rozlišit. Vektory mřížky budou tvaru $b_1 = (1, 0, w_1)$, $b_2 = (0, 1, w_2)$, kde $w_1 = \gamma(f_1) \bmod v^l$, $w_2 = \gamma(f_2) \bmod v^l$. Nyní w_1 i w_2 mají velkou normu, protože f_i jsou v-adické faktory, ale

$$b_1 + b_2 = (1, 1, w_1 + w_2) = (1, 1, \gamma(f_1 f_2) \bmod v^l) = (1, 1, \gamma(f) \bmod v^l)$$

má normu malou, protože $f \in \mathbb{Q}[x]$. Nyní musíme vyřešit otázku, jakým způsobem vnést do mřížky operaci $\bmod$. Protože chceme, aby se nám při modulení neměnilo prvních r souřadnic, stačí přidat do mřížky bázový vektor $b_3 = (0, 0, v^l)$. Jeho přičtením a odečtením se prvních r souřadnic nemění. Zadáme-li LLL algoritmu bázi b_1, b_2, b_3, vrátí nám $(1, 1, \gamma(f) \bmod v^l)$, odkud zrekonstruujeme faktor který bude součinem $f_1^1 f_2^1 = f$.

Protože algoritmy BvHKS i MvH používají obecně k reprezentaci faktorů k-rozměrné vektory, vypadá matice bázových vektorů mřížky následovně:

$$\begin{pmatrix}
1 & 0 & \dots & 0 & 0 & \dots & \dots & 0 \\
0 & \ddots & & \vdots & \vdots & & & \vdots \\
\vdots & & & 0 & \vdots & & & \vdots \\
0 & \dots & 0 & 1 & 0 & \dots & \dots & 0 \\
w_{1,1} & \dots & \dots & w_{1,r} & v^l & 0 & \dots & 0 \\
\vdots & & & \vdots & 0 & \ddots & & \vdots \\
\vdots & & & \vdots & \vdots & & & 0 \\
w_{k,1} & \dots & \dots & w_{k,r} & 0 & \dots & 0 & v^l
\end{pmatrix}.$$

Sloupce matice jsou bázovými vektory mřížky.
Prvky $w_{1,j}, \dots, w_{k,j}$ reprezentují j-tý v-adický faktor.
Posledních k bázových vektorů nám zprostředkuje operaci $\bmod v^l$.

Kapitola 3

Algoritmus BvHKS

3.1 Úmluva značení

Těleso v-adických čísel budeme značit stejně jako v předchozí kapitole $\mathbb{Q}_v$, okruh v-adických celých čísel $\mathbb{Z}_v$. Tělesem $\mathbb{F}_p$ budeme rozumět $\mathbb{Z}/p\mathbb{Z}$.

f budeme značit primitivní polynom ze $\mathbb{Z}[x]$ stupně $n > 1$, který je bezčtvercový nad $\mathbb{Q}$ a zároveň $\overline{f} = f \bmod v$ je bezčtvercový nad $\mathbb{F}_v$. Polynom g nad tělesem $\mathbb{F}$ nazýváme bezčtvercový nad $\mathbb{F}$, pokud neexistuje polynom $h \in \mathbb{F}[x]$, $deg(h) > 0$, že $h^2 \mid g$ v $\mathbb{F}[x]$. Polynom $f = \sum_{i=0}^{n} a_i x^i$ nad $\mathbb{Z}$ nazýváme primitivní, pokud $\mathrm{NSD}(a_0, a_1, \ldots, a_n) = 1$. Primitivní částí polynomu $f = \sum_{i=0}^{n} a_i x^i$ budeme rozumět polynom $pp(f) = \frac{1}{\mathrm{NSD}(a_0,a_1,\ldots,a_n)} \cdot f$. Dále $lc(h)$ budeme značit koeficient vedoucího členu polynomu h. Nechť v je prvočíslo takové, že $v \nmid lc(f)$ a $f \bmod v$ je bezčtvercový v $\mathbb{F}_v$.

Nechť $\overline{f_1}, \ldots, \overline{f_r}$ jsou monické ireducibilní polynomy nad $\mathbb{F}_v$ splňující $\overline{f} \equiv lc(\overline{f})\overline{f_1} \cdots \overline{f_r} \bmod v$. Dále $f_1, \ldots, f_r$ monické ireducibilní polynomy nad $\mathbb{Z}_v$ splňující rovnost $f = lc(f) f_1 \cdots f_r$ budeme nazývat v-adické faktory polynomu f. Monické ireducibilní polynomy $g_1, \ldots, g_s$ nad $\mathbb{Q}$, splňující $f = lc(f) g_1 \cdots g_s$, budeme nazývat $\mathbb{Q}$-faktory polynomu f.

Multiplikativní podgrupu racionálních funkcí nad $\mathbb{Q}_v$ generovanou množinou v-adických faktorů polynomu f budeme značit G_v a multiplikativní podgrupu generovanou $\mathbb{Q}$-faktory polynomu f budeme značit G_0.

Vektor $u(g) = (u(g)_1, u(g)_2, \ldots, u(g)_r) \in \{0,1\}^r$ značí vektor příslušný vektoru $g \in G_v$ splňující $g = \prod f_i^{u(g)_i}$, speciálně $u(f_i) = e_i$.

Symetrickým zbytkem čísla $a \in \mathbb{Z}$ modulo $\bmod v^l$ budeme rozumět s splňující $-v^l/2 < s \leq v^l/2$ takové, že $a \equiv s \bmod v^l$. Značíme $a \equiv s \operatorname{smod} v^l$.

Dále budeme používat $\|\cdot\|$ pro označení eukleidovské normy na vek-

torovém prostoru nad komplexními čísly, tedy $a = (a_1, \ldots, a_k) \in \mathbb{C}^k$, pak $\|(a_1, \ldots, a_k)\| = \sqrt{\sum_{i=1}^{k} |a_i|^2}$. Stejně budeme značit i normu polynomu $h = \sum_{i=1}^{k} a_i x^{i-1}$ nad $\mathbb{C}$ definovanou vztahem $\|h\| = \|(a_1, \ldots, a_k)\| = \sqrt{\sum_{i=1}^{k} |a_i|^2}$.

Nechť t je funkce z $\mathbb{N}$ do $\mathbb{N}$ a $t(n)$ je počet elementárních operací na vstupu délky n. Elementární operace je operace spočtená v konstantním čase. Řekneme, že $t(n)$ je asymptoticky menší rovno $r(n)$, značíme $t(n) \in O(r(n))$, pokud existuje konstanta $k > 0$ a $n_0 \in \mathbb{N}$, že $\forall n \geq n_0$ platí $0 \leq t(n) \leq r(n)$.

3.2 Logaritmická derivace a Mahlerova míra

Zavedení logaritmické derivace a základní vlastnosti

Definice (logaritmická derivace)**.** Zobrazení z racionálních funkcí nad $\mathbb{Q}_v$ bez konstantní nulové funkce do racionálních funkcí nad $\mathbb{Q}_v$

$$\phi : g \to \frac{g'}{g},$$

nazýváme logaritmickou derivací polynomu g.

Poznámka. ϕ je homomorfismus. Nechť $g, h \in \mathbb{Q}_v[x]$, pak $\phi(gh) = (gh)'/(gh) = (g'h + gh')/(gh) = g'/g + h'/h = \phi(g) + \phi(h)$.

Definice (homomorfismus Φ)**.** Definujme homomorfismus Φ z racionálních funkcí nad $\mathbb{Q}_v$ bez konstantní nulové funkce do racionálních funkcí nad $\mathbb{Q}_v$ předpisem

$$\Phi : g \to f\frac{g'}{g},$$

Φ je zřejmě homomorfismus.

Následující lemma nás upozorní na důležitou vlastnost homomorfismu Φ a sice, že zobrazuje $\mathbb{Q}$-faktory do $\mathbb{Z}[x]$.

Lemma 3.2.1. *Je-li $q \in G_v$, pak $\Phi(q) \in \mathbb{Z}_v[x]$. Je-li $q \in G_0$, pak $\Phi(q) \in \mathbb{Z}[x]$.*

Důkaz. Snadno nahlédneme, že pro faktor $f_i \in G_v$ platí $\Phi(f_i) \in \mathbb{Z}_v[x]$, neboť jednotlivé v-adické faktory jsou prvky $\mathbb{Z}_v[x]$ a protože $\mathbb{Z}_v[x]$ je okruh, je i f_i' elementem $\mathbb{Z}_v[x]$. Protože Φ je homomorfismus pro libovolný $q \in G_v$:

$\Phi(q) \in \mathbb{Z}_v[x]$.
Pro $i = 1, \ldots, s$ zapišme

$$g_i = \sum_{j=0}^{deg(g_i)} \frac{a_{ij}}{b_{ij}} x^j,$$

kde a_{ij}, $b_{ij} \in \mathbb{Z}$ a $\mathrm{NSD}(a_{ij}, b_{ij}) = 1$. Označme b nejmenší společný násobek prvků sjednocení $\bigcup_{i\in\{1,\ldots,s\}}\{b_{ij}|0 \leq j \leq deg(g_i)\}$. Je zřejmé, že $b \mid lc(f)$, neboť $f \in \mathbb{Z}[x]$ a g_i jsou monické. Tedy i bg_i dělí f v $\mathbb{Z}[x]$. Podívejme se na

$$\Phi(bg_i) = \frac{f}{bg_i}(bg_i)'.$$

Již víme, že $f/bg_i \in \mathbb{Z}[x]$, zřejmě i $(bg_i)' \in \mathbb{Z}[x]$, tedy i $\Phi(bg_i) \in \mathbb{Z}[x]$.
Protože $\Phi(b) = 0$, platí rovnost $\Phi(g_i) = \Phi(b) + \Phi(g_i) = \Phi(bg_i) \in \mathbb{Z}[x]$. Tedy $\Phi(g_i) \in \mathbb{Z}[x]$. Protože Φ je homomorfismus, dostáváme podobně jako v první části lemmatu závěr $q \in G_0$ implikuje $\Phi(q) \in \mathbb{Z}[x]$.

□

Další kroky povedou k získání horního odhadu na $\|\Phi(g)\|$, kde g je $\mathbb{Q}$-faktor. Tento poznatek je stěžejní pro odlišení $\mathbb{Q}$-faktoru od p-adických v části algoritmu pracující s mřížkou.

Mahlerova míra a základní vlastnosti

Definice (Mahlerova míra)**.** Mahlerovou mírou rozumíme funkci M, která polynomu $f(x) = lc(f)(x - x_1)(x - x_2)\ldots(x - x_n) \in \mathbb{C}[x]$ přiřadí

$$M(f) = |lc(f)| \prod_{i=1}^{n} \max\{|x_i|, 1\}.$$

Snadno nahlédneme, že $M(h_1 \cdot h_2) = M(h_1) \cdot M(h_2)$, kde $h_1, h_2 \in \mathbb{C}[x]$.

Lemma 3.2.2 (Horní odhad koeficientů $\Phi(g)$ pomocí Mahlerovy míry)**.** *Nechť $f \in \mathbb{Z}[x]$, $g \in \mathbb{Q}[x]$ polynomy, $g \mid f$, $deg(f) = n$, $deg(g) > 0$. Pak $M(\Phi(g)) \leq nM(f)$.*

Označíme-li $B_i = \binom{n-1}{i} nM(f)$ pro $i \in \{0, \ldots, n-1\}$ a $\Phi(g) = \sum_{i=0}^{n-1} b_i x^i$, dostáváme navíc odhad na i-tý koeficient $|b_i| \leq B_i$.

Důkaz. Využijeme nerovnosti $M(h') \leq deg(h) \cdot M(h)$ pro $h \in \mathbb{C}[x]$, jejíž důkaz provádět nebudeme, je možné ho nalézt v [8].

Použijeme-li tuto nerovnost na g, dostaneme $M(\Phi(g)) = M(fg'/g) = M(g') \cdot M(f/g) \leq deg(g) \cdot M(g) \cdot M(f/g) = deg(g) \cdot M(f) \leq nM(f)$.

Nechť $x_1, \ldots, x_{n-1}$ jsou komplexní kořeny polynomu $\Phi(g)$. Potom $\Phi(g) = lc(\Phi(g))(x-x_1)\cdots(x-x_{n-1})$. Označíme-li $\Phi(g) = lc(\Phi(g))(x^{n-1}+a_{n-2}x^{n-2}+\ldots+a_1x+a_0)$, pak po roznásobení vyjádření polynomu pomocí kořenů dostaneme, že koeficient u $(n-i)$-tého členu je roven

$$a_{n-i} = (-1)^{i+1} \sum_{1\leq j_1<j_2<\ldots<j_{i-1}\leq n-1} x_{j_1}x_{j_2}\cdots x_{j_{i-1}}, \quad i \in \{1,\ldots,n\}.$$

Odtud pak

$$|a_{n-i}| \leq \sum_{1\leq j_1<j_2<\ldots<j_{i-1}\leq n-1} |x_{j_1}x_{j_2}\cdots x_{j_{i-1}}|.$$

Počet možností, jak lze vybrat $(i-1)$-prvkovou podmnožinu kořenů z $n-1$, je $\binom{n-1}{i-1}$. Zbývá odhadnout shora součin libovolných $(i-1)$-kořenů. Bez újmy na obecnosti předpokládejme, že součin $|\prod_{j=1}^{i-1} x_j|$ je maximální, jinak kořeny přeznačíme. Odhadněme

$$\Big|\prod_{j=1}^{i-1} x_j\Big| \leq \prod_{j=1}^{i-1} |x_j| \leq \prod_{j=1}^{n-1} \max\{|x_j|, 1\}.$$

Dostáváme

$$|a_{n-i}| \leq \binom{n-1}{i-1} \prod_{j=1}^{n-1} \max\{|x_j|, 1\}.$$

Protože i se v odhadu vyskytuje pouze v kombinačním čísle, z vlastnosti kombinačních čísel $\binom{n}{k} = \binom{n}{n-k}$ dostáváme, že odhad na $(n-i)$-tý a $(i-1)$-ní koeficient je stejný. Za pomoci výsledku první části věty pak dostáváme

$$|b_{i-1}| = lc(\Phi(g))|a_{i-1}| \leq lc(\Phi(g))\binom{n-1}{i-1} \prod_{j=1}^{n-1} \max\{|x_j|, 1\}.$$

Ve výrazu vidíme rozepsanou $M(\Phi(g))$, dostaneme konečný odhad:

$$|b_{i-1}| \leq \binom{n-1}{i-1} M(\Phi(g)) \leq \binom{n-1}{i-1} nM(f), \quad i \in \{1,\ldots,n\}.$$

□

Lemma 3.2.3 (Odhad normy $\Phi(g)$)**.** *Nechť g je $\mathbb{Q}$-faktor polynomu $f \in \mathbb{Z}[x]$.*

Pak $\|\Phi(g)\| \leq B$, kde $B = 2^{n-1}n\|f\|$.

Důkaz. Použijeme stejné značení jako v předchozím důkazu. Tedy nechť $x_1, \ldots, x_{n-1}$ jsou komplexní kořeny polynomu $\Phi(g)$. Nechť $\Phi(g) = lc(\Phi(g))(x^{n-1} + a_{n-2}x^{n-2} + \ldots + a_1x + a_0)$. Již víme, že

$$|a_{n-i}| \leq \sum_{1 \leq j_1 < j_2 < \ldots < j_{i-1} \leq n-1} |x_{j_1}x_{j_2}\cdots x_{j_{i-1}}|.$$

Využijeme tohoto odhadu:

$$\|\Phi(g)\| = |lc(\Phi(g))|\sqrt{\sum_{i=0}^{n-2}|a_i|^2} \leq |lc(\Phi(g))|\sqrt{\sum_{i=1}^{n-1}\sum_{1 \leq j_1 < j_2 < \ldots < j_i \leq n-1}|x_{j_1}x_{j_2}\cdots x_{j_i}|}.$$

Vidíme, že oběma sumami sumujeme přes všechny podmnožiny množiny $I = \{1, \ldots, n-1\}$. Takových podmnožin je 2^{n-1}. Dostáváme odhad

$$\sum_{i=1}^{n-1}\sum_{1 \leq j_1 < j_2 < \ldots < j_i \leq n-1}|x_{j_1}x_{j_2}\cdots x_{j_i}| \leq 2^{n-1}\prod_{j \in M}|x_j|,$$

kde $M \subseteq I$ je maximální v tom smyslu, že

$$\prod_{j \in M}|x_j| = \max_{J \subseteq I}\Big\{\prod_{j \in J}|x_j|\Big\}.$$

Dále

$$\prod_{j \in M}|x_j| = \prod_{j \in M \wedge |x_j| < 1}|x_j| \cdot \prod_{j \in M \wedge |x_j| > 1}|x_j| \leq \prod_{j \in M \wedge |x_j| > 1}|x_j| = \prod_{i=1}^{n}\max\{|x_i|, 1\}.$$

Za použití Lemmatu 3.2.2 dostaneme:

$$\|\Phi(g)\| \leq |lc(\Phi(g))|\sqrt{2^{n-1}\prod_{i=1}^{n-1}\max\{|x_i|, 1\}} \leq$$

$$\leq |lc(\Phi(g))|2^{n-1}\prod_{i=1}^{n-1}\max\{|x_i|, 1\} \leq 2^{n-1}nM(f).$$

Zbývá dokázat nerovnost

$$M(f) \leq \|f\|. \tag{3.1}$$

Nejprve ukážeme, že pro $h \in \mathbb{C}[x]$, $z \in \mathbb{C}$ platí

$$\|(x-z)h(x)\| = \|(\bar{z}x-1)h(x)\|, \tag{3.2}$$

kde $\bar{z}$ značí číslo komplexně sdružené k z. Nechť $h = \sum_{i=0}^{m} h_i x^i$. Pak platí

$$\|(x-z)h(x)\|^2 = \left\| \sum_{i=0}^{m} h_i x^{i+1} - \sum_{i=0}^{m} h_i z x^i \right\|^2 = \left\| \sum_{i=0}^{m+1} (h_{i-1} - h_i z) x^i \right\|^2 = \sum_{i=0}^{m+1} |h_{i-1} - h_i z|^2$$

Protože $c \cdot \bar{c} = |c|^2$ pro $c \in \mathbb{C}$, můžeme pokračovat:

$$\sum_{i=0}^{m+1} |h_{i-1} - h_i z|^2 = \sum_{i=0}^{m+1} (h_{i-1} - h_i z)(\bar{h}_{i-1} - \bar{h}_i \bar{z}) =$$

$$= \sum_{i=0}^{m+1} (h_{i-1}\bar{h}_{i-1} + h_i \bar{h}_i z\bar{z}) - \sum_{i=0}^{m+1} (h_{i-1}\bar{h}_i \bar{z} + \bar{h}_{i-1} h_i z) =$$

$$= \sum_{i=0}^{m+1} (|h_{i-1}|^2 + |h_i|^2|z|^2) - \sum_{i=0}^{m+1} (h_{i-1}\bar{h}_i \bar{z} + \bar{h}_{i-1} h_i z) = \sum_{i=0}^{m} |h_i|^2 + \sum_{i=0}^{m} |h_i|^2 |z|^2 =$$

$$= (1 + |z|^2)\|h\|^2 - \sum_{i=0}^{m+1} (h_{i-1}\bar{h}_i \bar{z} + \bar{h_{i-1}} h_i z)$$

Z druhé strany

$$\|(\bar{z}x-1)h(x)\|^2 = \left\| \sum_{i=0}^{m} h_i \bar{z} x^{i+1} - \sum_{i=0}^{m} h_i x^i \right\|^2 = \left\| \sum_{i=0}^{m+1} (\bar{z}h_{i-1} - h_i) x^i \right\|^2 = \sum_{i=0}^{m+1} |\bar{z}h_{i-1} - h_i|^2.$$

Dále

$$\sum_{i=0}^{m+1} |\bar{z}h_{i-1} - h_i|^2 = \sum_{i=0}^{m+1} (\bar{z}h_{i-1} - h_i)(z\bar{h}_{i-1} - \bar{h}_i) =$$

$$= \sum_{i=0}^{m+1} (h_i \bar{h}_i + h_{i-1}\bar{h}_{i-1} z\bar{z}) - \sum_{i=0}^{m+1} (h_{i-1}\bar{h}_i \bar{z} + \bar{h}_{i-1} h_i z) =$$

$$= \sum_{i=0}^{m+1}(|h_i|^2+|h_{i-1}|^2|z|^2)-\sum_{i=0}^{m+1}(h_{i-1}\bar{h_i}\bar{z}+\bar{h}_{i-1}h_i z) = \sum_{i=0}^{m}|h_i|^2+\sum_{i=0}^{m}|h_i|^2|z|^2 =$$

$$= (1+|z|^2)\|h\|^2 - \sum_{i=0}^{m+1}(h_{i-1}\bar{h_i}\bar{z} + \bar{h_{i-1}}h_i z).$$

Vidíme, že strany se rovnají. Nyní přistoupíme k samotnému důkazu

$$M(f) \leq \|f\|.$$

Bez újmy na obecnosti, nechť $x_1, \ldots, x_k \in \mathbb{C}$ jsou právě kořeny f splňující $|x_i| > 1$, jinak je přeznačíme. Uvažme polynom

$$h = lc(f)\prod_{i=1}^{k}(\bar{x_i}x - 1)\prod_{i=k+1}^{n}(x - x_i) = \sum_{i=0}^{n}h_i x^i.$$

Na $\|h\|$ můžeme k-krát aplikovat rovnost 3.2 a dostaneme rovnost $\|h\| = \|f\|$. Dozajista platí

$$\sqrt{\sum_{i=0}^{n}|h_i|^2} = \|h\| \geq |h_n|.$$

Dále z definice h je $h_n = lc(f)\prod_{i=1}^{k}\bar{x_i}$. Potom

$$|h_n| = |lc(f)|\prod_{i=1}^{k}|\bar{x_i}| = |lc(f)|\prod_{i=1}^{k}|x_i| = M(f),$$

kde poslední rovnost plyne z výběru $x_1, \ldots, x_k$. Tedy

$$\|f\| = \|h\| \geq M(f).$$

Tím je důkaz lemmatu dokončen.

□

Na konci této kapitoly si ještě připravíme lemma, které využijeme v důkaze konečnosti algoritmu.

Nechť h je polynom nad $\mathbb{Z}_p$. Redukcí polynomu h budeme rozumět $\bar{h} = h \bmod v \in \mathbb{F}_v[x]$.

V následujícím lemmatu budeme redukovat polynom $g_j \in \mathbb{Q}[x]$. Jeho redukci můžeme provést, neboť je díky $v \nmid lc(f)$ zároveň v $\mathbb{Z}_p[x]$.

Lemma 3.2.4. *Nechť $G_0 \subsetneq G'(\subset G_v)$ a $(b_1, \ldots, b_h)$ je báze $\Phi(G')$ mod v^l. Pak existuje prvek $g \in G' \setminus G_0$ takový, že:*
(i) $f_i \mid \Phi(g)$ pro nějaké $i \in \{1, \ldots, r\}$.
(ii) $\overline{g_j} \nmid \overline{\Phi(g)} \forall 1 \leq j \leq s$.
(iii) $\Phi(g)$ mod v^l je tvaru $b_m + e \cdot \Phi(g_j) + \sum_{k \in K} \Phi(g_k)$ pro vhodné $m \in \{1, \ldots, h\}$, $j \in \{1, \ldots, s\}$, nějakou $K \subset \{1, \ldots, s\}$ a nějaké $e \in \mathbb{Z}$.

Důkaz. Nejprve najdeme ekvivalentní podmínku, která nám řekne, jak musí vypadat prvek $q \in G'$, aby splňoval $f_i \mid \Phi(q)$.

Libovolný prvek $q \in G'(\subseteq G_v)$ můžeme zapsat ve tvaru $q = f_1^{u_1} f_2^{u_2} \cdots f_r^{u_r}$, kde $u_i \in \mathbb{Z}, i \in \{1, \ldots, r\}$. Poznamenejme, že

$$\Phi(f_j) = f_1 f_2 \cdots f_{j-1} f_j' f_{j+1} \cdots f_r.$$

Zřejmě platí $f_i \mid \Phi(f_j) \Leftrightarrow i \neq j$, protože f_i jsou po dvou nesoudělné a ireducibilní v $\mathbb{Z}_v[x]$. Protože Φ je homomorfismus, platí $\Phi(f_j^{u_j}) = u_j \Phi(f_j)$. Ze stejného důvodu pro $q \in G_v$ dostáváme $\Phi(q) = \sum_{j=1}^{r} u_j \cdot \Phi(f_j)$. Díky bezčtvercovosti f platí následující ekvivalence: $f_i \mid \Phi(q) \Leftrightarrow f_i$ dělí každý sčítanec sumy $\sum_{j=1}^{r} u_j \cdot \Phi(f_j)$ a to platí, právě když $u_i = 0$. Tedy $f_i \mid \Phi(q) \Leftrightarrow u_i = 0$.

Definujme pro lib. prvek $q \in G_v$ jeho nosič jako $\mathrm{Supp}(q) = \{i | u_i \neq 0\}$. Tedy nosič polynomu q obsahuje indexy p-adických faktorů, které se podílejí na jeho zápise. Dále $g_j \mid \Phi(q)$, právě když žádný f_i, $i \in \mathrm{Supp}(g_j)$ nedělí q, to nastane, právě když $\mathrm{Supp}(q) \cap \mathrm{Supp}(g_j) = \varnothing$.

Nyní si vezmeme polynom spňující (i) a postupně zkonstruujeme prvek mající všechny tři vlastnosti (i),(ii),(iii). Zvolme pevné $q \in G' \setminus G_0$ splňující (i), jehož $\Phi(q)$ mod $v^l = b_m$ pro nějaké $m \in \{1, \ldots, h\}$. Mějme množinu $K \subseteq \{1, \ldots, s\}$ takovou, že $i \in K \Leftrightarrow g_i \mid \Phi(q)$, to je právě když $\mathrm{Supp}(q) \cap \mathrm{Supp}(g_j) = \varnothing$. Označme $q' = q \cdot \prod_{k \in K} g_k$. Nyní zřejmě platí $\mathrm{Supp}(q') \cap \mathrm{Supp}(g_j) \neq \varnothing$ pro $j \in \{1, \ldots, s\}$, tedy žádný faktor $g_j \nmid q'$, navíc q' stále splňuje (i). Můžeme $\Phi(q')$ zapsat jako $\Phi(q_0) + \sum_{k \in K} \Phi(q_k)$. Protože $\overline{f}$ je bezčtvercový, $\overline{g_i} \nmid \overline{\Phi(q')}$, protože $\overline{g_i}$ nedělí každý sčítanec sumy $\overline{\Phi(q_0)} + \sum_{k \in K} \overline{\Phi(q_k)}$. Nyní q' splňuje (i) i (ii).

(iii) Zapíšeme q' jako $q' = f_1^{v_1} f_2^{v_2} \cdots f_r^{v_r}$, $v_i \in \mathbb{Z}$. Protože $q' \notin G_0$, musí existovat $g_j = \prod_{i \in J} f_i$ takový, že $\{v_i | i \in J\}$ je víceprvková. Jinými slovy p-adické faktory f_i, $i \in J$ podílející se na zápise g_j, se nevyskytují v zápise q' ve stejné nenulové mocnině. Vezměme $e \in \{v_i | i \in J\}$ a označme $g = q'/g_j^e$. Nyní

$$\Phi\left(\frac{q'}{g_j^e}\right) = \Phi(q') - e\Phi(g_j) = b_m + \sum_{k \in K} \Phi(g_k) - e\Phi(g_j)$$

a g splňuje (i),(ii),(iii).

□

3.3 Algoritmus BvHKS neořezaná verze

Algoritmus 6 (Neořezaná verze BvHKS algoritmu)**.**
VSTUP: primitivní bezčtvercový polynom $f \in \mathbb{Z}[x]$
VÝSTUP: rozklad $f = lc(f)g_1g_2\cdots g_s$, kde $g_i \in \mathbb{Q}[x]$

0. najdi v, aby f mod v byl stále bezčtvercový, $v \nmid lc(f)$
1. najdi modulární faktory $f \equiv lc(f)\bar{f}_1\bar{f}_2\cdots\bar{f}_r \bmod v$
2. spočti $B_{max} = \max\{2B_i | 1 \leq i \leq n\}$,
   kde B_i je odhad daný Lemmatem 3.2.2
3. najdi nejmenší l splňující $v^{l-1} > B_{max}$, $l := 2l$,
   $B' := \sqrt{r + B^2}$, kde B z Lemmatu 3.2.3
4. naliftuj $f \equiv lc(f)f_1f_2\cdots f_r \bmod v^l$
5. spočti bázi mřížky L
6. najdi redukovanou bázi $b_1, \ldots, b_{r+n}$ mřížky L pomocí LLL algoritmu
7. označ $t = \min\{i : \|b_j^*\| > B' \; \forall j, i < j \leq m\}$,
   kde b_i^* jsou vektory vzniklé Gram-Schmidtovou ortogonalizací
8. označ L' mřížku vzniklou projekcí $\mathbb{Z}b_1 + \ldots + \mathbb{Z}b_t$ na prvních r souřadnic
9. zkus zrekonstruovat $\mathbb{Q}$-faktory z bázových vektorů $\in \{0,1\}^r$ mřížky L',
   pokud se to nepodaří, $l := 2l$ a jdi na krok 4

Vidíme, že v algoritmu nepoužíváme p-adické faktory (ve smyslu nekonečných posloupností), ale pouze jejich oříznutí mod v^l. Tím dostaneme konečný zápis jiných p-adických čísel, která odpovídají nějakým celým číslům. Díky tomu budeme moci některé další úvahy zúžit na polynomy ze $\mathbb{Z}[x]$.

V dalším okomentujeme jednotlivé kroky.

KROK 1 Pro faktorizaci nad konečným tělesem lze použít například polynomiální Berlekampův algoritmus pracující v čase $O(n^3v^2\log^3 v)$.

KROK 3 Tato volba l je heuristická. Nicméně $v^{l/2}$ je horní odhad na koeficienty u $\Phi(g)$, kde g je $\mathbb{Q}$-faktor. Tím, že liftujeme na dvojnásobnou délku koeficientů v p-adickém zápise, měli bychom být schopni LLL algoritmem od sebe odlišit $\mathbb{Q}$-faktory od p-adických. Pokud by bylo l malé a $\mathbb{Q}$-faktory se nepodařilo zrekonstruovat, l zdvojnásobíme a algoritmus projdeme znovu.

KROK 4 Pro nalezení rozkladu mod v^l použijeme polynomiální algoritmus Henselovo liftování pracující v čase $O(ln^3\log^2 v)$.

KROK 5 Označme $a_{i,j}$ $i-$tý koeficient polynomu $\Phi(f_j)$ mod v^l. Tedy

$$\Phi(f_j) \bmod v^l = \sum_{i=0}^{n-1} a_{i,j}x^i.$$

Nyní definujeme matici A mřížky L ve tvaru:

$$A = \begin{pmatrix} \mathrm{I}^{r\times r} & 0 \\ M & v^l \mathrm{I}^{n\times n} \end{pmatrix}, \text{ kde } M = \begin{pmatrix} a_{0,1} & \dots & a_{0,r} \\ \vdots & \ddots & \vdots \\ a_{n-1,1} & \dots & a_{n-1,r} \end{pmatrix} \text{ je matice koeficientů.}$$

Bázi mřížky tvoří sloupcové vektory matice.

KROK 6 Redukovanou bázi mřížky najdeme LLL algoritmem v čase $O((n+r)^6 \log^3 K)$, kde K je horní odhad normy bázových vektorů, jimiž je mřížka zadaná.

KROK 7, 8 Následující lemma nám osvětlí volbu B'.

Lemma 3.3.1. *Nechť g_j je $\mathbb{Q}$-faktor, $\Phi(g_j) \bmod v^l = \sum_{i=0}^{n-1} a_{i,j} x^i$. Označme $\tilde{g_j} \in L$ vektor odpovídající g_j takový, že*

$$\tilde{g_j} = \begin{pmatrix} u(g_j) \\ a_{0,j} \\ a_{1,j} \\ \vdots \\ a_{n-1,j} \end{pmatrix} \in L.$$

Pak

$$\|\tilde{g_j}\| \leq \sqrt{\|u(g_j)\|^2 + B^2} \leq \sqrt{r + B^2},$$

kde B je odhad daný Lemmatem 3.2.3. Označme $B' = \sqrt{r + B^2}$.

Důkaz. Zřejmý. □

Vektory mřížky odpovídající $\mathbb{Q}$-faktorům mají normu menší než B', budou proto ležet v mřížce generované vektory $b_1, \dots, b_t$, speciálně jejich projekce na prvních r souřadnic bude v mřížce L'. Pokud zvolíme l dostatečně velké, aby se nám do mřížky L' nedostali neracionální faktory, budou vektory $u(g_i)$, $i = 1, \dots, s$ generovat L'.

KROK 9 Pro získání $0-1$ báze, tedy bázových vektorů z $\{0,1\}^r$ použijeme Gauss-Jordanovu eliminaci na transponovanou matici bází mřížky L'. Ta nám najde jednoznačně určenou matici, která v případě, že l je dostatečně vysoké a mřížku tedy generují právě $\mathbb{Q}$-faktory, bude mít nutně na řádcích $u(g_i)$, $i = 1, \dots, s$.

Předpokládejme, že dostaneme bázový vektor $u \in \{0,1\}^r$ mřížky L'. Je-li f monický, získáme odpovídající faktor g snadno, neboť $g = \prod_{i=1}^{r} f_i^{u_i} \bmod v^l$. Pokud ovšem f není monický, vypočteme nejprve polynom $h \equiv lc(f) \prod_{i=1}^{r} f_i^{u_i} \bmod v^l$, odpovídající faktor g bude roven primitivní části polynomu h, tedy $g = pp(h)$. Nyní ověříme, zda $g \mid f$. Pakliže ne, není l dostatečně velké, opakujeme algoritmus s větším l.

3.4 Algoritmus BvHKS ořezaná verze

Jedním z vylepšení algoritmu je použití oříznuté mřížky, koeficienty $a_{i,j}$, které jsou z definice oříznuté shora ($\bmod v^l$), navíc ořízneme zdola.

Pro vhodná l_i, splňující $v^{l_i} > 2B_i$, kde B_i je odhad z Lemmatu 3.2.2, zavedeme ořezávací funkci Ψ. Nejprve označme $\overline{a_{i,j}} = a_{i,j} \bmod v^{l_i}$, tedy $\overline{a_{i,j}} \in (-\frac{v^{l_i}}{2}, \frac{v^{l_i}}{2}]$. Ořezávací funkci pak definujeme:

$$\Psi_{l_i}^{l}(a_{i,j}) = \frac{a_{i,j} - \overline{a_{i,j}}}{v^{l_i}}.$$

Mřížku $\tilde{L}$ pak definujeme maticí

$$\begin{pmatrix}
1 & 0 & \dots & 0 & 0 & \dots & \dots & 0 \\
0 & \ddots & & \vdots & \vdots & & & \vdots \\
\vdots & & & 0 & \vdots & & & \vdots \\
0 & \dots & 0 & 1 & 0 & \dots & \dots & 0 \\
\Psi_{l_0}^{l}(a_{0,1}) & \dots & \dots & \Psi_{l_0}^{l}(a_{0,r}) & v^{l_0} & 0 & \dots & 0 \\
\vdots & & & \vdots & 0 & \ddots & & \vdots \\
\vdots & & & \vdots & \vdots & & & 0 \\
\Psi_{l_{n-1}}^{l}(a_{n-1,1}) & \dots & \dots & \Psi_{l_{n-1}}^{l}(a_{n-1,r}) & 0 & \dots & 0 & v^{l_{n-1}}
\end{pmatrix},$$

jejíž sloupce tvoří bázi $\tilde{L}$.

Poznamenejme, že Ψ nám nezachovává homomorfismus, $\Psi(a+b) \neq \Psi(a) + \Psi(b)$, $a, b \in \mathbb{Z}$.

Lemma 3.4.1. *Nechť $a_1, \ldots, a_k \in \mathbb{Z}$, $a = \sum_{i=1}^{k} a_i$. Pak*

$$\Psi_m^l(a) = \varepsilon + \sum_{i=1}^{k} \Psi_m^l(a_i),$$

kde $\varepsilon \in \mathbb{Z}$, $|\varepsilon| \leq \frac{k}{2}$.

Důkaz. Platí

$$\frac{b+c}{v^m} = \frac{b}{v^m} + \frac{c}{v^m}, \text{ kde } b, c \in \mathbb{Z}$$

Nerovnost je tedy způsobena symetrickými zbytky vydělenými m-tou mocninou v. Rozdíl mezi b/v^m a $\Psi_m^l(b)$ je $\overline{b}/v^m$, což je v případě $v > 2$ v absolutní hodnotě méně než $1/2$, neboť $\overline{b} \in (-v^m/2, v^m/2]$ a v^m je liché. Dále

$$\Psi_m^l(a) \in \mathbb{Z}, \qquad \sum_{i=1}^{k} \Psi_m^l(a_i) \in \mathbb{Z},$$

tedy i $\varepsilon \in \mathbb{Z}$. Navíc $|\varepsilon| < \frac{k+1}{2}$, protože odhadujeme rozdíl pro k sčítanců a_i a jednou pro součet a. Protože ε je celé, platí $|\varepsilon| \leq \frac{k}{2}$.

Pro $v = 2$ je $|\overline{b}/v^m| \leq 1/2$. Tedy $|\varepsilon| \leq \frac{k+1}{2}$. Ukážeme, že ε splňující rovnost neexistuje. Pro spor nechť $\varepsilon = \frac{k+1}{2}$. Pak

$$\Psi_m^l(a) = \varepsilon + \sum_{i=1}^{k} \Psi_m^l(a_i).$$

Nutně všechny $\overline{a_i} = 1/2$, $\overline{a} = 1/2$.

$$\frac{a+1}{2^m} = -\frac{k+1}{2^m} + \sum_{i=1}^{k} \frac{a_i+1}{2^m}$$

$$a + 1 - \sum_{i=1}^{k} (a_i + 1) = -(k+1)$$

$$-k + 1 + a - \sum_{i=1}^{k} a_i = -(k+1)$$

$$1 = -1$$

Spor. Tedy máme ostrou nerovnost $|\varepsilon| < \frac{k+1}{2}$. Opět ε je celé, tedy $|\varepsilon| \leq \frac{k}{2}$. □

Nechť $l_i \in \mathbb{N}$ takové, že $v^{l_i} > 2B_i$ pro $i \in \{0, \ldots, n-1\}$. Je-li a_i i-tý koeficient polynomu $\Phi(g)$, kde g je $\mathbb{Q}$-faktor, platí po oříznutí $\Psi^l_{l_i}(a_i) = 0$ díky nerovnostem $|a_i| \leq B_i < v^{l_i}$. Máme-li $\mathbb{Q}$-faktor g a $\Phi(g) = \sum a_i x^i \bmod v^l$, pak mu odpovídá vektor mřížky

$$\tilde{g} = \begin{pmatrix} u(g) \\ \Psi^l_{l_0}(a_0) + \varepsilon_0 \\ \Psi^l_{l_1}(a_1) + \varepsilon_1 \\ \vdots \\ \Psi^l_{l_{n-1}}(a_{n-1}) + \varepsilon_{n-1} \end{pmatrix} = \begin{pmatrix} u(g) \\ \varepsilon \\ \varepsilon \\ \vdots \\ \varepsilon \end{pmatrix}.$$

Dostáváme mnohem lepší horní odhad $\|\tilde{g}\| \leq \sqrt{r + n(r/2)^2} =: B'$.

Algoritmus 7 (Ořezaný algoritmus BvHKS)**.**
VSTUP: primitivní bezčtvercový polynom $f \in \mathbb{Z}[x]$
VÝSTUP: rozklad $f = lc(f)g_1 g_2 \cdots g_s$, kde $g_i \in \mathbb{Q}[x]$

0. najdi v, aby $f \bmod v$ byl stále bezčtvercový, $v \nmid lc(f)$
1. najdi modulární faktory $\overline{f} \equiv lc(\overline{f})\overline{f_1 f_2} \cdots \overline{f_r} \bmod v$
2. spočti $l_0, \ldots, l_n$ splňující $v^{l_i} > 2B_i$,
   kde B_i je odhad daný Lemmatem 3.2.2
3. $B' := \sqrt{r + n(r/2)^2}$, spočti l splňující podmínku danou
   Lemmatem 3.4.2
4. naliftuj $f \equiv lc(f) f_1 f_2 \cdots f_r \bmod v^l$
5. spočti bázi mřížky $\tilde{L}$
6. najdi redukovanou bázi $b_1, \ldots, b_{r+n}$ mřížky $\tilde{L}$ pomocí LLL algoritmu
7. označ $t = \min\{i : \|b^*_j\| > B' \ \forall j, i < j \leq m\}$,
   kde b^*_i jsou vektory vzniklé Gram-Schmidtovou ortogonalizací
8. označ $\tilde{L}'$ mřížku vzniklou projekcí $\mathbb{Z}b_1 + \ldots + \mathbb{Z}b_t$
   na prvních r souřadnic
9. zrekonstruuj $\mathbb{Q}$-faktory z $0-1$ báze mřížky $\tilde{L}'$

Následující věta dokazuje konečnost ořezané verze algoritmu.

Věta 3.4.2 (Postačující podmínka na ukončení algoritmu BvHKS)**.** *Nechť $f \in \mathbb{Z}$, $deg(f) = n$, nechť v je prvočíslo takové, že $f \bmod v$ je bezčtvercový, v nedělí $lc(f)$ a $v = O(n \log n + n \log \|f\|)$. Pak ořezaná verze algoritmu BvHKS vrátí korektní výsledek a skončí, je-li splněna podmínka*

$$v^l > c^n(n) \cdot C^{n^2} \cdot \|f\|^{2n-1} \cdot (\log \|f\|)^n,$$

kde $c(n)$ lze dopočítat a C je konstanta z důsledku Lemmatu 2.3.2.

Důkaz. Důkaz věty provedeme v několika krocích. Nejprve ukážeme korektnost algoritmu, tedy že po proběhnutí algoritmu bude mřížka $\tilde{L}'$ obsahovat veškeré vektory odpovídající $\mathbb{Q}$-faktorům, tedy bude obsahovat celou mřížku L_0.
V dalších krocích pak bude naším cílem ukázat sporem, že splnění nerovnosti si vynutí ukončení algoritmu. Budeme předpokládat, že algoritmus neskončí. Najdeme vektor mřížky, který je obsažen v $\tilde{L}'$, nikoliv však v L_0, a ukážeme, že pak nutně nesplňuje nerovnost.

Mřížka $\tilde{L}'$ obsahuje L_0.

Z Lemmatu 2.3.1 víme, že všechny vektory mřížky $b \in L$, $\|b\| \leq B'$ jsou obsaženy v $\mathbb{Z}b_1 + \mathbb{Z}b_2 + \ldots + \mathbb{Z}b_t$, speciálně tedy i vektory odpovídající $\mathbb{Q}$-faktorům, neboť B' je horní odhad jejich normy. Jejich projekcí na prvních r souřadnic dostaneme mřížku $\tilde{L}'$ obsahující $u(g_j)$, $j \in \{1, \ldots, s\}$, které tvoří bázi mřížky L_0. Tedy $L_0 \subseteq \tilde{L}'$.

Výběr vektoru $\tilde{g}$ mřížky $\tilde{L}'$.

Je zřejmé, že multiplikativní grupa G_v je isomorfní aditivní grupě generované vektory $\{u(f_1), \ldots, u(f_n)\}$, podobně $\tilde{L}'$ je isomorfní multiplikativní grupě G' generované prvky množiny

$$\left\{ \prod_{i=1}^{r} f_i^{u_i} \mid u = (u_1, \ldots, u_r) \text{ je bázový vektor mřížky } \tilde{L}' \right\}.$$

Pro spor předpokládejme, že algoritmus neskončí, tedy $L_0 \subsetneq \tilde{L}' = \mathbb{Z}b_1 + \ldots + \mathbb{Z}b_t$. Nyní budeme chtít najít $g \in G' \setminus G_0$ splňující Lemma 3.2.4 a postupně zkonstruujeme jemu odpovídající vektor $\tilde{g}$ mřížky $\tilde{L}'$. Připomeňme, že $\Phi(g) = b_m + e \cdot \Phi(g_j) + \sum_{k \in K} \Phi(g_k)$ pro vhodné m, e, K, j.

Nejprve vezměme bázový vektor $\tilde{b_i}$ mřížky $\tilde{L}'$, jehož projekce na prvních r souřadnic neleží v L_0. Z Tvrzení 2.3.2 a rovnosti 2.1 víme, že $\|\tilde{b_i}\| \leq C^n B'$. Podobně, jako jsme v lemmatu přenásobovali faktory g_j, nyní přičteme vektory s indexy z množiny K. Označíme-li takto vzniklý vektor $g' = \tilde{b_i} + \sum_{k \in K} \tilde{b_k}$, dostáváme odhad

$$\|g'\| \leq C^n B' + sB' = (C^n + s)B',$$

neboť $\|\tilde{g_k}\| \leq B'$.

Nyní přičteme $e \in \mathbb{Z}$ násobek vektoru $\tilde{g_j}$. Odhadneme shora velikost e. Číslo e jsme vybírali z podmnožiny exponentů $v_1, \ldots, v_r$ takových, že $g' = f_1^{v_1} \cdots f_r^{v_r}$. Posledních n souřadnic vektoru $\tilde{g'}$ jsou oříznuté koeficienty

polynomu $\Phi(g')$. Protože $\Phi(g') = b_i + \sum_{i=1}^{r} v_i \Phi(f_i)$, můžeme e odhadnout $e \leq \|\tilde{g'}\|$. Dostaneme vektor $\tilde{g} = \tilde{g'} + e\tilde{g_j}$ splňující

$$\|\tilde{g}\| \leq \|\tilde{g'}\| + \|\tilde{g'}\|B' \leq \|\tilde{g'}\|(1 + B') \leq (C^n + s)B'(1 + B').$$

Dolní odhad resultantu. Označme $H = \Phi(g) \bmod v^l$. Na H se můžeme dívat jako na polynom nad celými čísly, neboť díky $\bmod v^l$ mají jeho koeficienty konečný rozvoj. Potom

$$\mathrm{Res}_{\mathbb{Q}}(f, H) \neq 0, \tag{3.3}$$

neboť kdyby $\mathrm{Res}_{\mathbb{Q}}(f, H) = 0$, pak nutně musí existovat společný faktor z $\mathbb{Q}[x]$, polynom f má však racionální faktory právě g_i, $1 \leq i \leq s$, tedy musí existovat $1 \leq j \leq s$, že $g_j \mid H$. Tím spíše $\overline{g_j} \mid \overline{H}$. To je ovšem ve sporu s výběrem g, který má vlastnost (ii).

Protože g splňuje podmínku (i), je $\mathrm{Res}_{\mathbb{Q}_p}(f, \Phi(g)) = 0$. Tím spíše platí $\mathrm{Res}_{\mathbb{Q}_p}(f, \Phi(g)) \equiv 0 \pmod{v^l}$. Z ekvivalentní definice resultantu pomocí determinantu je snadno vidět, že $0 \equiv Res_{\mathbb{Q}_p}(f, \Phi(g)) \equiv Res_{\mathbb{Q}_p}(f, \Phi(g) \bmod v^l)$. Protože $\Phi(g) \bmod v^l = H \in \mathbb{Z}[x]$, je $\mathrm{Res}_{\mathbb{Q}_p}(f, \Phi(g) \bmod v^l) = \mathrm{Res}_{\mathbb{Q}}(f, H)$. Protože $\mathrm{Res}_{\mathbb{Q}}(f, H) \equiv 0(\bmod v^l)$ a $\mathrm{Res}_{\mathbb{Q}}(f, H) \neq 0$ (3.3), nutně

$$v^l \mid \mathrm{Res}(f, H) \text{ a } |\mathrm{Res}(f, H)| \geq v^l. \tag{3.4}$$

Horní odhad resultantu.

Připomeňme, že $deg(H) = n-1$ a $deg(f) = n$. Použijeme Hadamardovy nerovnosti 2.2.2, dostáváme

$$|\mathrm{Res}(f, H)| \leq \|f\|^{n-1}\|H\|^n. \tag{3.5}$$

Horní odhad normy H.

Zbývá odhadnou normu $H = \Phi(g) \bmod v^l = \sum_{i=0}^{n-1} h_i x^i$. Vezměme $\tilde{l} = \max\{l_0, \ldots, l_{n-1}\}$. Budeme chtít dokázat, že $\tilde{l}$ splňuje

$$v^{\tilde{l}} \leq 2vB = 2v \cdot 2^{n-1} n \|f\|. \tag{3.6}$$

Nechť j je takové, že $\tilde{l} = l_j$. Protože l_j je nejmenší takové, že $2B_j < v^{l_j}$, platí $v^{l_j - 1} \leq 2B_j < v^{l_j}$. Po přenásobení nerovnosti prvočíslem v dostaneme $v^{l_j} \leq 2vB_j$. Dále

$$B_j = nM(f)\binom{n-1}{j} \leq nM(f)\sum_{i=0}^{n-1}\binom{n-1}{i} = nM(f)2^{n-1} \leq n2^{n-1}\|f\|,$$

kde poslední nerovnost jsme dostali z 3.1. Dostáváme tedy

$$v^{\tilde{l}} = v^{l_j} \leq 2v2^{n-1}n\|f\| = 2vB.$$

Z rovnosti

$$\Psi^l_{l_i}(h_i) = \frac{h_i - \bar{h_i}}{v^{l_i}}$$

máme $h_i = \Psi^l_{l_i}(h_i)v^{l_i} + \bar{h_i}$.
Potom $|h_i| \leq |\Psi^l_{l_i}(h_i)v^{l_i} + v^{l_i}\varepsilon_i| + v^{l_i}s/2$.

$$\|H\| = \sqrt{\sum_{i=0}^{n-1} |h_i|^2} \leq \sqrt{\sum_{i=0}^{n-1} \left|\Psi^l_{l_i}(h_i)v^{l_i} + v^{l_i}\varepsilon_i + \frac{s}{2}v^{l_i}\right|^2} \leq v^{\tilde{l}}\sqrt{\sum_{i=0}^{n-1} \left|\Psi^l_{l_i}(h_i) + \varepsilon_i + \frac{s}{2}\right|^2}.$$

Norma součtu je menší rovna součtu norem, tedy

$$v^{\tilde{l}}\sqrt{\sum_{i=0}^{n-1} \left|\Psi^l_{l_i}(h_i) + \varepsilon_i + \frac{s}{2}\right|^2} \leq v^{\tilde{l}}\left(\sqrt{\sum_{i=0}^{n-1} |\Psi^l_{l_i}(h_i) + \varepsilon_i|^2} + \frac{s}{2}\sqrt{n}\right).$$

Výraz s odmocninou je norma projekce vektoru mřížky, který odpovídá vektoru g, na posledních n souřadnic.

$$\tilde{g} = \begin{pmatrix} u(g) \\ \Psi^l_{l_0}(h_0) + \varepsilon_0 \\ \Psi^l_{l_1}(h_1) + \varepsilon_1 \\ \vdots \\ \Psi^l_{l_{n-1}}(h_{n-1}) + \varepsilon_{n-1} \end{pmatrix},$$

Dostáváme

$$\|H\| \leq v^{\tilde{l}}\big(\|\tilde{g}\| + \frac{s}{2}\sqrt{n}\big).$$

Tedy

$$\|H\| \leq v^{\tilde{l}}\left(\|\tilde{g}\| + \frac{ns}{2}\right) \leq v^{\tilde{l}}\left((C^n + s)B'(1 + B') + \frac{ns}{2}\right).$$

Dosadíme horní odhad $v^{\tilde{l}} \leq 2vB = 2v \cdot 2^{n-1}n\|f\|$ z rovnosti 3.6, dostaneme

$$\|H\| \leq 2v \cdot 2^{n-1}n\|f\|\left((C^n + s)B'(1 + B') + \frac{ns}{2}\right).$$

Z předpokladu $v = O(n \log n + n \log \|f\|)$, tedy asymptoticky $v \leq k(n \log n + n \log \|f\|)$ pro vhodné $k \in \mathbb{Z}$.

$$\|H\| \leq 2k(n \log n + n \log \|f\|)2^{n-1} n \|f\| \left((C^n + s)B'(1 + B') + \frac{ns}{2} \right).$$

Nyní provedeme sérii odhadů.

$$n \log n + n \log \|f\| \leq n2 \log(\max\{n, \|f\|\}) \leq 2n \log n \log \|f\|,$$

neboť $n > 1$, $\|f\| > 1$,

$$(C^n + s)B'(1 + B') + ns/2 \leq$$

$$\leq C^n((1 + s/C^n)\sqrt{r + (r/2)^2}(1 + \sqrt{r + (r/2)^2}) + ns/(2C^n)) \leq$$

$$\leq C^n((1 + n)\sqrt{n + (n/2)^2}(1 + \sqrt{n + (n/2)^2}) + n^2) \leq$$

$$\leq C^n(n^3(1/n + 1)\sqrt{1/n + 1/4}(1/n + \sqrt{1/n + 1/4}) + n^2) \leq$$

$$\leq 5 \cdot C^n n^3.$$

Máme

$$\|H\| \leq 2k(n \log n + n \log \|f\|)2^{n-1} n \|f\| \left((C^n + s)B'(1 + B') + \frac{ns}{2} \right) \leq$$

$$\leq 10 \cdot kn^5 \log n (2C)^n \|f\| \log \|f\|.$$

Označíme-li $c(n) = 10 \cdot kn^5 \log n$, máme

$$\|H\| \leq c(n)(2C)^n \|f\| \log \|f\|.$$

Závěr důkazu
Dostáváme

$$v^l \leq |\mathrm{Res}(f, H)| \leq \|f\|^{n-1} \|H\|^n \leq c^n(n) \cdot (2C)^{n^2} \cdot \|f\|^{2n-1} \cdot (\log \|f\|)^n.$$

Spor s předpokladem. □

Na závěr si ukážeme, že ořezaná verze algoritmu BvHKS má polynomiální asymptotickou složitost.

Lemma 3.4.3. *Označme* $h = \log \|f\|$. *Algoritmus má složitost* $O(n^{12} + n^9 h^3)$.

Důkaz. Podíváme se na složitost jednotlivých kroků. Z hlediska asymptotické složitosti je pro nás zajímavá jen část algoritmu pracující s LLL algoritmem. Předchozí kroky jsou asymptoticky polynomiální, viz [2] a jsou rychlejší než LLL algoritmus, jak uvidíme za chvíli.

Využijeme odhadu autorů Nguyen, Stehlé's uvedeného v [10] na složitost LLL algoritmu $O((d^5 m \log^3 K)$, kde d je dimenze mřížky, m je dimenze prostoru, ve kterém leží mřížka, K je horní odhad na normu vstupních bázových vektorů mřížky. V našem případě $d = m = n + r$, $\log K = \log \sqrt{r + nv^{2l}} = O(n^2 + nh)$. Dostáváme složitost $O(n^5 n(n^2 + nh)^3)) = O(n^{12} + n^9 h^3)$. □

Tento odhad na asymptotickou složitost není nikterak ohromující, stejnou má i původní LLL algoritmus na faktorizaci polynomů. Naše odhady však byly velmi velkorysé. Přesnější odhad je nabídnut v [3].

Kapitola 4

Algoritmus MvH

Jak již bylo řečeno, Mark van Hoeij využil jako první teorie mřížek a algoritmu LLL pro rychlou kombinaci faktorů. Stejně jako BvHKS pracuje s naliftovanými p-adickými faktory, které pak převádí do aditivní struktury pomocí homomorfismu. K tomu využívá stopy polynomu.

Definice. Nechť g je polynom. i-tou stopou polynomu g rozumíme

$$\mathrm{Tr}_i(g) = \sum_{\alpha | g(\alpha)=0} \alpha^i,$$

sumu přes všechny kořeny polynomu g i s násobností.

Veškerá potřebná teorie pro sestavení algoritmu a důkaz korektnosti je uvedena v práci [5].

Autoři algoritmu BvHKS ve své práci [3] tvrdí, že vztah

$$g'/g = \sum_{i=0}^{deg(g)} \mathrm{Tr}_i(g) x^{-(i+1)}$$

je klíčem k dokázání polynomiality algoritmu MvH. Lépe řečeno k dokázání polynomiality verze, kde p-adické faktory jsou reprezentovány přímo jejich stopami. Autoři však neuvádějí detaily a mě se je doplnit nepodařilo.

Literatura

[1] Baker A.: *An Introduction to p-adic Numbers and p-adic Analysis*, Glasgow, 2010.

[2] Barto L., Stanovský D.: *Počítačová algebra*, on-line verze, www.karlin.mff.cuni.cz/˜stanovsk/vyuka/skripta_palg.pdf, 2010.

[3] Belabas K., van Hoeij M., Klüners J., Steel A.: *Factoring Polynomials over Global Fields*, 2007.

[4] Gouvea F.Q.: *p-adic Numbers An Introduction*, Springer, 2000.

[5] van Hoeij M.: *Factoring Polynomials and the Knapsack Problem*, on-line verze, www.math.fsu.edu/˜hoeij/knapsack/paper/knapsack.ps.

[6] Koc C. K.: *A Tutorial on p-adic Arithmetic*, on-line verze, islab.oregonstate.edu/papers/r09padic.pdf, 2002.

[7] Lenstra A. K., Lenstra H. W., Lovász L.: *Factoring Polynomials with Rational Coefficients*, 2005.

[8] Mahler K.: *On the Zeros of the Derivative of Polynomial*, Proc. Roy. Soc. Ser. A, 1961.

[9] Mossinghoff M.J.: *Algorithms for the Determination of Polynomials with small Mahler Measure*, Austin, 1995.

[10] Nguyen P. Q., Stehlé D.: *Floating-Point LLL Revisited*, 2005.

Přílohy

Příklady na neořezanou verzi BvHKS algoritmu

Příklad. Najděte rozklad polynomu $f = 1 + 4x + 7x^2 + 6x^3 + x^4 + 4x^5 + 7x^6 + 6x^7$ v $\mathbb{Q}[x]$.

KROK 0
Zvolíme vhodné prvočíslo:

$v = 5$ je vyhovující, neboť $5 \nmid 6 = lc(f)$ a polynom $\bar{f} = (3 + x)(2 + x^2)(3 + x^2)(2 + 4x + x^2)$ je stále bezčtvercový.

KROK 1
$r = 4$
$\bar{f}_1 = 3 + x$
$\bar{f}_2 = 2 + x^2$
$\bar{f}_3 = 3 + x^2$
$\bar{f}_4 = 2 + 4x + x^2$

KROK 2
$M(f) = 6$
$B_{max} = 1680$

KROK 3
$l = 12$
$B' = 6399$

KROK 4
$f_1 = 3 + 6x$

$f_2 = 123327057 + x^2$
$f_3 = 120813568 + x^2$
$f_4 = 162760417 + 81380209x + x^2$

KROK 5

Báze mřížky:

1	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0	0	0	0
2	0	0	2	244140625	0	0	0	0	0	0
4	241627136	2513489	10	0	244140625	0	0	0	0	0
6	234086669	10053956	12	0	0	244140625	0	0	0	0
0	226546204	17594425	0	0	0	0	244140625	0	0	0
2	229059699	15080942	2	0	0	0	0	244140625	0	0
4	14	14	10	0	0	0	0	0	244140625	0
6	12	12	12	0	0	0	0	0	0	244140625

KROK 6

Redukovaná báze:

−2	1	3	4391	4536	16612547	9671645	13053441	10237531	12890501	−49359306
0	0	−1	−10936	30591	2309875	−2288928	17607657	−5605231	−43289243	2175999
0	0	−1	11113	−30408	2320685	−2291967	17625234	−5618676	−43273123	2184963
1	0	0	−1746	−1804	−12719314	−21345336	−3375873	−14030018	−3459631	−60184371
−2	2	6	5290	5464	7786466	−23347382	19355136	−7584974	18861740	25053271
2	4	12	−2810	−2907	10463763	−759851	8598062	42337214	7064624	3118837
0	6	18	−6262	−6476	−12270251	9632304	−1617718	−28467200	−3776846	−6034829
0	0	−4	−20044	−20711	19426875	−11816412	1466978	−7199424	1708403	10285946
−2	2	−10	−10778	−11138	−16210014	7484532	−2064193	20325515	−621733	−8472907
2	4	−16	2582	2666	4084888	5241315	23434258	−12344129	25795695	−5805591
0	6	−6	7518	7764	2610234	−8944277	−27676388	2518711	−26358458	10525756

KROK 7

Normy vektorů b_i^* :

4 10 29 30358 50833 37852670 38226997 47319943 56145555 73985782 81380208

Zvolíme $t = 3$.

KROK 8

Matice mřížky L' :

$$\begin{pmatrix} -2 & 1 & 3 \\ 0 & 0 & -1 \\ 0 & 0 & -1 \\ 1 & 0 & 0 \end{pmatrix}$$

KROK 9

Mřížka L', $\{0, 1\}$ bázové vektory:

$$\begin{matrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{matrix}$$

Zrekonstruujeme faktory nad $\mathbb{Z}$.
$u(g_1) = (1, 0, 0, 0)$, tedy $h_1 = 6 \cdot (3 + 6x) = 18 + 36x \text{ smod } 5^{12}$.
$g_1 = pp(h_1) = 1 + 2x$

$u(g_2) = (0, 1, 1, 0)$, tedy $h_2 = 6 \cdot (123327057 + x^2)(120813568 + x^2) = 6 + 6x^4 \text{ smod } 5^{12}$.
$g_2 = pp(h_2) = 1 + x^4$

$u(g_3) = (0, 0, 0, 1)$, tedy $h_3 = 6 \cdot (162760417 + 81380209x + x^2) = 2 + 4x + 6x^2 \text{ smod } 5^{12}$.
$g_3 = pp(h_1) = 1 + 2x + 3x^2$

RETURN
$(1 + 2x)(1 + x^4)(1 + 2x + 3x^2)$

Příklad. Najděte rozklad polynomu $f = -x - 2x^3 + 3x^4$ v $\mathbb{Q}[x]$.

KROK 0

Zvolíme vhodné prvočíslo:

$v = 2$ je vyhovující, neboť $2 \nmid 3 = lc(f)$ a polynom $\bar{f} = x(1+x)(1+x+x^2)$ je stále bezčtvercový.

KROK 1

$r = 3$

$\bar{f}_1 = x$

$\bar{f}_2 = 1 + x$

$\bar{f}_3 = 1 + x + x^2$

KROK 2

$M(f) = 3$

$B_{max} = 72$

KROK 3

$l = 16$

$B' = 119$

KROK 4

$f_1 = x$

$f_2 = 65535 + x$

$f_3 = 43691 + 43691x + x^2$

KROK 5

Báze mřížky:

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 21845 & 0 & 0 & 65536 & 0 & 0 & 0 \\ 0 & 43691 & 21845 & 0 & 65536 & 0 & 0 \\ 43690 & 43691 & 43689 & 0 & 0 & 65536 & 0 \\ 1 & 1 & 2 & 0 & 0 & 0 & 65536 \end{pmatrix}$$

KROK 6

Redukovaná báze:

0	3	3	1464	1389	−5617	−9387
3	0	0	−6914	4088	707	−7085
0	0	−3	5206	−2162	2096	−20945
0	−1	−1	−488	−463	−19973	3129
1	0	1	−4040	−19762	−463	4620
1	−2	3	9888	4040	487	−4886
3	3	−3	4962	1153	−718	7174

KROK 7

Normy vektorů b_i^* :

4	4	6	14697	19891	20873	24770

Zvolíme $t = 3$.

KROK 8

Matice mřížky L' :

$$\begin{matrix} 0 & 3 & 3 \\ 3 & 0 & 0 \\ 0 & 0 & -3 \end{matrix}$$

KROK 9

Mřížka L', $\{0, 1\}$ bázové vektory:

$$\begin{matrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{matrix}$$

Zrekonstruujeme faktory nad $\mathbb{Z}$.

$u(g_1) = (1, 0, 0)$, tedy $h_1 = 3 \cdot (x) = 3x \bmod 2^{16}$.
$g_1 = pp(h_1) = x$

$u(g_2) = (0, 1, 0)$, tedy $h_2 = 3 \cdot (65535 + x) = -3 + 3x \bmod 2^{16}$.
$g_2 = pp(h_2) = -1 + x$

$u(g_3) = (0, 0, 1)$, tedy $h_3 = 3 \cdot (43691 + 43691x + x^2) = 1 + x + 3x^2 \bmod 2^{16}$.
$g_1 = pp(h_1) = 1 + x + 3x^2$

RETURN

$(x)(-1 + x)(1 + x + 3x^2)$

Příklad. Najděte rozklad polynomu $f = -6-14x+4x^2+18x^3+4x^4$ v $\mathbb{Q}[x]$.

KROK 0

Zvolíme vhodné prvočíslo:

$v = 3$ je vyhovující, neboť $3 \nmid 4 = lc(f)$ a polynom $\bar{f} = x(2+x)(2+x+x^2)$ je stále bezčtvercový.

KROK 1

$r = 3$

$\bar{f}_1 = x$

$\bar{f}_2 = 2 + x$

$\bar{f}_3 = 2 + x + x^2$

KROK 2

$M(f) = 16$

$B_{max} = 384$

KROK 3

$l = 14$

$B' = 775$

KROK 4

$f_1 = 4676583 + x$

$f_2 = 2391485 + x$

$f_3 = 1901885 + 106390x + x^2$

KROK 5

Báze mřížky:

$$\begin{pmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 \\
3342427 & 4782966 & 3832026 & 4782969 & 0 & 0 & 0 \\
1955080 & 4782968 & 2827892 & 0 & 4782969 & 0 & 0 \\
2497875 & 4 & 4676588 & 0 & 0 & 4782969 & 0 \\
1 & 1 & 2 & 0 & 0 & 0 & 4782969
\end{pmatrix}$$

KROK 6

Redukovaná báze:

0	2	−4758	−7916	26898	42681	−669721
−1	−3	−877	13380	2140	−227419	−688348
0	2	−1881	−3510	−20668	79522	−1374454
3	8	267	−10196	−2964	−173706	3173
1	9	1618	20779	4869	93876	−6359
−4	7	3170	−5406	1535	−43532	3879
−1	3	−9397	−1556	−12298	−25694	675992

KROK 7

Normy vektorů b_i^* :

5 14 11313 28149 36478 318158 1807792

Zvolíme $t = 2$.

KROK 8

Matice mřížky L' :

$$\begin{matrix} 0 & 2 \\ -1 & -3 \\ 0 & 2 \end{matrix}$$

KROK 9

Mřížka L', $\{0, 1\}$ bázové vektory:

$$\begin{matrix} 1 & 0 \\ 0 & 1 \\ 1 & 0 \end{matrix}$$

Zrekonstruujeme faktory nad $\mathbb{Z}$.
$u(g_1) = (1, 0, 1)$, tedy $h_1 = 4 \cdot (4676583 + x)(1901885 + 106390x + x^2) = -12 - 4x + 16x^2 + 4x^3$ smod .
$g_1 = pp(h_1) = -3 - x + 4x^2 + 4x^3$

$u(g_2) = (0, 1, 0)$, tedy $h_2 = \cdot(2391485 + x) = 2 + 4x$ smod .
$g_2 = pp(h_2) = 1 + 2x$

RETURN

$(-3 - x + 4x^2 + 4x^3)(1 + 2x)$

Příklad. Najděte rozklad polynomu $f = 1 + 4x^5 + 2x^6 + x^7$ v $\mathbb{Q}[x]$.

KROK 0

Zvolíme vhodné prvočíslo:

$v = 2$ je vyhovující, neboť $2 \nmid 1 = lc(f)$ a polynom $\bar{f} = (1 + x)(1 + x + x^3)(1 + x^2 + x^3)$ je stále bezčtvercový.

KROK 1

$r = 3$
$\bar{f}_1 = 1 + x$
$\bar{f}_2 = 1 + x + x^3$
$\bar{f}_3 = 1 + x^2 + x^3$

KROK 2

$M(f) = 4$
$B_{max} = 1120$

KROK 3

$l = 24$
$B' = 2101$

KROK 4

$f_1 = 549367 + x$
$f_2 = 10711591 + 5608465x + 6240988x^2 + x^3$
$f_3 = 11958625 + 11002354x + 9986863x^2 + x^3$

KROK 5

Báze mřížky:

1	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0	0	0
10268615	10552071	12733746	16777216	0	0	0	0	0	0
12997455	8396119	12160858	0	16777216	0	0	0	0	0
12467351	5111816	15975265	0	0	16777216	0	0	0	0
11513247	5388299	16652886	0	0	0	16777216	0	0	0
14440551	6737730	12376171	0	0	0	0	16777216	0	0
16227851	10536234	6790359	0	0	0	0	0	16777216	0
1	3	3	0	0	0	0	0	0	16777216

KROK 6

Redukovaná báze:

1	−52456	−31758	40506	26311	14684	−100386	−37171	−105317	−776636
1	56148	63294	−49560	−114965	72126	67631	−33523	62086	−2570176
1	3827	−51105	24543	132606	−39628	−14345	69785	−68420	−2476491
0	138602	4446	89724	−1862	−12066	−61671	−58984	−84513	−66570
0	−92062	72742	−7004	80530	159038	−173999	126328	−2953	−77218
0	90331	−196947	69621	−1545	10216	−6399	74828	77529	44337
0	−35802	17122	180584	80902	−16698	89569	110408	−47569	−5430
20	−94751	56847	603	−77615	−55876	−23635	54472	−29315	−31229
12	82917	−95909	17879	79469	23756	8625	−132496	130697	35311
7	127469	4809	−34545	79234	112178	59472	71615	−124319	860579

KROK 7

Normy vektorů b_i^* :

24	274130	225137	215245	226367	184795	206795	200777	177056	3751499

Zvolíme $t = 1$.

KROK 8

Matice mřížky L' :

1
1
1

KROK 9

Mřížka L', $\{0, 1\}$ bázové vektory:

1
1
1

Zrekonstruujeme faktory nad $\mathbb{Z}$.

$u(g_1) = (1, 1, 1)$, tedy $h_1 = 1 \cdot (549367 + x)(10711591 + 5608465x + 6240988x^2 + x^3)(11958625 + 11002354x + 9986863x^2 + x^3) = 1 + 4x^5 + 2x^6 + x^7 \bmod 2^{24}$.

$g_1 = pp(h_1) = 1 + 4x^5 + 2x^6 + x^7$

RETURN

$(1 + 4x^5 + 2x^6 + x^7)$

Implementace BvHKS neořezaná verze

Na CD je přiložena implementace algoritmu BvHKS neořezané verze v programu Wolfram Mathematica v6.0. Implementace algoritmu kopíruje zde uvedenou verzi. Na začátku navíc testuje bezčtvercovost polynomu, není-li bezčtvercový, vrátí chybu. Spouští se voláním příkazu

```
Algoritmus[poly]
```

kde *poly* je polynom, který chceme rozložit. Pro snadnější testování a hledání vhodných příkladů je naimplementována funkce pro generování bezčtvercového polynomu.

```
Gener[fakt,koef,deg]
```

První parametr ovlivňuje počet faktorů, druhý velikost koeficientů, třetí jejich stupeň.

Na testovaných datech fungoval algoritmus správně, nicméně jsem neprováděl pečlivé ladění, ani optimalizaci. Algoritmus vznikl pouze za účelem konkrétnější představy o běhu algoritmu.