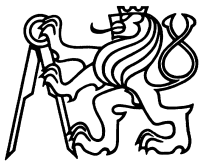
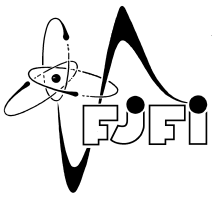


České vysoké učení technické v Praze
Fakulta jaderná a fyzikálně inženýrská



Kanonický rozklad tenzorů násobení polynomů a násobení matic na GF(2)

Canonical decomposition of tensors of polynomial multiplication and matrix multiplication in GF(2)

Bakalářská práce

Autor: **Vojtěch Obhlídal**

Vedoucí práce: **Ing. Petr Tichavský, DSc.**

Akademický rok: 2019/2020

České vysoké učení technické v Praze, Fakulta jaderná a fyzikálně inženýrská

Katedra: matematiky

Akademický rok: 2019/2020

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Student:	Vojtěch Obhlídal
Studijní program:	Aplikace přírodních věd
Obor:	Matematické inženýrství
Zaměření:	Aplikované matematicko-stochastické metody
Název práce (česky):	Kanonický rozklad tenzorů násobení polynomů a násobení matic na GF(2)
Název práce (anglicky):	Canonical decomposition of tensors of polynomial multiplication and matrix multiplication in GF(2)

Pokyny pro vypracování:

1. Na základě dostupné literatury sepište motivaci pro rozklad tenzorů na GF(2).
2. Popište souvislost mezi řešením soustavy polynomiálních rovnic na GF(2) a řešením úlohy splnitelnosti booleovských formulí (SAT problem).
3. Nalezněte rozklady tenzorů násobení polynomu (matic) do maximálního stupně, kam až to půjde, pomocí vybraných SAT solverů.
4. Nalezněte rozklady, které budou mít omezený počet jedniček ve faktorových maticích.

Doporučená literatura:

1. R. Zippel, Effective polynomial computations. Springer, 1993.
2. D. A. Bini, V. Pan, Polynomial and Matrix Computations. Birkhäuser Basel, 1994.
3. A. Biere, M. Heule, H. van Maaren, T. Walsh, Handook of satisfiability. IOS Press, 2009.

Jméno a pracoviště vedoucího bakalářské práce:

Ing. Petr Tichavský, DSc.
Ústav teorie informace a automatizace, AV ČR, v.v.i., Pod Vodárenskou věží 4, 182 00, Praha 8

Jméno a pracoviště konzultanta:

Datum zadání bakalářské práce: 31.10.2019

Datum odevzdání bakalářské práce: 7.7.2020

Doba platnosti zadání je dva roky od data zadání.

V Praze dne 23. října 2019

garant oboru

vedoucí katedry

ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE
Fakulta jaderná a fyzikálně inženýrská
-1-

děkan

Poděkování:
Chtěl bych zde poděkovat především svému školiteli Ing. Petru Tichavskému, DSc. za pečlivost, ochotu, vstřícnost a odborné i lidské zázemí při vedení mé bakalářské práce.

Tato práce vznikla s podporou Grantové agentury České republiky skrze projekt 17-00902S.

Čestné prohlášení:
Prohlašuji, že jsem tuto práci vypracoval samostatně a uvedl jsem všechnu použitou literaturu.

V Praze dne 19. července 2020 Vojtěch Obhlídal

Název práce:

Kanonický rozklad tenzorů násobení polynomů a násobení matic na GF(2)

Autor: Vojtěch Obhlídal

Obor: Matematické inženýrství

Zaměření: Aplikované matematicko-stochastické metody

Druh práce: Bakalářská práce

Vedoucí práce: Ing. Petr Tichavský, DSc., Ústav teorie informace a automatizace

Abstrakt: Cílem této práce je uvést čtenáře do problematiky hledání rozkladů tenzorů násobení polynomů a násobení matic. Také si klade za cíl popsat již existující algoritmy, diskutovat jejich složitost s důrazem na počet potřebných násobení a nalézt jim odpovídající rozklady tenzorů násobení. Dále se zaměřuje na seznámení se SAT solvery a uvádí metodu založenou na jejich použití. Ta převádí úlohu řešení soustavy algebraických rovnic na úlohu splnitelnosti soustavy booleovských formulí, s jejímž využitím lze hledat rozklady tenzorů. V tomto textu jsou dále prezentovány nové rozklady tenzorů, které byly získány výše zmíněnou metodou se zaměřením na jejich symetrie.

Klíčová slova: Karatsubův algoritmus, násobení matic, násobení polynomů, rozklad tenzoru, SAT solver, splnitelnost booleovských formulí

Title:

Canonical decomposition of tensors of polynomial multiplication and matrix multiplication in GF(2)

Author: Vojtěch Obhlídal

Abstract: The aim of this thesis is to introduce the reader into the problematics of finding the decompositions of tensors of polynomial multiplication and matrix multiplication. Another goal is to describe already existing algorithms, discuss their time complexity with the emphasis to quantity of multiplications needed and finding their corresponding tensor decompositions. Moreover, it focuses on acquainting the reader with SAT solvers and it presents a method based on their usage. The problem of solving the system of algebraic equations is transformed into the boolean satisfiability problem and it is possible to find decompositions of tensors with it. Furthermore, new decompositions of tensors obtained by method mentioned above are presented in this text with focus on their symmetries.

Key words: boolean satisfiability problem, decomposition of tensors, Karatsuba algorithm, matrix multiplication, polynomial multiplication, SAT solver

Obsah

Značení a zkratky **7**

Úvod **10**

1 Rozklad tenzorů **11**
1.1 Vlastnosti tenzorů . . . 11
1.2 Tenzor násobení matic . . . 12
1.2.1 Standardní algoritmus . . . 12
1.2.2 Strassenův algoritmus . . . 13
1.3 Tenzor násobení polynomů . . . 14
1.3.1 Standardní algoritmus . . . 15
1.3.2 Karatsubův algoritmus . . . 15
1.3.3 Srovnání algoritmů . . . 18
1.3.4 Toom-Cookův algoritmus . . . 20
1.4 Symetrie rozkladu tenzoru . . . 21
1.5 GF(2) . . . 22

2 Splnitelnost booleovských formulí **24**
2.1 Vlastnosti boolevských formulí . . . 24
2.2 SAT solvery . . . 26
2.3 Převod soustavy rovnic na SAT problém a jeho řešení . . . 26
2.4 Převod rozkladu z GF(2) na $\mathbb{Z}$ . . . 29

3 Získané rozklady **30**
3.1 Symetrie faktorových matic . . . 30
3.1.1 Předpoklad pevných sloupců a řádků . . . 31
3.2 Výsledky . . . 32
3.2.1 Násobení polynomů $n \in \{2, 3, 4\}$ . . . 33
3.2.2 Násobení polynomů $n = 5$ . . . 33
3.2.3 Násobení polynomů $n = 6$ . . . 35
3.2.4 Násobení polynomů $n = 7$ . . . 36

Závěr **39**

Značení a zkratky

Matice a vektory

Značení	**Význam**
$\mathbb{T}$	číselné těleso
$\mathbb{N}, \mathbb{Z}, \mathbb{R}$	množina přirozených, celých, reálných čísel
$\hat{n}$, kde $n \in \mathbb{N}$	množina $\{1, 2, \cdots, n\}$,

Matice a vektory

Značení	**Význam**
vektory	malá písmena a_1, a_2, b_1, b_2, x, y, atd.,
matice	velká písmena A, B, C, X, Y, atd.,
tenzory třetího řádu (dále jen tenzory)	velká písmena psaná kaligraficky $\mathcal{T}, \mathcal{T}_1, \mathcal{P}, \mathcal{P}_{2,2}$ atd.
$A \in \mathbb{T}^{n \times m}$	reálná matice s rozměry $n \times m$, s prvky $a_{ij} \in \mathbb{T}$,
$A_{i,:} \in \mathbb{T}^{n_2}$	i-tý řádek matice $A \in \mathbb{T}^{n_1 \times n_2}$,
$A_{:,j} \in \mathbb{T}^{n_1}$	j-tý sloupec matice $A \in \mathbb{T}^{n_1 \times n_2}$,
A^T	transpozice matice A,
A^{-1}	matice inverzní k A,
$\mathrm{vec}(A) \in \mathbb{T}^{nm}$	vektorizace matice $A \in \mathbb{T}^{n \times m}$,

Tenzory

Značení	Význam
$\mathcal{T}_{n_1,n_2,n_3} \in \mathbb{T}^{n_1 \times n_2 \times n_3}$	tenzor třetího řádu o rozměrech $n_1 \times n_2 \times n_3$,
$(\mathcal{T}_{n_1,n_2,n_3})_{i,j,k} \in \mathbb{T}$	i,j,k-tý prvek tenzoru $\mathcal{T}_{n_1 n_2 n_3} \in \mathbb{T}^{n_1 \times n_2 \times n_3}$,
$\mathcal{T}(:, i_2, i_3) \in \mathbb{T}^{n_1}$	vlákno tenzoru třetího řádu v módu 1,
$\mathcal{T}(:, :, i_3) \in \mathbb{T}^{n_1 \times n_2}$	řez tenzoru třetího řádu v módu (1,2),
$\mathrm{vec}(\mathcal{T}) \in \mathbb{T}^{n_1 n_2 n_3}$	vektorizace tenzoru $\mathcal{T} \in \mathbb{T}^{n_1 \times n_2 \times n_3}$,
$T^{(1)} \in \mathbb{T}^{n_1 \times n_2 n_3}$	$[T(:,:,1), T(:,:,2), \ldots, T(:,:,n_3)]$, matrizace tenzoru $\mathcal{T} \in \mathbb{T}^{n_1 \times n_2 \times n_3}$ podle třetí dimenze,
$T^{(2)} \in \mathbb{T}^{n_1 \times n_2 n_3}$	$[T(:,1,:), T(:,2,:), \ldots, T(:,n_2,:)]$, matrizace tenzoru $\mathcal{T} \in \mathbb{T}^{n_1 \times n_2 \times n_3}$ podle druhé dimenze,
$T^{(3)} \in \mathbb{T}^{n_2 \times n_1 n_3}$	$[T(1,:,:), T(2,:,:), \ldots, T(n_1,:,:)]$, matrizace tenzoru $\mathcal{T} \in \mathbb{T}^{n_1 \times n_2 \times n_3}$ podle první dimenze,

Operace

Vektorizace

Nechť $A \in \mathbb{T}^{n \times m}$, potom definujeme operaci $\mathrm{vec}(A)$ matice A vztahem

$$\mathrm{vec}(A) = \begin{pmatrix} A_{:,1} \\ A_{:,2} \\ \vdots \\ A_{:,m} \end{pmatrix} \in \mathbb{T}^{nm}.$$

Dále nechť $\mathcal{T} \in \mathbb{T}^{n_1 \times n_2 \times n_3}$, definujeme vektorizaci tenzoru $\mathcal{T}$ vztahem

$$\mathrm{vec}(\mathcal{T}) = \begin{pmatrix} \mathcal{T}(:,1,1) \\ \vdots \\ \mathcal{T}(:,n_1,1) \\ \mathcal{T}(:,1,2) \\ \vdots \\ \mathcal{T}(:,n_1,2) \\ \vdots \\ \mathcal{T}(:,n_1,n_2) \end{pmatrix} \in \mathbb{T}^{n_1 n_2 n_3}.$$

Násobení podle dimenze

Nechť $\mathcal{T} \in \mathbb{T}^{n_1 \times n_2 \times n_3}$ a $E \in \mathbb{T}^{n_1 \times m}$, potom produkt $\mathcal{T} \times_1 E = \mathcal{S} \in \mathbb{T}^{m \times n_2 \times n_3}$ nazveme násobení podle 1. dimenze, kde definiční vztah pro prvky $\mathcal{S}$ je dán vztahem

$$(\mathcal{S})_{i,j,k} = \sum_{l=1}^{n_1} \mathcal{T}_{l,j,k} E_{l,i}.$$

Analogicky lze provádět násobení podle 2. (resp. 3.) dimenze $\times_2$ (resp. $\times_3$).

Kroneckerův součin

Nechť $A \in \mathbb{T}^{n_1 \times n_2}$ je matice s prvky $a_{i,j}$, $B \in \mathbb{T}^{n_3 \times n_4}$. Potom Kroneckerův součin je definován vztahem

$$A \otimes B = \begin{pmatrix} a_{1,1}B & a_{1,2}B & \dots & a_{1,n_2}B \\ a_{2,1}B & a_{2,2}B & \dots & a_{2,n_2}B \\ \vdots & \vdots & \ddots & \vdots \\ a_{n_1,1}B & a_{n_1,2}B & \dots & a_{n_1,n_2}B \end{pmatrix} \in \mathbb{T}^{n_1 n_3 \times n_2 n_4}.$$

Khatri-Rao produkt

Nechť $A \in \mathbb{T}^{n_1 \times n_3}$ je matice s prvky $a_{i,j}$, $B \in \mathbb{T}^{n_2 \times n_3}$. Khatri-Rao produkt je definován tímto způsobem

$$A \odot B = \begin{pmatrix} a_{1,1}B_{:,1} & a_{1,2}B_{:,2} & \dots & a_{1,n_3}B_{:,n_3} \\ a_{2,1}B_{:,1} & a_{2,2}B_{:,2} & \dots & a_{2,n_3}B_{:,n_3} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n_1,1}B_{:,1} & a_{n_1,2}B_{:,2} & \dots & a_{n_1,n_3}B_{:,n_3} \end{pmatrix} \in \mathbb{T}^{n_1 n_2 \times n_3}.$$

Hadamardův produkt

Nechť $A, B \in \mathbb{T}^{n_1 \times n_2}$ jsou matice s prvky $a_{i,j}$, resp. $b_{i,j}$. Potom Hadamardův produkt má tvar

$$A * B = \begin{pmatrix} a_{1,1}b_{1,1} & a_{1,2}b_{1,2} & \dots & a_{1,n_2}b_{1,n_2} \\ a_{2,1}b_{2,1} & a_{2,2}b_{2,2} & \dots & a_{2,n_2}b_{2,n_2} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n_1,1}b_{n_1,1} & a_{n_1,2}b_{n_1,2} & \dots & a_{n_1,n_2}b_{n_1,n_2} \end{pmatrix} \in \mathbb{T}^{n_1 \times n_2}.$$

Úvod

Násobení matic a polynomů jsou ve výpočetní technice velmi běžné operace. Jejich časová složitost je polynomiální, například výpočet součinu čtvercových matic, popřípadě polynomů stejných stupňů pomocí standardních algoritmů má složitost $O(n^3)$, respektive $O(n^2)$.
Standardní algoritmy ale nejsou tím nejefektivnějším nástrojem pro tento výpočet. Byly navrženy algortimy, jejichž implementací se celková výpočetní doba snižuje, což je výhodné zejména při práci s maticemi velkých rozměrů či polynomy vyšších stupňů.

Jedním z prvních významnějších algoritmů pro efektivnější násobení čtvercových matic, publikovaný v roce 1969, je Strassenův algoritmus [6], který snižuje asymptotickou složitost z $O(n^3)$ na $O(n^{2.807})$. Dalších zlepšení pak bylo dosaženo v roce 1990 díky Coppersmithovu-Winogradovu algoritmu [7] se složitostí $O(n^{2.3755})$ a následně jeho optimalizací v [8],[9] a [10] na hodnotu $O(n^{2.3729})$.

Nový algoritmus pro násobení polynomů stejného stupně, prezentovaný v roce 1963, je Karatsubův-Ofmanův algoritmus (známý jako Karatsubův algoritmus) [5]. Ten snižuje složitost polynomiálního násobení z $O(n^2)$ na $O(n^{1.585})$. Násobení polynomů lze také provést pomocí rychlé Fourierovy transformace se složitostí $O(n \log n)$. Je ale třeba přejít na komplexní aritmetiku, což v některých aplikacích není žádané.

Již zmíněné násobení matic, resp. polynomů lze reprezentovat tenzorem násobení matic, resp. polynomů. Ten lze rozložit na tzv. faktorové matice, které odpovídají algoritmu použitému pro dané násobení. Tato práce se zabývá uvedením čtenáře do této problematiky, analýzou již známých publikovaných algoritmů na násobení matic a polynomů a nalezení rozkladů tenzorů jim odpovídajícím. Dále je cílem prezentovat a otestovat metodu hledání rozkladu tenzorů využívající software zvaný SAT solver. Princip této metody spočívá v převedení soustavy algebraických rovnic na úlohu splnitelnosti booleovských formulí a následné řešení pomocí již zmíněného SAT solveru.

Text je strukturován do tří kapitol. V kapitole 1 uvádíme základní definice nutné k porozumění dané problematice. Dále jsou zde rozebrány algoritmy na násobení matic a polynomů a jejich srovnání. Pro násobení matic se zaměřujeme na standardní a Strassenův algoritmus, k tématu násobení polynomů prezentujeme standardní, Karatsubův a Toom-Cookův algoritmus. V této kapitole také k těmto algoritmům uvádíme jim odpovídající rozklady tenzorů a algoritmus popisující tento převod pro tenzory násobení polynomů. První kapitola navíc diskutuje symetrie tenzorů a je zde definováno těleso GF(2) a je popsán způsob jeho využití při hledání rozkladů.
Ve druhé kapitole popisujeme vlastnosti boolevských fomulí a diskutujeme úlohu splnitelnosti boolevských formulí v souvislosti s jejím využitím při řešení soustavy rovnic. Tato kapitola navíc obsahuje rozsáhlý příklad, který ilustruje celý proces.
V kapitole 3 jsou prezentovány získané výsledky se zaměřením na symetrie a zkoumání různých konfigurací těchto symetrií. Dále jsou zde uvedeny konkrétní rozklady tenzorů násobení polynomů, které byly nalezeny s využitím metody SAT solverů.

Kapitola 1

Rozklad tenzorů

V tomto textu budeme tenzorem označovat tenzor 3. řádu, tj. $\mathcal{T} \in \mathbb{T}^{n_1 \times n_2 \times n_3}$.

1.1 Vlastnosti tenzorů

Definice 1.1. Nechť $a \in \mathbb{T}^{n_1}, b \in \mathbb{T}^{n_2}, c \in \mathbb{T}^{n_3}$ jsou nenulové sloupcové vektory. Potom každý tenzor s prvky

$$(\mathcal{T})_{ijk} = a_i b_j c_k$$

nazveme **tenzorem s hodností 1**, značíme $\text{rank}(\mathcal{T}) = 1$.

Definice 1.2 (Hodnost tenzoru)**.** Nechť $A \in \mathbb{T}^{n_1 \times R}, B \in \mathbb{T}^{n_2 \times R}, C \in \mathbb{T}^{n_3 \times R}$. Pak **hodnost tenzoru** $\mathcal{T} \in \mathbb{T}^{n_1 \times n_2 \times n_3}$ je minimální počet tenzorů s hodností 1 nutných k vyjádření $\mathcal{T}$ pomocí sumace, tj.

$$(\mathcal{T})_{i,j,k} = \sum_{r=1}^{R} A_{i,r} B_{j,r} C_{k,r}. \tag{1.1}$$

Tedy $\text{rank}(\mathcal{T}) = R$ a v souladu s Definicí 1.1 platí $A = [a_1, a_2, \ldots, a_R] \in \mathbb{T}^{n_1 \times R}$, $B = [b_1, b_2, \ldots, b_R] \in \mathbb{T}^{n_2 \times R}$ a $C = [c_1, c_2, \ldots, c_R] \in \mathbb{T}^{n_3 \times R}$. Matice A, B a C nazýváme **faktorové matice** a $\mathcal{T} = [[A, B, C]]$ označuje **kanonický rozklad** tenzoru $\mathcal{T}$.

V další části textu budeme pracovat s reálným tělesem, popřípadě jeho podmnožinami. Další věty a definice proto vyslovíme pro aritmetiku reálného tělesa.

1.2 Tenzor násobení matic

Definice 1.3. Nechť E, F jsou libovolné matice $E \in \mathbb{R}^{P\times Q}$ a $F \in \mathbb{R}^{Q\times R}$ a jejich součin $EF = G \in \mathbb{R}^{P\times R}$. Dále nechť $e = \mathrm{vec}(E^T)$, $f = \mathrm{vec}(F^T)$ a $g = \mathrm{vec}(G)$. Pak násobení matic lze chápat jako bilineární zobrazení $\Phi : \mathbb{R}^{PQ} \times \mathbb{R}^{QR} \to \mathbb{R}^{PR}$ takové, že platí

$$g = \mathrm{vec}(G) \coloneqq \Phi(e, f) \quad \forall E, F.$$

Toto zobrazení lze reprezentovat tenzorem $\mathcal{T}_{PQR} \in \mathbb{R}^{PQ\times QR\times PR}$, který nazveme **tenzorem násobení matic** rozměrů $P \times Q$ a $Q \times R$ a zobrazení Φ lze zapsat ve tvaru

$$\mathrm{vec}(G) = \Phi(e, f) = \mathcal{T}_{PQR} \times_1 \mathrm{vec}(E^T)^T \times_2 \mathrm{vec}(F^T)^T.$$

Pro prvky tenzoru platí

$$[\mathcal{T}_{PQR}]_{ijk} \in \{0, 1\} \qquad \forall i \in \widehat{PQ}, j \in \widehat{QR}, k \in \widehat{PR}.$$

Příkladem může být tenzor násobení matic o rozměrech 2×2, pro který platí $\mathcal{T} \in \mathbb{R}^{4\times4\times4}$ a je tvaru

$$\mathcal{T}_{222} = \left(\begin{array}{cccc|cccc|cccc|cccc} 1&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0\\ 0&0&1&0&0&0&0&0&0&0&0&1&0&0&0&0\\ 0&0&0&0&1&0&0&0&0&0&0&0&0&1&0&0\\ 0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&1 \end{array}\right),$$

kde svislé čáry oddělují vrstvy třetí dimenze.
Předmětem zkoumání v této kapitole jsou zejména čtvercové matice. Proto se při analýze vlastností násobení matic omezíme jen na tuto konkrétní podmnožinu, tedy na $\mathbb{R}^{n\times n}$.

1.2.1 Standardní algoritmus

Nechť $A, B \in \mathbb{R}^{n\times n}$. Potom standardní algoritmus pro výpočet součinu $C = AB$ je

$$\begin{pmatrix} c_{1,1} & c_{1,2} & \dots & c_{1,n} \\ c_{2,1} & c_{2,2} & \dots & c_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ c_{n,1} & c_{n,2} & \dots & c_{n,n} \end{pmatrix} = \begin{pmatrix} a_{1,1} & a_{1,2} & \dots & a_{1,n} \\ a_{2,1} & a_{2,2} & \dots & a_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n,1} & a_{n,2} & \dots & a_{n,n} \end{pmatrix} \begin{pmatrix} b_{1,1} & b_{1,2} & \dots & b_{1,n} \\ b_{2,1} & b_{2,2} & \dots & b_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ b_{n,1} & b_{n,2} & \dots & b_{n,n} \end{pmatrix},$$

kde

$$c_{i,j} = a_{i,1}b_{1,j} + a_{i,2}b_{2,j} + \cdots + a_{i,n}b_{n,j} \quad \forall i, j \in \hat{n}.$$

Takto provedený výpočet vyžaduje n násobení pro každý z n^2 koeficinetů $c_{i,j}$, celkem tedy n^3 operací skalárního násobení. Sčítání provedeme $(n-1)$ pro n^2 koeficientů, ve výsledku se jedná o $n^2(n-1)$ operací. Celková složitost standardního "učebnicového" algoritmu tedy je $O(n^3+n^2(n-1)) = O(2n^3-n^2) \cong O(n^3)$.

Standardní algoritmus je přímou aplikací zobrazení Φ z Definice 1.3. Využití jiných algoritmů, které mají nižší časovou náročnost, ale vyžaduje nalezení rozkladu tenzoru dle vzorce (1.1), jehož faktorové matice odpovídají danému algoritmu.

Při násobení dvou matic rozměru 2×2 standardní algoritmus vyžaduje 8 skalárních násobení a 4 sčítání. Tento konkrétní případ je uveden jako příklad pro srovnání s následujícím algoritmem.

1.2.2 Strassenův algoritmus

Při využití Strassenova algoritmu dojde ke snížení asymptotické složitosti na $O(n^{\log_2 7})$.

Jak již bylo uvedeno, násobení matic $\mathbb{R}^{2\times 2}$ odpovídá tenzoru maticového násobení a má tvar

$$\mathcal{T}_{222} = \left(\begin{array}{cccc|cccc|cccc|cccc} 1&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0\\ 0&0&1&0&0&0&0&0&0&0&0&1&0&0&0&0\\ 0&0&0&0&1&0&0&0&0&0&0&0&0&1&0&0\\ 0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&1 \end{array}\right).$$

Jeho kanonický rozklad $\mathcal{T}_{222} = [[A, B, C]]$ pro Strassenův algoritmus má faktorové matice

$$A = \begin{pmatrix} 1&0&1&0&1&-1&0\\ 0&0&0&0&1&0&1\\ 0&1&0&0&0&1&0\\ 1&1&0&1&0&0&-1 \end{pmatrix},$$

$$B = \begin{pmatrix} 1&1&0&-1&0&1&0\\ 0&0&1&0&0&1&0\\ 0&0&0&1&0&0&1\\ 1&0&-1&0&1&0&1 \end{pmatrix},$$

$$C = \begin{pmatrix} 1&0&0&1&-1&0&1\\ 0&1&0&1&0&0&0\\ 0&0&1&0&1&0&0\\ 1&-1&1&0&0&1&0 \end{pmatrix}.$$

Z těchto faktorových matic lze vyčíst algoritmus vyžadující jen 7 operací skalárního násobení. Zapíšeme-li celou situaci pro násobení matic $E, F \in \mathbb{R}^{2\times 2}$ s prvky $e_{i,j}$, resp. $f_{i,j}$ pomocí součtů a součinů, dostaneme

$$\begin{aligned} m_1 &= (e_{1,1} + e_{2,2})(f_{1,1} + f_{2,2})\\ m_2 &= (e_{2,1} + e_{2,2})f_{1,1}\\ m_3 &= e_{1,1}(f_{1,2} - f_{2,2})\\ m_4 &= e_{2,2}(-f_{1,1} + f_{2,1})\\ m_5 &= (e_{1,1} + e_{1,2})f_{2,2}\\ m_6 &= (-e_{1,1} + e_{2,1})(f_{1,1} + f_{1,2})\\ m_7 &= (e_{1,2} - e_{2,2})(f_{2,1} + f_{2,2}) \end{aligned}$$

$$\begin{aligned} g_{1,1} &= m_1 + m_4 - m_5 + m_7\\ g_{1,2} &= m_3 + m_5\\ g_{2,1} &= m_2 + m_4\\ g_{2,2} &= m_1 - m_2 + m_3 + m_6 \end{aligned}$$

kde $g_{i,j}$ jsou prvky matice $G = EF$.

Jak již bylo zmíněno, asymptotická složitost Strassenova algoritmu je obecně $O(n^{log_2 7}) = O(n^{2.807})$. V uvedeném příkladu je nutné provést 7 skalárních násobení a 18 sčítání, opravdu tedy došlo ke snížení počtu násobení, ale ke zvýšení počtu sčítání.

1.3 Tenzor násobení polynomů

Definice 1.4. Nechť $\mathrm{p}(x) = a_n x^n + \ldots + a_1 x + a_0$, $\mathrm{q}(x) = b_m x^m + \ldots + b_1 x + b_0$ jsou polynomy stupňů n, resp. m, s koeficienty z $\mathbb{R}$. Označme součin těchto polynomů

$$\mathrm{r}(x) = \mathrm{p}(x)\mathrm{q}(x) = c_p x^p + \ldots + c_1 x + c_0, \text{ kde } p = n + m.$$

Dále označme

$$\overline{p} = \begin{pmatrix} a_n \\ a_{n-1} \\ \vdots \\ a_0 \end{pmatrix}, \quad \overline{q} = \begin{pmatrix} b_m \\ b_{m-1} \\ \vdots \\ b_0 \end{pmatrix}, \quad \overline{r} = \begin{pmatrix} c_p \\ c_{p-1} \\ \vdots \\ c_0 \end{pmatrix}.$$

Násobení polynomů pak lze reprezentovat bilineárním zobrazením $\Omega : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^p$ a platí

$$\overline{r} \coloneqq \Omega(\overline{p}, \overline{q}).$$

Toto zobrazení lze reprezentovat tenzorem $\mathcal{P}_{n+1,m+1} \in \mathbb{R}^{(n+1)\times(m+1)\times(n+m-1)}$, které nazveme **tenzorem násobení polynomů** stupňů $n \times m$ a zobrazení Ω je tvaru

$$\overline{r} = \Omega(\overline{p}, \overline{q}) = \mathcal{P}_{n+1,m+1} \times_1 \overline{p}^T \times_2 \overline{q}^T.$$

Pro prvky tenzoru platí

$$[\mathcal{P}_{n+1,m+1}]_{ijk} \in \{0, 1\} \qquad \forall i \in \{0, \ldots, n+1\}, \forall j \in \{0, \ldots, m+1\}, \forall k \in \{0, \ldots, n+m-1\}.$$

Tato práce se bude dále zaměřovat na násobení polynomů stejných stupňů a zkoumání rozkladů tenzorů jim odpovídajících. Proto se, obdobně jako v případě matic, omezíme na polynomy stejných stupňů.

1.3.1 Standardní algoritmus

Opět nejprve připomeneme standardní algoritmus pro násobení polynomů.

Nechť máme dva polynomy n-tého stupně, $\mathrm{p}(x) = a_0 + a_1 x + \cdots + a_n x^n$, $\mathrm{q}(x) = b_0 + b_1 x + \cdots + a_n x^n$. Přepíšeme-li polynomy do tvaru sumy pro lepší přehlednost, jejich součin lze vyjádřit následujícím způsobem:

$$\mathrm{r}(x) = \left(\sum_{i=0}^{n} a_i x^i\right)\left(\sum_{j=0}^{n} b_j x^j\right) = \sum_{i=0}^{2n} x^i \cdot \left(\sum_{\substack{s+t=i \\ s,t\geq 0}} a_s b_t\right) = \sum_{i=0}^{n}\sum_{j=0}^{n} a_i b_j x^{i+j}. \tag{1.2}$$

Určíme počet skalárních operací nutných k výpočtu koeficientů polynomu $\mathrm{r}(x)$. Nejprve je nutné spočítat součiny $a_i b_j$ pro $\forall i, j \in \{0, 1, \cdots, n\}$. K tomu je potřeba $(n+1)^2$ operací násobení. Výsledný polynom má pouze $2n + 1$ koeficientů a proto musíme dále provést $(n + 1)^2 - (2n + 1) = n^2$ sčítání. Celková složitost je rovna $O((n + 1)^2 + n^2) = O((2n^2 + 2n + 1) \cong O(n^2)$.

Stejně jako jsme uvedli již v kapitole 1.2, sofistikovanější algoritmy lze ztotožnit s konkrétním rozkladem na faktorové matice. V další části kapitoly některé z nich uvedeme.

1.3.2 Karatsubův algoritmus

Jedním z efektivnějších nástrojů pro násobení dvou polynomů je Karatsubův algoritmus, který byl poprvé publikován v [5] již v roce 1963. Jednalo se o algoritmus k vynásobení dvou polynomů stupně 1 za využití pouze 3 násobení (namísto standardních 4).
Uvažme tedy dva polynomy stupně 1, $\mathrm{p}(x) = a_1 x + a_0$, $\mathrm{q}(x) = b_1 x + b_0$. S využitím standardního algoritmu spočteme jejich součin, tedy

$$\begin{aligned}\mathrm{r}(x) &= \mathrm{p}(x)\mathrm{q}(x) \\ &= (a_1 x + a_0)(b_1 x + b_0) \\ &= a_1 b_1 x^2 + (a_0 b_1 + a_1 b_0) x + a_0 b_0.\end{aligned}$$

Při tom je třeba využít 4 skalární násobení při určení koeficientů $p_i q_j$ pro $i, j \in \{0, 1\}$ a 1 operaci sčítání k výpočtu $a_0 b_1 + a_1 b_0$.
Označme součiny

$$D_0 = a_0 b_0,\ D_1 = a_1 b_1,\ D_{0,1} = (a_0 + a_1)(b_0 + b_1).$$

Aplikujeme-li Karatsubův algoritmus, s využitím součinů D_0, D_1 a $D_{0,1}$ rozepíšeme polynom r(x)

$$\begin{aligned}\mathrm{r}(x) &= \mathrm{p}(x)\mathrm{q}(x) \\ &= (a_1 x + a_0)(b_1 x + b_0) \\ &= a_1 b_1 x^2 + (a_0 b_1 + a_1 b_0) x + a_0 b_0 \\ &= a_1 b_1 x^2 + [(a_0 + a_1)(b_0 + b_1) - a_0 b_0 - a_1 b_1] x + a_0 b_0 \\ &= D_1 x^2 + (D_{0,1} - D_0 - D_1) x + D_0\end{aligned}$$

K získání výsledného polynomu je tedy nutné napočítat součiny D_0, D_1 a $D_{0,1}$, k tomu je třeba 3 skalární násobení a 2 operace sčítání. Další 2 součty jsou potřeba k sečtení $D_{0,1} - D_0 - D_1$. Celkem tedy 3 operace násobení, 4 operace sčítání, tudíž byl počet násobení snížen o 1 oproti standardnímu algoritmu.

Oba algoritmy, standardní i Karatsubův, zapišme pomocí faktorových matic rozkladu tenzoru násobení polynomů stupně 1, který je tvaru

$$\mathcal{P}_{2,2} = \left(\begin{array}{cc|cc|cc} 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \end{array}\right).$$

Rozklad odpovídající standardnímu algoritmu

$$A = \begin{pmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{pmatrix} \qquad C = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix}$$
$$B = \begin{pmatrix} 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \end{pmatrix}$$

se oproti Karatsubovu algoritmu

$$A = B = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \end{pmatrix} \quad C = \begin{pmatrix} 1 & 0 & 0 \\ -1 & 1 & -1 \\ 0 & 0 & 1 \end{pmatrix}$$

liší v hodnosti (tedy v počtu skalárních násobení), což jsme již zanalyzovali v předchozím odstavci. Postup, jakým lze algoritmus zapsat pomocí faktorových matic, bude uveden v další části této kapitoly.

Karatsubův algoritmus lze zobecnit i pro polynomy vyšších stupňů. Vhodným uspořádáním do součinů D_i a $D_{j,k}$ lze vždy vylepšit standardní postup pro násobení polynomů ve smyslu snížení počtu násobení. I tento zobecněný postup budeme nadále označovat jako Karatsubův algoritmus a zavedeme pro něj zkratku KA. Následující věta byla převzána z [2].

Věta 1.5 (Karatsubův algoritmus)**.** Uvažujme polynomy n-tých stupňů $\mathrm{p}(x) = a_0 + a_1 x + \cdots + a_n x^n$, $\mathrm{q}(x) = b_0 + b_1 x + \cdots + b_n x^n$. Označme součiny

$$D_i \coloneqq a_i b_i \qquad \forall i \in 0, 1, \ldots, n$$
$$D_{s,t} \coloneqq (a_s + a_t)(b_s + b_t) \qquad \forall i \in 1, \ldots, 2n-1;\ (\forall s, t \in \mathbb{N})\,(s + t = i,\ 0 \le s < t \le n)$$

Nechť $\mathrm{r}(x)$ je součinem polynomů $\mathrm{p}(x)$ a $\mathrm{q}(x)$ a jeho koeficienty označme c_i, tedy

$$\mathrm{r}(x) = \mathrm{p}(x)\mathrm{q}(x) = \sum_{i=0}^{2n} c_i x^i.$$

Pak pro koeficienty c_i platí:

$$c_0 = D_0$$
$$c_{2n} = D_n$$
$$c_i = \begin{cases} \sum\limits_{\substack{s+t=i \\ n \ge t > s \ge 0}} D_{s,t} - \sum\limits_{\substack{s+t=i \\ n \ge t > s \ge 0}} (D_s + D_t) & \text{pro i lichá, } 0 < i < 2n \\ \sum\limits_{\substack{s+t=i \\ n \ge t > s \ge 0}} D_{s,t} - \sum\limits_{\substack{s+t=i \\ n \ge t > s \ge 0}} (D_s + D_t) + D_{i/2} & \text{pro i sudá, } 0 < i < 2n \end{cases}$$

Důkaz. Nejprve provedeme ověření pro liché koeficienty. Rozepišme koeficienty c_i ve tvaru (1.2)

$$c_i = \sum_{\substack{s+t=i \\ n \ge s, t \ge 0}} a_s b_t.$$

Nyní uvažme sumu koeficientů $D_{s,t}$ takových, že $s + t = i$ a $n \geq t > s \geq 0$. Rozepíšeme-li ji, získáme

$$\begin{aligned}\sum_{\substack{s+t=i\\ n\geq t>s\geq 0}} D_{s,t} &= \sum_{\substack{s+t=i\\ n\geq t>s\geq 0}} (a_s + a_t)(b_s + b_t) = \sum_{\substack{s+t=i\\ n\geq t>s\geq 0}} (a_s b_s + a_s b_t + a_t b_s + a_t b_t)\\ &= \sum_{\substack{s+t=i\\ n\geq t>s\geq 0}} (a_s b_t + a_t b_s) \;+ \sum_{\substack{s+t=i\\ n\geq t>s\geq 0}} (a_s b_s + a_t b_t)\\ &= \sum_{\substack{s+t=i\\ n\geq t>s\geq 0}} a_s b_t \;+ \sum_{\substack{s+t=i\\ n\geq t>s\geq 0}} a_t b_s \;+ \sum_{\substack{s+t=i\\ n\geq t>s\geq 0}} a_t b_t \;+ \sum_{\substack{s+t=i\\ n\geq t>s\geq 0}} a_s b_s\end{aligned}$$

To je možné přepsat díky jednoduché úpravě sum, pro které platí:

$$\sum_{\substack{s+t=i\\ n\geq t>s\geq 0}} a_s b_t = \sum_{\substack{s+t=i\\ n\geq s>t\geq 0}} a_t b_s \implies \sum_{\substack{s+t=i\\ n\geq t>s\geq 0}} (a_s b_t + a_t b_s) = \sum_{\substack{s+t=i\\ n\geq t,s\geq 0}} a_s b_t$$

Upravme dále předchozí výraz:

$$\begin{aligned}\sum_{\substack{s+t=i\\ n\geq t>s\geq 0}} D_{s,t} &= \sum_{\substack{s+t=i\\ n\geq t,s\geq 0}} a_s b_t \;+ \sum_{\substack{s+t=i\\ n\geq t>s\geq 0}} (a_s b_s + a_t b_t)\\ &= c_i \;+ \sum_{\substack{s+t=i\\ n\geq t>s\geq 0}} (D_s + D_t)\end{aligned}$$

Jednoduchou úpravou rovnice dostáváme definiční vztah pro c_i, kdy i je liché.
Pro sudé i musíme zohlednit členy $a_s b_s = D_s$ pro $s = t = i/2$, který je obsažen v sumě c_i. Rozepišme tedy vztah pro i sudé:

$$\begin{aligned}\sum_{\substack{s+t=i\\ n\geq t>s\geq 0}} D_{s,t} &= \sum_{\substack{s+t=i\\ n\geq t,s\geq 0}} a_s b_t \;+ \sum_{\substack{s+t=i\\ n\geq t>s\geq 0}} (a_s b_s + a_t b_t)\\ &= (c_i - D_{i/2}) \;+ \sum_{\substack{s+t=i\\ n\geq t>s\geq 0}} (D_s + D_t)\end{aligned}$$

Stejně jako v předchozím případě, i teď získáme definiční vztah pro c_i.
Zbývá tedy jen rozmyslet případy c_0 a c_{2n}, ale jde pouze o krajní případy koeficientů c_i pro sudé i. □

Počet násobení odpovídá počtu koeficientů D_i a $D_{s,t}$. Počet D_i je $n + 1$, jelikož je třeba napočítat pro $\forall i \in \{0, \ldots, n\}$. Čísla $D_{s,t} = (a_s + a_t)(b_s + b_t)$ určíme pro každou dvojici $s, t \in \hat{n}$, kterých je

$$\binom{n+1}{2} = \frac{(n+1)(n)}{2} = \frac{n^2 + n}{2}.$$

Celkový počet násobení je tedy

$$(n+1) + \frac{n^2 + n}{2} = \frac{n^2 + 3n + 2}{2}.$$

1.3.2.1 Rekurzivní aplikace a *dummy coefficients*

Algoritmus lze dále pro vhodná n zefektivnit využitím rekurzivní aplikace KA [2]. Uvážíme-li například polynom p(x) n-tého stupně, kde n je liché, lze zapsat ve tvaru

$$\mathrm{p}(x) = \sum_{i=0}^{n} a_i x^i = A_1(x) x^{\frac{n+1}{2}} + A_2(x)\,,$$

kde $A_1(x), A_2(x)$ jsou polynomy řádu $\frac{n-1}{2}$ se stejným počtem koeficientů $\frac{n+1}{2}$.

Při součinu polynomů, které lze zapsat obdobným způsobem, tedy že jejich řád $n-1$ můžeme vyjádřit ve tvaru součinu $n = u \cdot v$ (n je počet koeficientů a_i), nejprve zapíšeme polynom v analogickém tvaru a aplikujeme Karatsubův algoritmus. Na získané součiny D_i a $D_{s,t}$ pak aplikujeme Karatsubův algoritmus znovu. Tento postup lze zobecnit i pro polynomy řádu $\left(\prod_{i=1}^{k} n_i - 1\right)$ s použitím k-té rekurze.
Při násobení polynomů, jejichž počet koeficientů n je prvočíselný, nelze výše uvedené zlepšení použít. Je tedy třeba použít klasický přístup KA a dochází tak k jevu, při kterém výpočet součinu polynomů řádu o jedno vyšší (s počtem koeficientů $n+1$) vyžaduje méně operací než součin těchto polynomů. K odstranění tohoto jevu lze použít tzv. metodu nepravých koeficientů (z ang. *dummy coefficients*), která spočívá v přičtení členu $a_n x^n$, pro který ale platí $a_n = 0$. Pak už lze využít rekurzivní postup, ze kterého vynecháme veškeré násobení týkající se nulového koeficientu a_n. Pak jistě platí, že počet operací bude menší a dojde k odstranění nežádoucího jevu.
Další vylepšení a rozšíření KA jsou prezentována v [1], pro některá n se jedná o značné snížení počtu násobení.

Při zkoumání KA je stěžejní počet koeficientů polynomů. Proto budeme v další části textu pro polynom s nenulovými koeficienty značit počet koeficientů polynomu n, pokud nebude uvedeno jinak.

1.3.3 Srovnání algoritmů

Porovnejme nyní Karatsubův algoritmus (1.5) včetně vylepšení zmíněných v [1] a [2] se standardním algoritmem pro malá n. V tabulce uvádíme počet koeficientů polynomů n, počet skalárních násobení (*#MUL*), počet sčítání (*#ADD*) a násobení konstantou ze $\mathbb{Z}$ nutných k vynásobení polynomů (*#C*).

	Standardní algoritmus		KA [1.5]		KA[2]		[1]		
n	#MUL	#ADD	#MUL	#ADD	#MUL	#ADD	#MUL	#ADD	#C
2	4	1	3	4	3	4	3	4	0
3	9	4	6	13	6	13	6	13	0
4	16	9	10	27	9	24	9	24	0
5	25	16	15	46	15	46	13	86	20
6	36	25	21	71	18	59	17	119	19
7	49	36	28	99	24	85	22	165	37
8	64	49	36	133	27	100	27	100	0

Všechny algoritmy uvedené v tabulce odpovídají rozkladům nad $\mathbb{Z}$. To znamená, že prvky faktorových matic jsou z množiny $\mathbb{Z}$, ve většině případů dokonce z množiny $\{-1, 0, 1\}$. KA a algoritmy z něj vycházející dosahují oproti standardnímu významného snížení počtu násobení. Ten je ale kompenzován

zvýšením počtu sčítání, v případě [1] i zvýšením násobení konstantou $c \in \mathbb{Z}$ takovou, že $|c| > 1$.
Významnou roli hrají polynomy s počtem koeficientů

$$n = 2^k \quad \forall k \in \mathbb{N}, \text{ pro něž je } \#\text{MUL} = 3^k.$$

Právě tyto polynomy jsou důvodem, že asymptotická složitost KA je $O(n^{log_2 3}) = O(n^{1.585})$.

Algoritmus 1.6 (Přepis algoritmu násobení polynomů do faktorových matic)**.** Nechť máme dva polynomy n-tého stupně, $\mathrm{p}(x) = a_0 + a_1 x + \cdots + a_n x^n$, $\mathrm{q}(x) = b_0 + b_1 x + \cdots + a_n x^n$. Nechť G je algoritmus na násobení polynomů, který udává součin těchto polynomů ve tvaru

$$\mathrm{r}(x) = \mathrm{p}(x)\mathrm{q}(x) = \sum_{i=0}^{2n} h_i x^i,$$

kde h_i jsou tvaru

$$h_i = \sum_{k=1}^{R} \gamma_{i,k} D_k \quad \forall \gamma_{i,k} \in \mathbb{Z}, \ \forall i \in \widehat{2n}.$$

Číslo R značí celkový počet skalárních součinů potřebných k vypočtení součinu polynomů a dále platí

$$D_j = \left(\alpha_{j,0}a_0 + \alpha_{j,1}a_1 + \ldots + \alpha_{j,n}a_n\right)\left(\beta_{j,0}b_0 + \beta_{j,1}b_1 + \ldots + \beta_{j,n}b_n\right)$$

kde $\alpha_{j,k}, \beta_{j,k} \in \mathbb{Z}, \ \forall j \in \hat{m}, \ \forall k \in \hat{n}$.
Potom lze algoritmus G zapsat do faktorových matic tak, že odpovídá rozkladu tenzoru $\mathcal{P}_{n+1,n+1}$ s hodností $\mathrm{rank}(\mathcal{P}_{n+1,n+1}) = R$.

Ověření algoritmu. Faktorové matice mají rozměr

$$A, B \in \mathbb{R}^{(n+1)\times m}, \ C \in \mathbb{R}^{(2n+1)\times m}.$$

Matice A a B musí splňovat rovnost

$$\left(A^T \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_n \end{pmatrix}\right) * \left(B^T \begin{pmatrix} b_0 \\ b_1 \\ \vdots \\ b_n \end{pmatrix}\right) = \begin{pmatrix} D_1 \\ D_2 \\ \vdots \\ D_m \end{pmatrix},$$

kde symbolem * je označen Hadamardův součin. Tento vztah vede na soustavu rovnic

$$D_j = \left(A_{1,j}a_0 + A_{2,j}a_1 + \ldots A_{(n+1),j}a_n\right)\left(B_{1,j}b_0 + B_{2,j}b_1 + \ldots B_{(n+1),j}b_n\right)$$

pro $\forall j \in \hat{m}$. Také ale platí, že D_j lze napsat ve tvaru

$$D_j = \left(\alpha_{j,0}a_0 + \alpha_{j,1}a_1 + \ldots + \alpha_{j,n}a_n\right)\left(\beta_{j,0}b_0 + \beta_{j,1}b_1 + \ldots + \beta_{j,n}b_n\right)$$

kde $\alpha_{j,k}, \beta_{j,k} \in \mathbb{Z}, \ \forall j \in \hat{m}, \ \forall k \in \hat{n}$.
Pak pouhým porovnáním koeficientů u a_j pro $\forall j \in n$ přiřadíme prvkům matic A a B hodnoty

$$A_{j,i} = \alpha_{i,(j-1)} \text{ a } B_{j,i} = \beta_{i,(j-1)} \quad \forall i \in \hat{m}, \ \forall j \in \widehat{n+1}$$

Matice C splňuje rovnost

$$C\begin{pmatrix} D_1 \\ D_2 \\ \vdots \\ D_m \end{pmatrix} = \begin{pmatrix} h_0 \\ h_1 \\ \vdots \\ h_{2n+1} \end{pmatrix}.$$

Opět rozepíšeme do soustavy rovnic a porovnáme

$$h_j = \left(C_{j,1}D_1 + C_{j,2}D_2 + \dots C_{j,m}D_m\right) \stackrel{!}{=} \sum_{k=1}^{m} \gamma_{j,k}D_k$$

pro $\forall\, j \in \widehat{2n+1}$. Prvky matice C tedy určuje rovnost

$$C_{i,j} = \gamma_{i,j} \quad \forall i \in \widehat{2n+1},\ \forall\, j \in \hat{m}$$

Příklad 1.7. Algoritmus 1.6 demonstrujeme na Karatsubově algoritmu pro násobení polynomů stupně 2. Uvažujme dva polynomy, $p(x) = a_0 + a_1x + a_2x^2$, $q(x) = b_0 + b_1x + a_2x^2$. Nalezneme koeficienty $D_i, D_{s,t}, c_i$ z Karatsubova algoritmu, tedy:

$$\begin{array}{ll} D_0 = a_0b_0 & h_0 = D_0 \\ D_1 = a_1b_1 & h_1 = D_{0,1} - D_0 - D_1 \\ D_2 = a_2b_2 & h_2 = D_{0,2} - D_0 - D_2 + D_1 \\ D_{0,1} = (a_0 + a_1)(b_0 + b_1) & h_3 = D_{1,2} - D_1 - D_2 \\ D_{0,2} = (a_0 + a_2)(b_0 + b_2) & h_4 = D_2 \\ D_{1,2} = (a_1 + a_2)(b_1 + b_2) & \end{array}$$

Faktorové matice A, B mají rozměr 3×6, matice C je 5×6. S využitím vztahů pro prvky matic z Algoritmu 1.6 určíme faktorové matice

$$A = B = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{pmatrix} \qquad C = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ -1 & 1 & 0 & -1 & 0 & 0 \\ -1 & 0 & 1 & 1 & 0 & -1 \\ 0 & 0 & 0 & -1 & 1 & -1 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}.$$

1.3.4 Toom-Cookův algoritmus

Tento algoritmus je založen na faktu, že k určení koeficientů polynomu stupně n jej stačí vyčíslit v $n + 1$ bodech. Odvození lze nalézt v [11], popřípadě [3].
Uvažme dva polynomy stupně $\mathrm{p}(x) = a_0 + a_1x + \cdots + a_nx^n$, $\mathrm{q}(x) = b_0 + b_1x + \cdots + a_nx^n$. Pak rozklad tenzoru maticového násobení $\mathcal{P}_{n+1,n+1,2n+1}$ nalezneme s hodností $\mathrm{rank}(\mathcal{P}_{n+1,n+1,2n+1}) = 2n + 1$. Volme $2n + 1$ libovolných různých čísel $\mu_j \in \mathbb{C}$ a sestrojme faktorové matice A a B

$$A = B = \begin{pmatrix} 1 & 1 & \cdots & 1 & 1 \\ \mu_1 & \mu_2 & \cdots & \mu_{2n} & \mu_{2n+1} \\ \mu_1^2 & \mu_2^2 & \cdots & \mu_{2n}^2 & \mu_{2n+1}^2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \mu_1^n & \mu_2^n & \cdots & \mu_{2n}^n & \mu_{2n+1}^n \end{pmatrix}.$$

Matici C pak dopočítáme ze soustavy rovnic

$$(\mathcal{T})_{i,j,k} = \sum_{r=1}^{R} A_{i,r} B_{j,r} C_{k,r}$$

a máme

$$C = \left(\begin{pmatrix} 1 & 1 & \cdots & 1 & 1 \\ \mu_1 & \mu_2 & \cdots & \mu_{2n} & \mu_{2n+1} \\ \mu_1^2 & \mu_2^2 & \cdots & \mu_{2n}^2 & \mu_{2n+1}^2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \mu_1^{2n} & \mu_2^{2n} & \cdots & \mu_{2n}^{2n} & \mu_{2n+1}^{2n} \end{pmatrix}^T \right)^{-1}.$$

Na základě Toom-Cookova algoritmu tedy tvrdíme, že hodnost tenzoru nad $\mathbb{R}$ je nejvýše $2n + 1$. Matice C^T je Vandermondova matice a pro velká n může být špatně podmíněná, zvláště v reálném oboru. Jelikož ale čísla μ_i lze volit libovolně, můžeme zajistit, aby matice C byla dobře podmíněná, například rovnoměrnou volbou μ_i na jednotkové kružnici v komplexní rovině. Pak lze výpočet provést pomocí rychlé Fourierovy transformace se složitostí $O(n \log n)$.

1.4 Symetrie rozkladu tenzoru

Symetrie rozkladu tenzoru násobení polynomů je velmi důležitá vlastnost. Na jejím základě lze potom na rozklad uvalit dodatečný předpoklad, který významně ulehčí hledání nových rozkladů.

Symetrický tenzor je invariantní vůči permutaci některých indexů. Konkrétně tenzor násobení polynomů má tuto záměnnost v 1. a 2. složce, tedy platí

$$\mathcal{P}_{i,j,k} = \mathcal{P}_{j,i,k} \quad \forall i, j, k.$$

Kvůli této invarianci lze rozpoznat částečnou nebo úplnou symetrii rozkladu tenzoru.

Definice 1.8. Mějme symetrický tenzor $\mathcal{T}$ invariantní vůči permutaci dvou indexů. Pak řekneme, že rozklad $\mathcal{T} = [[A, B, C]]$ je *částečně symetrický*, pokud platí

$$A = B \ \vee \ B = C \ \vee \ A = C.$$

Definice 1.9. Mějme symetrický tenzor $\mathcal{T}$ invariantní vůči permutaci ve všech třech indexech. Pak řekneme, že rozklad $\mathcal{T} = [[A, B, C]]$ je *úplně symetrický*, pokud platí

$$A = B = C.$$

V příkladech jsme již ukázali, že rozklad tenzoru není jednoznačný, jelikož jsme k danému tenzoru nalezli rozklady s jinou hodností, popřípadě dva různé rozklady stejné hodnosti. Díky tomuto faktu lze pro některé tenzory nalézt pro určitou hodnost jak symetrický rozklad, tak i nesymetrický. Například tenzor, pro který existuje úplně symetrický i nesymetrický rozklad, platí

$$\mathcal{T} = [[\tilde{A}, \tilde{A}, \tilde{A}]] = [[A, B, C]].$$

Původní idea, zvaná Comonova hypotéza, přepokládala, že každý tenzor $\mathcal{T}$, pro který existuje nesymetrický rozklad s hodností $\text{rank}(\mathcal{T}) = R$, existuje také symetrický rozklad se stejnou hodností. Tato

myšlenka byla ale vyvrácena protipříkladem tenzoru [15], jehož nesymetrická reprezentace má menší hodnost než úplně symetrický rozklad. V praxi ale většinou platí, že symetrický a nesymetrický rozklad mají stejnou hodnost.

Invariance tenzoru násobení polynomů umožňuje pouze částečnou symetrii rozkladu. Na základě předchozího odstavce proto lze tuto symetrii přepokládat a hledat rozklad tenzoru $\mathcal{T}$ ve tvaru

$$\mathcal{T} = [[A, A, C]].$$

1.5 GF(2)

Obecně lze nalézt rozklad tenzoru na faktorové matice nad $\mathbb{R}$, jak bylo uvedeno v (1.3.4), ale i nad $\mathbb{Z}$ nebo na množině $\{-1, 0, 1\}$ (1.3.3). Výhodou rozkladu nad $\mathbb{R}$ je, že jeho hodnost je stejná nebo nižší než je hodnost libovolného rozkladu nad $\mathbb{Z}$. Naším cílem ovšem je nalézt rozklady odpovídající algoritmu, který potřebuje co nejmenší počet skalárních násobení a rozklad nad $\mathbb{R}$ nám do algoritmu vnáší násobení konstantou z $\mathbb{R}$, které je mnohem složitější než násobení celým číslem. Způsob nalezení rozkladu nad $\mathbb{Z}$ v této práci spočívá ve využití tělesa GF(2), které bude objasněno v další části této sekce.

Definice 1.10 (Těleso GF(2))**.** Nechť T je dvouprvková množina {0,1}. Dále definujeme operace sčítání a násobení na množině T následovně:

$$\forall a, b \in \mathrm{T}: \begin{array}{ll} a \oplus b = a + b & (\mathrm{mod}\ 2) \\ a \odot b = a * b & (\mathrm{mod}\ 2) \end{array}$$

Pak těleso $(T, \oplus, \odot)$ nazveme GF(2), *Galois field*.

Poznámka 1.11. Operace na GF(2) lze ekvivalentně definovat pomocí binárních logických operací. Násobení odpovídá konjunkci (AND), sčítání odpovídá exkluzivní disjunkci (XOR). To je ilustrováno na pravdivostních tabulkách:

a	b	$a \wedge b$
0	0	0
0	1	0
1	0	0
1	1	1

a	b	$a \veebar b$
0	0	0
0	1	1
1	0	1
1	1	0

Lemma 1.12. Nechť A, B a C jsou faktorové matice rozkladu $\mathcal{T} = [[A, B, C]]$ nad $\mathbb{Z}$. Pak z nich lze sestrojit faktorové matice rozkladu $\mathcal{T} = [[\tilde{A}, \tilde{B}, \tilde{C}]]$ nad tělesem GF(2).

Důkaz. Nejprve sestrojíme nové faktorové matice $\tilde{A}, \tilde{B}, \tilde{C}$ a poté ověříme, že se jedná o rozklad $\mathcal{T}$. Definujme tedy

$$\begin{array}{ll} \tilde{A} \coloneqq A & (\mathrm{mod}\ 2), \\ \tilde{B} \coloneqq B & (\mathrm{mod}\ 2), \\ \tilde{C} \coloneqq C & (\mathrm{mod}\ 2). \end{array}$$

Tyto nové matice mají jednotlivé prvky pouze z množiny $\{0, 1\}$.

Rovnost $\mathcal{T} = [[\tilde{A}, \tilde{B}, \tilde{C}]]$ ukážeme pomocí ověření vzorce $(\mathcal{T})_{i,j,k} = \sum_{r=1}^{R} \tilde{A}_{i,r}\tilde{B}_{j,r}\tilde{C}_{k,r}$ pro prvky tenzoru $\mathcal{T}$. Prvky faktorových matic $\tilde{A}, \tilde{B}, \tilde{C}$ vyjádříme vztahy

$$\begin{aligned} \tilde{A}_{i,j} &= A_{i,j} \pmod 2, \\ \tilde{B}_{i,j} &= B_{i,j} \pmod 2, \\ \tilde{C}_{i,j} &= C_{i,j} \pmod 2. \end{aligned}$$

Dosazením do vzorce dostaneme

$$\begin{aligned} \sum_{r=1}^{R} \tilde{A}_{i,r}\tilde{B}_{j,r}\tilde{C}_{k,r} &= \sum_{r=1}^{R} \Big(A_{i,r} \pmod 2\Big)\Big(B_{j,r} \pmod 2\Big)\Big(C_{k,r} \pmod 2\Big) \\ &= \sum_{r=1}^{R} \Big(A_{i,r}B_{j,r}A_{k,r} \pmod 2\Big) \\ &= \sum_{r=1}^{R} A_{i,r}B_{j,r}A_{k,r} \pmod 2 \\ &= (\mathcal{T})_{i,j,k}\,, \end{aligned}$$

kde poslední rovnost plyne z faktu, že $\sum_{r=1}^{R} A_{i,r}B_{j,r}A_{k,r} \in \{0, 1\}$, a tedy operace $\pmod 2$ nezmění hodnotu prvku tenzoru $\mathcal{T}$.

□

Pokud tedy máme rozklad nad $\mathbb{Z}$, lze jeho faktorové matice snadno převést nad GF(2), tj. platí

$$\text{existence rozkladu nad } \mathbb{Z} \implies \text{existence rozkladu nad GF(2).}$$

Vyvstává proto otázka, zda by tento proces šel nějakým způsobem obrátit a byla by zaručena platnost i opačné implikace. Tato implikace sice obecně neplatí, tedy

$$\text{existence rozkladu nad GF(2)} \not\Longrightarrow \text{existence rozkladu nad } \mathbb{Z},$$

ale existují takové rozklady, které lze převést nad $\mathbb{Z}$. V další části této práce si nejprveme ukážeme, jakým způsobem lze hledat rozklad nad GF(2) a poté, pokud to je možné, jakým způsobem jej převést nad $\mathbb{Z}$.

Kapitola 2

Splnitelnost booleovských formulí

Ve smyslu Sekce 1.5 se snažíme o nalezení rozkladu nad GF(2) a tedy nalezení jeho faktorových matic. Jejich prvky jsou vázány rovnicemi vyplývajícími z 1.1. Takovou soustavu rovnic lze sice řešit přímo například pomocí dosazovací metody, ale pro větší úlohy se ukazuje jako neefektivní. V následující tabulce lze nahlédnout počty proměnných (x) a nezávislých rovnic (#eq), které úloha generuje v závislosti na hodnosti tenzoru $\mathcal{T}$.

n	rank	x	#eq
2	3	15	9
3	6	48	30
4	9	99	70
5	13	182	135
6	17	289	231
7	22	440	364
8	27	621	540

Jelikož se ale jedná o soustavu nad GF(2), lze ji převést na úlohu splnitelnosti booleovských proměnných. Algoritmus, kterým tento převod lze uskutečnit, bude prezentován později. Nejprve je ale popsána úloha splnitelnosti boolevských formulí a jsou diskutovány některé zajímavé vlastnosti.

Problém splnitelnosti soustavy booleovských formulí je úloha teoretické informatiky, jejímž řešením je logický výraz TRUE (1) nebo FALSE (0), tedy vyjádření, zda je soustava splnitelná, či nikoliv. Navazující úlohou, za předpokladu splnitelnosti soustavy, je nalézení alespoň jednoho konkrétního řešení.

2.1 Vlastnosti boolevských formulí

Definice 2.1. Číselnou proměnnou x nazveme **booleovskou** právě tehdy, když $x \in \{0, 1\}$. Označme libovolnou booleovskou proměnnou jako **booleovskou formuli** (ekvivalentně lze užít i název logická formule). Pak pokud F je booleovská formule, pak i $\neg F$ je booleovská formule (často značeno $\overline{F}$). Pokud F a G jsou booleovské formule, pak $(F \vee G)$ a $(F \wedge G)$ jsou také booleovské formule.
Výrazem Var(F) rozumíme všechny booleovské proměnné, které formule F obsahuje.

Definice 2.2. Přiřazením α rozumíme přiřazení hodnot $\alpha_1, \ldots, \alpha_k$ boolevským proměnným $x_1, \ldots, x_k$, značíme

$$\alpha = \{x_1 = a_1, x_2 = a_2, \ldots, x_k = a_k\},$$

kde $a_1, a_2, \ldots, a_k \in \{0, 1\}$.
Nechť je α přiřazení, pak lze α aplikovat na booleovskou formuli F, označíme

$$F\alpha = F\{x_1 = a_1, x_2 = a_2, \ldots, x_k = a_k\}.$$

Výrazem Var(α) rozumíme všechny booleovské proměnné, které patří do přiřazení α.

Přiřazením α k formuli F dojde k eliminaci některých boolevských proměnných. Pokud navíc platí

$$\text{Var}(F) \subset \text{Var}(\alpha),$$

dojde k eliminaci všech proměnných booleovské formule F a tedy k jejímu vyřešení.

Definice 2.3. Booleovskou formuli F nazveme **splnitelnou**, pokud existuje přiřazení α takové, že $F\alpha = 1$. V opačném případě nazveme F **nesplnitelnou**.

Definice 2.4. Nechť $x_1, x_2, \ldots, x_n$ jsou boolevské proměnné. Pak booleovská formule F je v **algebraické normální formě** (označíme $F \in ANF$), pokud ji lze zapsat ve tvaru

$$F = C_1 \veebar C_2 \veebar \cdots \veebar C_m,$$

kde $\veebar$ značí exkluzivní disjunkci.
Jednotlivé formule C_i nazveme klauzule, které jsou tvaru

$$C_i = (\, u_{i,1} \wedge u_{i,2} \wedge \cdots \wedge u_{i,k}) \,,$$

kde

$$u_{i,j} \in \{x_1, x_2, \ldots, x_n\}.$$

Definice 2.5. Booleovská formule F je v **konjunktivní normální formě** (označíme $F \in CNF$, z ang. *conjunctive normal form*), pokud ji lze zapsat ve tvaru

$$F = C_1 \wedge C_2 \wedge \cdots \wedge C_m.$$

Jednotlivé formule C_i nazveme klauzule, které jsou tvaru

$$C_i = (\, u_{i,1} \vee u_{i,2} \vee \cdots \vee u_{i,k}) \,,$$

kde

$$u_{i,j} \in \{x_1, x_2, \ldots, x_n\} \cup \{\overline{x}_1, \overline{x}_2, \ldots, \overline{x}_n\}.$$

Povšimněme si, že v jednotlivých klauzulích ANF nejsou povoleny negace proměnných, zatímco v klauzulích CNF se nacházet mohou.

Poznámka 2.6. Při zápisu formule F je běžné užívat + namísto $\vee$ a dále v klauzulích C_i psát konjunkce ve tvaru algebraického součinu, tj.

$$u_{i,1} \wedge u_{i,2} \wedge \cdots \wedge u_{i,k} = u_{i,1}u_{i,2} \ldots u_{i,k}.$$

Důvod tohoto upraveného zápisu je prostý. Jelikož se při řešení pohybujeme na tělese GF(2), můžeme soustavu booleovských formulí považovat za soustavu algebraických rovnic, jak bylo objasněno v Definici 1.10 a Poznámce 1.11.

V následující části této kapitoly nastíníme princip, jakým lze transformovat úlohu ve formě ANF do tvaru CNF, a tudíž převést úlohu řešení soustavy algebraických rovnic na problém splnitelnosti booleovských formulí.

2.2 SAT solvery

Satisfiability solver, zkráceně SAT solver, je program schopný vyhodnotit splnitelnost soustavy booleovských formulí ve tvaru CNF, popřípadě nalézt konkrétní řešení. Jejich účinnost se na vybraných úlohách testuje během každoroční souteže *SAT Competition*.

V rámci této práce byly využity vybrané SAT solvery. Nejužitečnějším SAT solverem, který byl využit v rámci této práce a který byl klíčový při hledání řešení soustavy CNF, je software MiniSat [13]. Všechna řešení, která budou dalé prezentována, byla získaná s jeho využitím, konkrétně ve verzi Minisat v1.14. Dalším z nich je CryptoMiniSat [14], který je implementován v softwaru Bosphorus [12]. Tento software se při hledání řešení neukázal jako optimální, jelikož s jeho využitím nebyla nalezena řešení, která byla poté získána pomocí MiniSatu. Software Bosphorus se ale stal důležitou součástí při řešení problému, protože jednou z jeho implementovaných funkcí je převod soustavy ANF na soustavu CNF. Při tomto procesu jsou do soustavy vloženy pomocné proměnné, které sice zajišťují přepis jednotlivých formulí, ale zvětšují celkový počet rovnic v soustavě.

2.3 Převod soustavy rovnic na SAT problém a jeho řešení

Uvažme tedy problém nalezení rozkladu tenzoru nad GF(2) s využitím SAT solverů.
Následující algoritmus udává způsob, kterým lze vytvořit soustavu algebraických rovnic, jejímž řešením nalezneme prvky faktorových matic rozkladu tenzoru.

Data: tenzor $\mathcal{T}$, požadovaná hodnost R
Result: faktorové matice A, B, C, soustava s
$[n_1, n_2, n_3] = \text{size}(\mathcal{T})$;
symbolické proměnné $x_{i,l}, y_{j,l}, z_{k,l} \quad \forall i \in \hat{n_1}, \forall j \in \hat{n_2}, \forall k \in \hat{n_3}, \forall l \in \hat{R}$;
for $i \in \hat{n_1}, l \in \hat{R}$ **do**
  $A_{i,l} = x_{i,l}$
end
for $j \in \hat{n_2}, l \in \hat{R}$ **do**
  $B_{j,l} = y_{j,l}$
end
for $k \in \hat{n_3}, l \in \hat{R}$ **do**
  $C_{k,l} = z_{k,l}$
end
$(\mathcal{T}_1)_{i,j,k} = \sum_{r=1}^{R} A_{i,r} B_{j,r} C_{k,r}$;
s = vec $(\mathcal{T}_1)$ + vec $(\mathcal{T})$;

Máme-li soustavu algebraických rovnic nad GF(2), lze takovou soustavu ztotožnit se soustavou ve tvaru ANF. Dále pak využijeme software Bosphorus, který tuto soustavu ANF převede na CNF, jejíž řešení získáme s pomocí softwaru Minisat. Celý tento proces ilustruje následující příklad.

Příklad 2.7. Demonstrujeme algoritmus při hledání faktorových matic pro rozklad tenzoru $\mathcal{P}_{2,2}$ násobení polynomů stupně 1, pro jehož rozměry platí $\mathcal{P}_{2,2} \in \mathbb{R}^{2,2,3}$ a hodnost $\operatorname{rank}(\mathcal{P}_{2,2}) = 3$. Rozměry faktorových matic jsou tedy $A, B \in \mathbb{R}^{2,3}$, $C \in \mathbb{R}^{3,3}$.
Nejprve sestavíme faktorové matice, jejichž prvky jsou symbolické proměnné $x_1, x_2, \ldots, x_{15}$. Předpokládáme symetrický rozklad, a tedy že matice A a B jsou identické.

$$A = B = \begin{pmatrix} x_1 & x_3 & x_5 \\ x_2 & x_4 & x_6 \end{pmatrix}, \qquad C = \begin{pmatrix} x_7 & x_{10} & x_{13} \\ x_8 & x_{11} & x_{14} \\ x_9 & x_{12} & x_{15} \end{pmatrix}.$$

Poté aplikujeme vzorec $(\mathcal{T}_1)_{i,j,k} = \sum_{r=1}^{R} A_{i,r} B_{j,r} C_{k,r}$ a sestrojíme tenzor $\mathcal{T}_1$. Požadujeme rovnost $\mathcal{T}_1 = \mathcal{P}_{2,2}$ a jelikož jsme na tělese GF(2), přičtením jednotlivých prvků $\mathcal{T}_1$ k tenzoru $\mathcal{P}_{2,2}$ získáme soustavu rovnic. Z předpokladu rovnosti matic A a B vyplývá, že některé rovnice budou duplicitní a proto jsou ze soustavy odstraněny.

$$\begin{aligned}
&x_7 \cdot x_1^2 + x_{10} \cdot x_3^2 + x_{13} \cdot x_5^2 + 1 = 0 \\
&x_1 \cdot x_2 \cdot x_7 + x_3 \cdot x_4 \cdot x_{10} + x_5 \cdot x_6 \cdot x_{13} = 0 \\
&x_7 \cdot x_2^2 + x_{10} \cdot x_4^2 + x_{13} \cdot x_6^2 = 0 \\
&x_8 \cdot x_1^2 + x_{11} \cdot x_3^2 + x_{14} \cdot x_5^2 = 0 \\
&x_1 \cdot x_2 \cdot x_8 + x_3 \cdot x_4 \cdot x_{11} + x_5 \cdot x_6 \cdot x_{14} + 1 = 0 \\
&x_8 \cdot x_2^2 + x_{11} \cdot x_4^2 + x_{14} \cdot x_6^2 = 0 \\
&x_9 \cdot x_1^2 + x_{12} \cdot x_3^2 + x_{15} \cdot x_5^2 = 0 \\
&x_1 \cdot x_2 \cdot x_9 + x_3 \cdot x_4 \cdot x_{12} + x_5 \cdot x_6 \cdot x_{15} = 0 \\
&x_9 \cdot x_2^2 + x_{12} \cdot x_4^2 + x_{15} \cdot x_6^2 + 1 = 0
\end{aligned}$$

Jelikož požadujeme splnění soustavy nad GF(2), platí $x_i^2 = x_i$ pro libovolné $i \in \{1, 2, \cdots, 15\}$, takže ze soustavy odstraníme druhé mocniny.

$$\begin{aligned}
&x_7 \cdot x_1 + x_{10} \cdot x_3 + x_{13} \cdot x_5 + 1 = 0 \\
&x_1 \cdot x_2 \cdot x_7 + x_3 \cdot x_4 \cdot x_{10} + x_5 \cdot x_6 \cdot x_{13} = 0 \\
&x_7 \cdot x_2 + x_{10} \cdot x_4 + x_{13} \cdot x_6 = 0 \\
&x_8 \cdot x_1 + x_{11} \cdot x_3 + x_{14} \cdot x_5 = 0 \\
&x_1 \cdot x_2 \cdot x_8 + x_3 \cdot x_4 \cdot x_{11} + x_5 \cdot x_6 \cdot x_{14} + 1 = 0 \\
&x_8 \cdot x_2 + x_{11} \cdot x_4 + x_{14} \cdot x_6 = 0 \\
&x_9 \cdot x_1 + x_{12} \cdot x_3 + x_{15} \cdot x_5 = 0 \\
&x_1 \cdot x_2 \cdot x_9 + x_3 \cdot x_4 \cdot x_{12} + x_5 \cdot x_6 \cdot x_{15} = 0 \\
&x_9 \cdot x_2 + x_{12} \cdot x_4 + x_{15} \cdot x_6 + 1 = 0
\end{aligned}$$

Nyní je soustava ve tvaru ANF, kterou převede Bosphorus na tvar CNF. Soustava ve formátu CNF má pak tvar

−17 2 0	−17 8 0	17 − 2 − 8 0	−18 4 0
−18 11 0	18 − 4 − 11 0	−19 6 0	−19 14 0
19 − 6 − 14 0	17 18 19 0	−17 − 18 19 0	−17 18 − 19 0
17 − 18 − 19 0	−20 2 0	−20 3 0	−20 8 0
20 − 2 − 3 − 8 0	−21 4 0	−21 5 0	−21 11 0
21 − 4 − 5 − 11 0	−22 6 0	−22 7 0	−22 14 0
22 − 6 − 7 − 14 0	−20 21 22 0	20 − 21 22 0	20 21 − 22 0
−20 − 21 − 22 0	−23 3 0	−23 8 0	23 − 3 − 8 0
−24 5 0	−24 11 0	24 − 5 − 11 0	−25 7 0
−25 14 0	25 − 7 − 14 0	−23 24 25 0	23 − 24 25 0
23 24 − 25 0	−23 − 24 − 25 0	−26 2 0	−26 9 0
26 − 2 − 9 0	−27 4 0	−27 12 0	27 − 4 − 12 0
−28 6 0	−28 15 0	28 − 6 − 15 0	−26 27 28 0
26 − 27 28 0	26 27 − 28 0	−26 − 27 − 28 0	−29 2 0
−29 3 0	−29 9 0	29 − 2 − 3 − 9 0	−30 4 0
−30 5 0	−30 12 0	30 − 4 − 5 − 12 0	−31 6 0
−31 7 0	−31 15 0	31 − 6 − 7 − 15 0	29 30 31 0
−29 − 30 31 0	−29 30 − 31 0	29 − 30 − 31 0	−32 3 0
−32 9 0	32 − 3 − 9 0	−33 5 0	−33 12 0
33 − 5 − 12 0	−34 7 0	−34 15 0	34 − 7 − 15 0
−32 33 34 0	32 − 33 34 0	32 33 − 34 0	−32 − 33 − 34 0
−35 2 0	−35 10 0	35 − 2 − 10 0	−36 4 0
−36 13 0	36 − 4 − 13 0	−37 6 0	−37 16 0
37 − 6 − 16 0	−35 36 37 0	35 − 36 37 0	35 36 − 37 0
−35 − 36 − 37 0	−38 2 0	−38 3 0	−38 10 0
38 − 2 − 3 − 10 0	−39 4 0	−39 5 0	−39 13 0
39 − 4 − 5 − 13 0	−40 6 0	−40 7 0	−40 16 0
40 − 6 − 7 − 16 0	−38 39 40 0	38 − 39 40 0	38 39 − 40 0
−38 − 39 − 40 0	−41 3 0	−41 10 0	41 − 3 − 10 0
−42 5 0	−42 13 0	42 − 5 − 13 0	−43 7 0
−43 16 0	43 − 7 − 16 0	41 42 43 0	−41 − 42 43 0
−41 42 − 43 0	41 − 42 − 43 0,		

kde

$$k \sim x_{k-1},\ -k \sim \overline{x_{k-1}} \quad \forall k \in \{2, \ldots, 43\}$$

a 0 značí konec boolovské formule. Proměnné x_k pro $k > 16$ jsou pomocné proměnné. Na vyřešení této soustavy ve tvaru CNF využijeme Minisat a dostaneme řešení

$$\text{sol} = \left[\, 0\,1\,1\,1\,1\,0\,0\,1\,1\,0\,1\,0\,1\,1\,0\,0\,0\,1\,0\,0\,0\,0\,0\,0\,0\,1\,1\,0\,1\,0\,1\,1\,0\,0\,0\,0\,0\,0\,0\,1\,0\,0 \,\right],$$

které řeší soustavu CNF.

Posledním krokem je substituce řešení za původní symbolické proměnné $x_1, x_2, \ldots, x_{15}$. Dostaneme tak matice

$$A = B = \begin{pmatrix} 0 & 1 & 1 \\ 1 & 1 & 0 \end{pmatrix}, \qquad C = \begin{pmatrix} 0 & 0 & 1 \\ 1 & 1 & 1 \\ 1 & 0 & 0 \end{pmatrix}.$$

Předchozí příklad uvádí, jak lze nalézt faktorové matice nad GF(2). Algoritmus lze ještě upravit uvalením předpokladů na symetrie faktorových matic, které rozebereme v následující kapitole.

2.4 Převod rozkladu z GF(2) na $\mathbb{Z}$

K převedení faktorových matic z GF(2) na $\mathbb{Z}$ použijeme následující algoritmus.

Data: tenzor násobení polynomů $\mathcal{P}$, hodnost R = rank($\mathcal{P}$), faktorové matice A, B, C rozkladu $\mathcal{P}$ nad GF(2)
Result: soustava s
$[n_1, n_2, n_3]$ = size($\mathcal{P}$) ;
symbolické proměnné $x_{i,l}, y_{j,l}, z_{k,l} \quad \forall i \in \hat{n_1}, \forall j \in \hat{n_2}, \forall k \in \hat{n_3}, \forall l \in \hat{R}$;
for $i \in \hat{n_1}, j \in \hat{n_2}, l \in \hat{R}$ **do**
  if $A_{i,l} = 1$ **then**
    $\tilde{A}_{i,l} = x_{i,l}$
  else
    $\tilde{A}_{i,l} = 0$
  end
  if $B_{j,l} = 1$ **then**
    $\tilde{B}_{j,l} = y_{j,l}$
  else
    $\tilde{B}_{j,l} = 0$
  end
end
for $k \in \hat{n_3}, l \in \hat{R}$ **do**
  $\tilde{C}_{k,l} = z_{k,l}$
end
$(\mathcal{T}_1)_{i,j,k} = \sum_{r=1}^{R} \tilde{A}_{i,r}\tilde{B}_{j,r}\tilde{C}_{k,r}$;
s = vec $(\mathcal{T}_1)$ − vec $(\mathcal{T})$;

Ze známých rozkladů nad $\mathbb{Z}$ je patrné, že pro matice A, B platí $A_{i,j}, B_{i,j} \in \{-1, 0, 1\}$. Tuto vlastnost proto předpokládáme a tudíž v maticích A, B požadujeme změny jen v některých znaménkách. Pro elementy matice C platí $C_{i,j} \in \mathbb{Z}$, nahrazujeme tedy celou matici C symbolickými proměnnými.
Výstup algoritmu je soustava rovnic s pro symbolické proměnné $x_{i,l}, y_{j,l}, z_{k,l}$, kterou řešíme dosazovací metodou. Výsledkem pak je rovnice (popřípadě soustava), s několika volnými proměnnými, které položíme rovny takovým celým číslům, že splňují zmíněnou rovnici (soustavu). Poté zpětně dopočítáme ostatní proměnné.
Úskalí této metody spočívá ve volbě volných proměnných. Takové přidělení hodnot ze $\mathbb{Z}$ nemusí vůbec existovat, nebo i přes splnění rovnice (soustavy) narazíme na problém, že zpětným dosazením do soustavy mají některé proměnné hodnoty z jiné množiny, například z $\mathbb{Q}$.

Kapitola 3

Získané rozklady

Výsledky, které jsou publikovány v rámci této práce, se týkají pouze rozkladů pro násobení polynomů a byly dosaženy metodou popsanou v Kapitole 2. Hodnost těchto rozkladů je srovnatelná s rozklady publikovanými v [2], ale jedná se pouze o rozklady nad GF(2).
Vyznamným faktorem pro tyto výsledky jsou různé konfigurace symetrických sloupců ve faktorových maticích.

3.1 Symetrie faktorových matic

Abychom snížili počet proměnných, uvažujeme ve faktorových maticích různé typy symetrie. Ty jsou například patrné i v rozkladech odpovídajících Karatsubovu algoitmu.

Definice 3.1. Nechť A je matice $A \in \mathbb{R}^{n\times m}$. Pak definujeme $\overline{A}$ převrácenou matici k A vztahem

$$\left[\overline{A}\right]_{ij} := A_{(n+1-i),j}. \tag{3.1}$$

Definice 3.2. Řekneme, že matice A má **backward symetrii** právě tehdy, když existuje taková permutační matice P, že platí

$$\overline{A} = AP$$

Je patrné, že některé sloupce se po převrácení zobrazují samy na sebe, jiné se v páru s jiným sloupcem zobrazují cyklicky.

Definice 3.3. Nechť $A \in \mathbb{R}^{n\times R}$ je matice s backward symetrií. Řekneme, že sloupec $A_{:,j}$ má symetrii **self-backward**, pokud platí

$$A_{:,j} = \overline{A_{:,j}}.$$

Řekneme, že sloupce $A_{:,k}$ a $A_{:,l}$ mají **cyklickou** backward symetrii, platí-li

$$A_{:,k} = \overline{A_{:,l}} \ \wedge \ A_{:,l} = \overline{A_{:,k}}.$$

Definice 3.4. Nechť $\mathcal{T}$ je tenzor násobení polynomů a $\mathcal{T} = [[A, B, C]]$ je jeho kanonický rozklad na faktorové matice. Pak řekneme, že rozklad tenzoru $\mathcal{T}$ má **backward symetrii** právě tehdy, když platí

$$\mathcal{T} = [[\overline{A}, \overline{B}, \overline{C}]].$$

3.1.1 Předpoklad pevných sloupců a řádků

Při předpokladu, že matice $A = B \in \mathbb{R}^{n,R}$, uvažujeme dva pevné sloupce v maticích A a B a dva pevné řádky v matici C. Označme $T_{A1}, T_{A2} \in \mathbb{R}^{n,1}$ první dva sloupce ve faktorové matici A. Pak

$$[T_{A1}, T_{A2}] = \begin{pmatrix} 1 & 0 \\ 0 & 0 \\ \vdots & \vdots \\ 0 & 0 \\ 0 & 1 \end{pmatrix}.$$

Analogicky označme $T_{C1}, T_{C2} \in \mathbb{R}^{1,R}$ první dva řádky ve faktorové matici C. Pak

$$[T_{C1}, T_{C2}] = \begin{pmatrix} 1 & 0 & \dots & 0 & 0 \\ 0 & 1 & \dots & 0 & 0 \end{pmatrix}.$$

Jelikož jsou faktorové matice invariantní vůči permutaci sloupců, resp. řádků, lze T_{A1}, T_{A2}, resp. T_{C1}, T_{C2} umístit na libovolnou pozici. Musí ale platit

$$A_{:,j} = T_{A1} \implies C_{j,:} = T_{C1}$$

a stejně i pro T_{A2} a T_{C2}.
Fakt, že se tyto sloupce, resp. řádky v maticích nachází, plyne z násobení polynomů, které tyto matice reprezentují. Uvážíme-li dva polynomy stupně n, pak absolutní člen $a_0 b_0$ a koeficient u n-tého členu $a_n b_n$ přesně odpovídají těmto řádkům a sloupcům. Odvození lze nalézt v [1].

Definice 3.5. Nechť máme kanonický rozklad $\mathcal{T} = [[A, B, C]]$, jeho faktorové matice mají backward symetrii a obsahují pevné sloupce a řádky z 3.1.1. Pak definujeme **konfiguraci symetrie** vztahem

$$S_c(\mathcal{T}) := (R_1, R_2),$$

kde

$$R_1 = \#\{j \mid A_{:,j} \text{ má symetrii self-backward}\},\ R_2 = \frac{\#\{k, l \mid A_{:,k} \text{ a } A_{:,l} \text{ má cyklickou backward symetrii}\}}{2}.$$

Celková hodnost tenzoru pak je $\operatorname{rank}(\mathcal{T}) = R = 2 + R_1 + 2R_2$.

3.2 Výsledky

V následující tabulce je zaznamenáno, pro jaké konkrétní podmínky symetrie byly nalezeny rozklady nad GF(2) za využití SAT solverů. V tabulce je zachycena hodnost, s jakou byl tenzor rozložen, dále pak počet sloupců se symetrií self-backward a počet párů sloupců s cyklickou backward symetrií. Všechny rozklady v tabulce byly navíc hledány za předpokladu pevných řádku a sloupců. Zachováme-li značení ze Sekce 3.5, je pak celková hodnost $R = 2 + R_1 + 2R_2$.
Klíčovým krokem je řešení soustavy CNF, jejímž vyřešením byl prakticky nalezen rozklad nad GF(2). Tyto pojmy proto můžeme ztotožnit a nalezení řešení nazvat nalezením rozkladu. Dále je v tabulce uveden počet původních proměnných *x* soustavy algebraických rovnic a počet všechn proměnných *Var* včetně pomocných, které při převedení vytvořil software Bosphorus. Dalším údajem v tabulce je celková velikost soustavy CNF (*size*), tj. počet klauzulí. Sloupec *status* označuje, zda byl rozklad nalezen (SAT), byla prokázána jeho neexistence pro konkrétní konfiguraci symetrií (UNSAT), nebo nebyl nalezen v čase větším než 15000s (UNF). Dále je v tabulce uvedena doba *time*, za jakou SAT solver vyřešil úlohu, resp. po jaké výpočetní době byl software zastaven.

n	rank	R_1	R_2	x	Var	size	status	time(s)
5	12	0	5	109	1369	10808	UNSAT	93
	12	2	4	110	1244	10332	UNSAT	207
	12	4	3	111	1119	9890	UNSAT	0
	12	6	2	112	994	9418	UNSAT	0
	12	8	1	113	869	8931	UNSAT	0
	13	1	5	119	1421	11193	UNSAT	2000
	13	3	4	120	1296	10717	SAT	1
	13	5	3	121	1171	10254	UNSAT	0
	13	7	2	122	1046	9793	UNSAT	0
6	16	0	7	186	3399	28386	UNF	20265
	16	2	6	186	3129	27360	UNF	22250
	16	4	5	186	2859	26334	UNF	19217
	16	6	4	186	2589	25308	UNSAT	265
	16	8	3	186	2319	24339	UNSAT	0
	17	1	7	198	3465	28989	UNF	22896
	17	3	6	198	3195	27963	SAT	183
	17	5	5	198	2925	26937	UNF	30307
	17	7	4	198	2655	25911	UNSAT	663
	17	9	3	198	2385	24929	UNSAT	0
7	21	3	8	299	6789	60412	UNF	18897
	21	5	7	300	6394	58894	UNF	15292
	21	7	6	301	5999	57376	UNF	15620
	21	9	5	302	5604	55858	UNF	15611
	22	2	9	313	7309	63646	UNF	15157
	22	4	8	314	6914	62128	SAT	683
	22	6	7	315	6519	60610	SAT	12238
	22	8	6	316	6124	59092	UNF	15406
	22	10	5	317	5729	57574	UNF	15378

Konkrétní nalezené rozklady nad GF(2) jsou uvedeny v další části této kapitoly.

3.2.1 Násobení polynomů $n \in \{2, 3, 4\}$

Pro nízké stupně polynomů již není příliš prostoru pro snížení hodnosti. Rozklady odpovídající Karatsubovu algoritmu z [2] zatím nebyly zlepšeny a nepovedlo se to ani při vypracovávání této práce. Všechny tyto výsledky se však podařilo replikovat, ať už ve stejné formě, nebo kvůli nejednoznačnosti v podobě nového rozkladu. Počet operací byl ale stejný, a proto tyto rozklady nejsou příliš významné. Jelikož jsme některé již uvedli v podobě příkladů, nebudeme je prezentovat.

3.2.2 Násobení polynomů $n = 5$

Rozklad tenzoru $\mathcal{P}_{5,5}$ na rank $(\mathcal{P}_{5,5}) = 13$, $S_c(\mathcal{P}_{5,5}) = (3, 4)$.

$$A = B = \begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 \end{pmatrix}$$

$$C = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Tento rozklad se v rámci této práce nepodařilo rozšířit nad $\mathbb{Z}$, známe ale jiný rozklad získaný z [1], který má stejnou konfiguraci symetrie.

$$A = B = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 1 & -1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & -1 & -1 & 0 & -1 & 1 \\ 0 & 0 & 0 & 1 & 0 & -1 & 1 & -1 & 0 & -1 & -1 & -1 & 1 \end{pmatrix}$$

$$C = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & -1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 2 & 0 & 1 & -1 & -1 & 0 & -1 & 0 & 0 & 1 & 0 & 0 \\ -2 & -2 & 0 & -3 & 0 & 3 & -1 & 2 & 0 & 1 & -2 & -1 & 1 \\ 3 & 1 & 1 & 3 & 1 & -4 & 1 & -1 & -1 & -2 & 2 & 2 & -1 \\ -3 & 0 & -2 & -2 & -1 & 3 & 0 & 0 & 2 & 1 & -1 & -2 & 1 \\ 1 & 0 & 2 & 1 & 0 & -1 & -1 & 0 & -1 & 0 & 0 & 1 & 0 \\ 0 & 0 & -1 & -1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

3.2.3 Násobení polynomů $n = 6$

Rozklad tenzoru $\mathcal{P}_{6,6}$ na rank $(\mathcal{P}_{6,6}) = 17$, $S_c(\mathcal{P}_{6,6}) = (3, 6)$.

$$A = B = \begin{pmatrix} 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{pmatrix}$$

$$C = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Tento rozklad se v rámci této práce nepodařilo rozšířit nad $\mathbb{Z}$, opět ale známe jiný rozklad získaný z [1], který má stejnou konfiguraci symetrie.

$$A = B = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & -1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & -1 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & -1 & -1 & 0 & 1 & 1 \end{pmatrix}$$

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

3.2.4 Násobení polynomů $n = 7$

Rozklad tenzoru $\mathcal{P}_{7,7}$ na rank $(\mathcal{P}_{7,7}) = 22$, $S_c(\mathcal{P}_{7,7}) = (4, 8)$.

$$A = B = \begin{pmatrix}
1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\
0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 \\
0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\
0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1
\end{pmatrix}$$

$$C = \begin{pmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\
0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\
1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\
1 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 1 \\
1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 \\
0 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\
0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{pmatrix}$$

Tento rozklad se v rámci této práce nepodařilo rozšířit nad $\mathbb{Z}$, ale získaný rozklad nad GF(2) odpovídá rozkladu z [1]. Uvádíme tedy tento rozklad, který je rozšířením předchozího z GF(2) na $\mathbb{Z}$, opomeneme-li permutaci sloupců, vůči které je ale rozklad invariantní.

$$A = B = \begin{pmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & -1 & 0 & 1 & 0 & 0 & -1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & -1 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & -1 & 0 & -1 & -1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 1 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & -1 & -1 & 1 & 0 & -1 & -1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 1 & 1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & -1 & -1 & 1 & 1
\end{pmatrix}$$

$$C = \begin{pmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
-1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & -1 & 1 & 0 & 0 & 0 & 1 & 0 & -1 & 0 & 0 & -1 & 1 & 1 & 0 & -1 & 0 & 0 & 0 & 1 & 0 \\
1 & -1 & 1 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 \\
-3 & 1 & -2 & -1 & -3 & 0 & -2 & -2 & -2 & 2 & 1 & 1 & 4 & -4 & -1 & 2 & 0 & 1 & 1 & 0 & -5 & 1 \\
3 & 0 & 3 & 2 & 3 & 0 & 3 & 3 & 3 & -2 & -2 & -2 & -6 & 6 & 1 & -2 & -1 & -1 & -2 & 1 & 6 & -1 \\
-2 & 0 & -3 & -1 & -2 & 1 & -3 & -2 & -2 & 1 & 2 & 1 & 4 & -5 & 0 & 1 & 1 & 0 & 2 & -1 & -4 & 1 \\
0 & 0 & 1 & -1 & 1 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 1 & 0 & -1 & 0 & -1 & 1 & 0 & 0 & 0 & -1 & 0 & 1 & 1 & 0 \\
0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{pmatrix}$$

Rozklad tenzoru $\mathcal{P}_{7,7}$ na rank $(\mathcal{P}_{7,7}) = 22$, $S_c(\mathcal{P}_{7,7}) = (6, 7)$.

$$A = B = \begin{pmatrix}
1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 1 \\
0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\
0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\
0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 \\
0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 1 \\
0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1
\end{pmatrix}$$

$$C = \begin{pmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 1 \\
1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 1 \\
0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\
1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 \\
1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\
1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\
1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{pmatrix}$$

Rozklad nad $\mathbb{Z}$ se nalézt nezdařilo. Pro tento typ konfigurační symetrie se ani nepodařilo dohledat takový rozklad v dostupné literatuře.

Závěr

Tato práce se zaměřuje na seznámení čtenáře s problematikou tenzorů násobení matic a polynomů a jejich rozklady. Dále uvádí doposud užívané algoritmy určené k násobení matic a polynomů s důrazem na počet operací násobení nutných k výpočtu součinu matic, resp. polynomů. Tyto algoritmy pak dává do souvislosti s rozklady tenzorů a jejich faktorovými maticemi.
Důležitým přínosem tohoto textu je prezentace a testování metody hledání rozkladu tenzorů pomocí SAT solverů. Ta využívá převodu soustavy algebraických rovnic na úlohu splnitelnosti booleovských formulí. Její účinnost byla testována na tenzorech násobení polynomů s 5, 6 a 7 koeficienty. Tyto rozklady se podařilo nalézt se srovnatelnou hodností jako ty uvedené v [1], ale pouze nad tělesem GF(2). Jejich převedení nad $\mathbb{Z}$ bylo řešeno pomocí eliminační metody prezentované v Sekci 2.4. Její úskalí spočívá ve velikosti soustavy rovnic, jejíž řešení hledáme.
Také je zde publikována přehledová tabulka pro různé konfigurace symetrií shrnující jejich existenci. Ta také obsahuje podrobnosti o velikosti úloh řešených pomocí SAT solveru a jim odpovídajícím výpočetních časech.

Literatura

[1] P. L. Montgomery: *Five, six, and seven-term Karatsuba-like formulae.* IEEE Transactions on Computers, 54(3), 362–369, 2005

[2] A. Weimerskirch, C. Paar: *Generalizations of the Karatsuba Algorithm for Efficient Implementations.* IACR Cryptology ePrint Archive, 2003.

[3] T. Kováč: *Kanonický rozklad tenzorů maticového násobení.* Bakalářská práce, 2018.

[4] U. Schöning, J. Torán: *The Satisfiability Problem: Algorithms and Analyses.* Lehmanns Media, 2013.

[5] A. Karatsuba, Yu. Ofman: *Multiplication of Multidigit Numbers on Automata.* Soviet Physics Doklady, Vol. 7, p.595, 1963.

[6] V. Strassen: *Gaussian elimination is not optimal.* Numer. Math., 13: 354–356, 1969.

[7] D. Coppersmith, S. Winograd: *Matrix multiplication via arithmetic progressions.* Journal of Symbolic Computation., 9: 251–280, 1990.

[8] A. M. Davie, A. J. Stothers: *Improved bound for complexity of matrix multiplication.* Proceedings of the Royal Society of Edinburgh, 143A (2): 351–370, 2013.

[9] V. Vassilevska Williams: *Breaking the Coppersmith-Winograd barrier.* Proceedings of the 44th Symposium on Theory of Computing, STOC '12, 2012.

[10] F. Le Gall: *Powers of tensors and fast matrix multiplication.* Proceedings of the 39th International Symposium on Symbolic and Algebraic Computation, ISSAC, 2014.

[11] M. Bodrato: *Towards Optimal Toom-Cook Multiplication for Univariate and Multivariate Polynomials in Characteristic 2 and 0.* Proceedings of the WAIFI'07 ser. Lecture Notes in Computer Science, vol. 4547, pp. 119-136, 2007.

[12] D. Choo, M. Soos, K. M. A. Chai, K. S. Meel: *BOSPHORUS: Bridging ANF and CNF Solvers.* 2019 Design, Automation & Test In Europe Conference & Exhibition (DATE), 2019.

[13] N. Sörensson, N. Een: *MiniSat v1.13 – A SAT Solver with Conflict-Clause Minimization.*, SAT Competition 2005, URL: `http://minisat.se/downloads/MiniSat_v1.13_short.pdf`.

[14] M. Soos, K. Nohl, C. Castelluccia: *Extending SAT Solvers to Cryptographic Problems.* Proceedings of the 12th International Conference on Theory and Applications of Satisfiability Testing, 2009.

[15] Y. Shitov: *A counterexample to Comon's conjecture.* ArXiv:1705.08740 e-prints, 2017, URL: `https://epubs.siam.org/doi/pdf/10.1137/17M1131970`.