Bemerkung: $\mathcal{F}$ is nach Lemma 144 und anschließender Bemerkung eine Bijektion.

Lemma 147

Für $\vec{a}, \vec{b} \in \mathbb{C}^n$ gilt

$$\mathcal{F}(\vec{a} * \vec{b}) = \mathcal{F}(\vec{a}) \cdot \mathcal{F}(\vec{b}).$$

[Bem.: Hierbei ist die Dimension der DFT $\geq$ dem Grad von $P_{\vec{c}}$ zu wählen, ω entsprechend!]

Beweis:

Es gilt

$$\begin{aligned}
\mathcal{F}(\vec{a}) \cdot \mathcal{F}(\vec{b}) &= (P_{\vec{a}}(1)P_{\vec{b}}(1), P_{\vec{a}}(\omega)P_{\vec{b}}(\omega), \ldots, P_{\vec{a}}(\omega^{n-1})P_{\vec{b}}(\omega^{n-1})) \\
&= (P_{\vec{c}}(1), P_{\vec{c}}(\omega), \ldots, P_{\vec{c}}(\omega^{n-1})) \\
&= \mathcal{F}(\vec{c}), \quad \text{mit } \vec{c} = \vec{a} * \vec{b}.
\end{aligned}$$

$\square$

Idee: Berechne $\vec{a} * \vec{b}$ vermöge $\mathcal{F}^{-1}(\mathcal{F}(\vec{a}) \cdot \mathcal{F}(\vec{b}))$. Die komponentenweise Multiplikation $F(\vec{a}) \cdot \mathcal{F}(\vec{b})$ benötigt nur $O(n)$ Operationen.

Jedoch: $\mathcal{F}$ ist eine lineare Abbildung $\mathcal{F}(\vec{a}) = \Omega \cdot \vec{a}$, mit $\Omega = (\omega^{kl})_{0 \leq l,k \leq n-1}$. Die Matrixmultiplikation benötigt aber $\Omega(n^2)$ Operationen (also keine offensichtliche Verbesserung im Vergleich zur klassischen Polynom-Multiplikation)!

Ausweg: "Divide and Conquer"!!!

3.5.2 Berechnung der diskreten Fouriertransformation (FFT)

Sei $n = 2^k$ eine 2er-Potenz. Zerlege $\vec{a} = (a_0, \ldots, a_{n-1})$ in einen

$$\begin{array}{rll} \text{geraden Anteil} & \vec{a}_g = (a_0, a_2, \ldots, a_{n-2}) & \text{und einen} \\ \text{ungeraden Anteil} & \vec{a}_u = (a_1, a_3, \ldots, a_{n-1}) & \end{array}$$

Dann gilt:

$$P_{\vec{a}}(x) = P_{\vec{a}_g}(x^2) + xP_{\vec{a}_u}(x^2)\,.$$

Beispiel 148

Sei $\vec{a} = (1, 2, 4, 8)$, also $P_{\vec{a}}(x) = 1 + 2x + 4x^2 + 8x^3$. Damit ist $\vec{a}_g = (1, 4)$ und $\vec{a}_u = (2, 8)$, also

$$\begin{aligned} &P_{\vec{a}_g}(x^2) + xP_{\vec{a}_u}(x^2) \\ &\qquad = 1 \cdot (x^2)^0 + 4 \cdot (x^2)^1 + x \cdot (2 \cdot (x^2)^0 + 8 \cdot (x^2)^1) \\ &\qquad = 1 + 2 \cdot x + 4 \cdot x^2 + 8 \cdot x^3 \end{aligned}$$

Lemma 149

Ist $\mathcal{F}_{\frac{n}{2},\omega^2}(\vec{a}_g) = (c_0, \ldots, c_{\frac{n}{2}-1})$ *und* $\mathcal{F}_{\frac{n}{2},\omega^2}(\vec{a}_u) = (d_0, \ldots, d_{\frac{n}{2}-1})$, *so gilt* $\mathcal{F}_{n,\omega}(\vec{a}) = (e_0, \ldots, e_{n-1})$ *mit*

$$\begin{aligned}
e_i &= P_{\vec{a}}(\omega^i) \\
&= P_{\vec{a}_g}(\omega^{2i}) + \omega^i P_{\vec{a}_u}(\omega^{2i}) \\
&= c_i + \omega^i d_i \\
e_{\frac{n}{2}+i} &= P_{\vec{a}}(\omega^{\frac{n}{2}+i}) \\
&= P_{\vec{a}_g}(\omega^{2(\frac{n}{2}+i)}) + \omega^{\frac{n}{2}+i} P_{\vec{a}_u}(\omega^{2(\frac{n}{2}+i)}) \\
&= c_i + \omega^{\frac{n}{2}+i} d_i
\end{aligned}$$

für $i = 0, \ldots, \frac{n}{2} - 1$.

Bem.: ω^2 ist primitive $\frac{n}{2}$-te Einheitswurzel. Natürlich ist $\omega^{2\frac{n}{2}} = 1$.

Dies liefert folgenden Divide-and-Conquer-Algorithmus:

```
DFT(a⃗,ω)
  Eingabe: a⃗ = (a_0,...,a_{n-1}), n = 2^k, ω
  Ausgabe: F_{n,ω}(a⃗) = (e_0,...,e_{n-1})

  if n = 1 then e_0 := a_0
  else
     a⃗_g := (a_0, a_2, ..., a_{n-2})
     a⃗_u := (a_1, a_3, ..., a_{n-1})
     (c_0, ..., c_{n/2-1}) := DFT(a⃗_g, ω^2)
     (d_0, ..., d_{n/2-1}) := DFT(a⃗_u, ω^2)
     for i = 0 to n/2 - 1 do
        e_i := c_i + ω^i d_i
        e_{n/2+i} := c_i + ω^{n/2+i} d_i
     endfor
  endif
  return(e_0, ..., e_{n-1})
```

Satz 150

Der Algorithmus `DFT` *berechnet* $\mathcal{F}_{n,\omega}(\vec{a})$ *auf Eingabe* $n = 2^k$, $\vec{a}$, ω *in* $T(n) = O(n \log n)$ *Operationen.*

Beweis:

Aus dem Algorithmus erhält man folgende Rekursion

$$T(n) = 2T(n/2) + cn$$

mit einer Konstante $c > 0$ und $T(1) = 1$. Mit $n = 2^k$ folgt

$$\begin{aligned} T(2^k) &= 2T(2^{k-1}) + cn = 2(2T(2^{k-2}) + cn/2) + cn \\ &= \ldots = 2^\ell T(2^{k-\ell}) + \ell cn \end{aligned}$$

Speziell für $\ell = k$ gilt $T(2^k) = kc2^k + 2^k T(1)$, und wir erhalten $T(2^k) = O(2^k k) = O(n \log n)$. $\square$

3.5.3 Berechnung der inversen diskreten Fouriertransformation

Satz 151

Es gilt

$$\mathcal{F}_{n,\omega}^{-1} = \frac{1}{n}\mathcal{F}_{n,\omega^{-1}}.$$

Bemerkung: ω^{-1} ist ebenso eine primitive n-te Einheitswurzel.
Zum Beweis von Satz 151 benötigen wir folgendes Lemma:

Lemma 152

Ist ω eine primitive n-te Einheitswurzel, so gilt

$$\sum_{j=0}^{n-1} \omega^{kj} = 0$$

für alle $k = 1, \ldots, n-1$.

Beweis:

Für jedes $a \in \mathbb{C}$, $a \neq 1$, gilt $\sum_{j=0}^{n-1} a^j = \frac{a^n-1}{a-1}$. Speziell für $a = \omega^k$ ist $a^n = \omega^{kn} = 1$, $(k = 1, \ldots, n-1)$. □

Nun zum Beweis von Satz 151.

Beweis:

Sei $\vec{e} = \mathcal{F}_{n,\omega}(\vec{a}) = (e_0, \ldots, e_{n-1})$. Wir zeigen, dass gilt:

$$\frac{1}{n}\mathcal{F}_{n,\omega^{-1}}(\vec{e}) = \vec{a}$$

$$\begin{aligned} P_{\vec{e}}(\omega^{-k}) &= \sum_{j=0}^{n-1} e_j \omega^{-kj} = \sum_{j=0}^{n-1} P_{\vec{a}}(\omega^j)\omega^{-kj} \\ &= \sum_{j=0}^{n-1}\sum_{i=0}^{n-1} a_i \omega^{ij}\omega^{-kj} = \sum_{i=0}^{n-1} a_i \sum_{j=0}^{n-1} \omega^{(i-k)j} = n a_k, \end{aligned}$$

denn nach Lemma 152 ist $\sum_{j=0}^{n-1} \omega^{(i-k)j} = 0$, falls $i \neq k$.

Im Fall $i = k$ gilt $\sum_{j=0}^{n-1} \omega^{(i-k)j} = n$. □