Algorithmentheorie

02 – Polynomprodukt und Fast Fourier Transformation

Prof. Dr. S. Albers
Prof. Dr. Th. Ottmann

Polynomprodukt

Berechnung des Produkts zweier Polynome *p, q* vom Grade < *n*

p,q Grad *n-1*, *n* Koeffizienten

↓ **Auswertung:** $\omega_{2n}^{0}, \omega_{2n}^{1}, \ldots, \omega_{2n}^{2n-1}$

$2n$ Punkt/Wertpaare $\left(\omega_{2n}^{i}, p\left(\omega_{2n}^{i}\right)\right)$ und $\left(\omega_{2n}^{i}, q\left(\omega_{2n}^{i}\right)\right)$

↓ **Punktweise Multiplikation**

$2n$ Punkt/Wertpaare $\left(\omega_{2n}^{i}, pq\left(\omega_{2n}^{i}\right)\right)$

↓ **Interpolation**

pq Grad $2n$-2, $2n$-1 Koeffizienten

Diskrete Fourier Transformation

n-te komplexe Haupteinheitswurzel: $\omega_n = e^{2\pi i / n}$

$$DFT_n(p) = (p(\omega_n^0), p(\omega_n^1), \ldots, p(\omega_n^{n-1}))$$

Fast Fourier Transformation:
Berechung von $DFT_n(p)$ mittels eines Divide-and Conquer Ansatzes

Diskrete Fourier Transformation

Ansatz: (*n* sei gerade)

$$
\begin{aligned}
p(x) &= a_0 + a_1 x + \ldots + a_{n-1} x^{n-1} \\
&= a_0 + a_2 x^2 + \ldots + a_{n-2} x^{n-2} + \\
&\quad a_1 x + a_3 x^3 + \ldots + a_{n-1} x^{n-1} \\
&= a_0 + a_2 x^2 + \ldots + a_{n-2} \left(x^2\right)^{(n-2)/2} + \\
&\quad x\left(a_1 + a_3 x^2 + \ldots + a_{n-1} \left(x^2\right)^{(n-2)/2}\right) \\
&= p_0\left(x^2\right) + x p_1\left(x^2\right)
\end{aligned}
$$

$$p_0(x) = a_0 + a_2 x + \ldots + a_{n-2} x^{(n-2)/2}$$

$$p_1(x) = a_1 + a_3 x + \ldots + a_{n-1} x^{(n-2)/2}$$

Diskrete Fourier Transformation

Auswertung für $k = 0, \ldots, n-1$

$$p\left(\omega_n^k\right) = \begin{cases} p_0\left(\omega_{n/2}^k\right) + \omega_n^k p_1\left(\omega_{n/2}^k\right), \\ \text{falls } k < n/2 \\ p_0\left(\omega_{n/2}^{k-n/2}\right) - \omega_n^{k-n/2} p_1\left(\omega_{n/2}^{k-n/2}\right) \\ \text{falls } k \geq n/2 \end{cases}$$

$$\begin{aligned} DFT_n(p) &= (p_0(\omega_{n/2}^0), \ldots, p_0(\omega_{n/2}^{n/2-1}), p_0(\omega_{n/2}^0), \ldots, p_0(\omega_{n/2}^{n/2-1})) \\ &+ (\omega_n^0 p_1(\omega_{n/2}^0), \ldots, \omega_n^{n/2-1} p_1(\omega_{n/2}^{n/2-1}), \\ &\quad -\omega_n^0 p_1(\omega_{n/2}^0), \ldots, -\omega_n^{n/2-1} p_1(\omega_{n/2}^{n/2-1})) \end{aligned}$$

6. Fast Fourier Transformation

Algorithmus *FFT*

Input: Ein Array *a* mit den *n* Koeffizienten eines Polynoms *p* und $n = 2^k$

Output: $DFT_n(p)$

1. **if** $n = 1$ **then** /* *p* ist konstant */
2. **return** *a*
3. $d^{[0]} = FFT([a_0, a_2, \ldots, a_{n-2}], n/2)$
4. $d^{[1]} = FFT([a_1, a_3, \ldots, a_{n-1}], n/2)$
5. $\omega_n = e^{2\pi i/n}$
6. $\omega = 1$
7. **for** $k = 0$ **to** $n/2 - 1$ **do** /* $\omega = \omega_n^k$ */
8. $d_k = d_k^{[0]} + \omega \cdot d_k^{[1]}$
9. $d_{k+n/2} = d_k^{[0]} - \omega \cdot d_k^{[1]}$
10. $\omega = \omega_n \cdot \omega$
11. **return** *d*

FFT : Beispiel

$$p(x) = 3x^3 - 15x^2 + 18x + 0$$

a = [0, 18,-15, 3]

$a^{[0]}$ = [0, -15] $a^{[1]}$ = [18, 3]

FFT([0, -15], 2) = (FFT([0],1) + FFT([-15],1), FFT([0],1) - FFT([-15],1))
= (-15,15)

FFT([18, 3],2) = (FFT([18],1) + FFT([3],1), FFT([18],1) - FFT([3],1))
= (21,15)

$k = 0$; $\omega = 1$

d_0 = -15 + 1 * 21 = 6 d_2 = -15 – 1 * 21 = -36

$k = 1$; $\omega = i$

d_1 = 15 + i*15 d_3 = 15 – i*15

FFT(a, 4) = (6, 15+15i, -36, 15 –15i)

7. Analyse

T(n) = Zeit um ein Polynom vom Grad < n an den Stellen $\omega_{2n}^{0}, \omega_{2n}^{1}, \ldots, \omega_{2n}^{2n-1}$ auszuwerten.

T(1) = O(1)

T(n) = 2 T(n/2) + O(n)

= O(n log n)

Polynomprodukt

Berechnung des Produkts zweier Polynome *p, q* vom Grade < *n*

p,q Grad *n-1*, *n* Koeffizienten

↓ **Auswertung durch FFT:** $\omega_{2n}^0, \omega_{2n}^1, \ldots, \omega_{2n}^{2n-1}$

2n Punkt/Wertpaare $\left(\omega_{2n}^i, p\left(\omega_{2n}^i\right)\right)$ und $\left(\omega_{2n}^i, q\left(\omega_{2n}^i\right)\right)$

↓ **Punktweise Multiplikation**

2n Punkt/Wertpaare $\left(\omega_{2n}^i, pq\left(\omega_{2n}^i\right)\right)$

↓ **Interpolation**

pq Grad *2n-2*, *2n-1* Koeffizienten

Interpolation

Umrechnung der Punkt/Wert-Darstellung in die Koeffizientendarstellung

Gegeben: $(x_0, y_0), \ldots, (x_{n-1}, y_{n-1})$ mit $x_i \neq x_j$, für alle $i \neq j$

Gesucht: Polynom p mit Koeffizienten $a_0, \ldots, a_{n-1}$,
so dass

$$
\begin{aligned}
p(x_0) &= a_0 + a_1 x_0 + \ldots + a_{n-1} x_0^{n-1} &= y_0 \\
p(x_1) &= a_0 + a_1 x_1 + \ldots + a_{n-1} x_1^{n-1} &= y_1 \\
p(x_2) &= a_0 + a_1 x_2 + \ldots + a_{n-1} x_2^{n-1} &= y_2 \\
&\vdots & \vdots \\
p(x_{n-1}) &= a_0 + a_1 x_{n-1} + \ldots + a_{n-1} x_{n-1}^{n-1} &= y_{n-1}
\end{aligned}
$$

Interpolation

Matrixschreibweise

$$\begin{pmatrix} 1 & x_0 & \cdots & x_0^{n-1} \\ 1 & x_1 & \cdots & x_2^{n-1} \\ & & \vdots & \\ 1 & x_{n-1} & \cdots & x_{n-1}^{n-1} \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_{n-1} \end{pmatrix} = \begin{pmatrix} y_0 \\ y_1 \\ \vdots \\ y_{n-1} \end{pmatrix}$$

Interpolation

Gleichungssystem

$$\begin{pmatrix} 1 & x_0 & \cdots & x_0^{n-1} \\ 1 & x_1 & \cdots & x_1^{n-1} \\ & & \vdots & \\ 1 & x_{n-1} & \cdots & x_{n-1}^{n-1} \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_{n-1} \end{pmatrix} = \begin{pmatrix} y_0 \\ y_1 \\ \vdots \\ y_{n-1} \end{pmatrix}$$

ist lösbar für $x_i \neq x_j$, für alle $i \neq j$

Spezialfall (hier) : $x_i = \omega_n^i$

Definition:

$$V_n = \left(\omega_n^{ij}\right)_{i,j}, \quad a = (a_i), \quad y = (y_i)$$

$$\mathrm{V}_{\mathrm{n}} a = y \quad \Rightarrow \quad a = V_n^{-1} y$$

Interpolation

Satz

Für alle $0 \le i, j \le n - 1$ gilt:

$$\left(V_n^{-1}\right)_{ij} = \frac{\omega_n^{-ij}}{n}$$

Beweis

$$V_n^{-1} = \left(\frac{\omega_n^{-ij}}{n}\right)_{i,j}$$

zu zeigen:

$$V_n^{-1} V_n = I_n$$

Interpolation

Betrachte Eintrag von $V_n^{-1} V_n$ in Zeile i, Spalte j:

$$\left(V_n^{-1} V_n\right)_{ij} =$$

$$\begin{pmatrix} & & \cdots & \\ \frac{1}{n} & \frac{\omega_n^{-i}}{n} & \cdots & \frac{\omega_n^{-i(n-1)}}{n} \\ & & \vdots & \\ & & \cdots & \end{pmatrix} \begin{pmatrix} \cdots & 1 & \cdots \\ \cdots & \omega_n^{j} & \cdots \\ \cdots & \omega_n^{2j} & \cdots \\ & \vdots & \\ \cdots & \omega_n^{(n-1)j} & \cdots \end{pmatrix}_{ij}$$

Interpolation

$$\left(V_n^{-1}V_n\right)_{ij} = \sum_{k=0}^{n-1} \frac{\omega_n^{-ik}}{n} \omega_n^{jk} = \frac{1}{n}\sum_{k=0}^{n-1} \omega_n^{(-i+j)k}$$

Fall 1: *i* = *j*

$$\frac{1}{n}\sum_{k=0}^{n-1} \omega_n^{(-i+j)k} = \frac{1}{n}\sum_{k=0}^{n-1} \omega_n^{0\cdot k} = 1$$

Fall 2: $i \neq j$, d.h. $-(n-1) \leq -i + j \leq n-1$ und damit $n \nmid -i + j$:

$$\frac{1}{n}\sum_{k=0}^{n-1} \omega_n^{(-i+j)k} = 0$$

Interpolation

Summationslemma

Für alle $n > 0,\ j \geq 0$ mit $n \nmid j$ gilt :

$$\sum_{k=0}^{n-1} \omega_n^{jk} = 0$$

Interpolation

$$a_i = \left(V_n^{-1} y\right)_i$$

$$= \left(\frac{1}{n}, \frac{\omega_n^{-i}}{n}, \ldots, \frac{\omega_n^{-i(n-1)}}{n}\right) \begin{pmatrix} y_0 \\ y_1 \\ \vdots \\ y_{n-1} \end{pmatrix}$$

$$= \sum_{k=0}^{n-1} y_k \frac{\omega_n^{-ik}}{n}$$

$$= \frac{1}{n} \sum_{k=0}^{n-1} y_k \left(\omega_n^{-i}\right)^k$$

Interpolation

$$a = \frac{1}{n}\left(\sum_{k=0}^{n-1} y_k \left(\omega_n^{-0}\right)^k, \sum_{k=0}^{n-1} y_k \left(\omega_n^{-1}\right)^k, \cdots, \sum_{k=0}^{n-1} y_k \left(\omega_n^{-(n-1)}\right)^k\right)$$

$$r(x) = y_0 + y_1 x + y_2 x^2 + \cdots + y_{n-1} x^{n-1}$$

$$a = \frac{1}{n}\left(r\left(\omega_n^{-0}\right), r\left(\omega_n^{-1}\right), \ldots, r\left(\omega_n^{-(n-1)}\right)\right)$$

Interpolation und DFT

$$a = \frac{1}{n}\left(r\left(\omega_n^{-0}\right), r\left(\omega_n^{-1}\right), \ldots, r\left(\omega_n^{-(n-1)}\right)\right)$$

$$a = \frac{1}{n}\left(r\left(\omega_n^{n}\right), r\left(\omega_n^{n-1}\right), \ldots, r\left(\omega_n^{1}\right)\right) \qquad \text{denn } \omega_n^n = 1$$

$$a_i = \frac{1}{n}\left(DFT_n(r)\right)_{n-i} \qquad (i \neq 0)$$

$$a_0 = \frac{1}{n}\left(DFT_n(r)\right)_0$$

Polynomprodukt durch FFT

Berechnung des Produkts zweier Polynome *p, q* vom Grade < *n*

p,q Grad *n-1*, *n* Koeffizienten

↓ **Auswertung durch FFT:** $\omega_{2n}^0, \omega_{2n}^1, \ldots, \omega_{2n}^{2n-1}$

$2n$ Punkt/Wertpaare $\left(\omega_{2n}^i, p\left(\omega_{2n}^i\right)\right)$ und $\left(\omega_{2n}^i, q\left(\omega_{2n}^i\right)\right)$

↓ **Punktweise Multiplikation**

$2n$ Punkt/Wertpaare $\left(\omega_{2n}^i, pq\left(\omega_{2n}^i\right)\right)$

↓ **Interpolation durch FFT**

pq Grad $2n$-2, $2n$-1 Koeffizienten