Over het vermenigvuldigingsalgoritme van Schönhage en Strassen

Opmerkingen door T. H. Koornwinder bij §8.3 in von zur Gathen & Gerhard, Modern computer algebra

College Computeralgebra, UvA, januari–maart 2003

1. Inleiding

Laat $k \in \mathbb{Z}_{>0}$, $n := 2^k$ en $\omega := e^{2\pi i/n}$. Als we een polynoom $p(x)$ in x met coëfficiënten in $\mathbb{Q}$ sturen naar hetzelfde polynoom $p(\omega)$ in ω:

$$p(x) = q_0 + q_1 x + \cdots + q_m x^m \mapsto p(\omega) = q_0 + q_1\omega + \cdots + q_m\omega^m,$$

dan geeft dit 0 desda $p(x)$ deelbaar is door $x^{\frac{1}{2}n} + 1$ (dit is het cyclotomische polynoom $\Phi_n(x)$ omdat n een macht van 2 is), m.a.w.,

$$p(x) \bmod (x^{\frac{1}{2}n} + 1) \mapsto p(\omega) : \ \mathbb{Q}[x]/\langle x^{\frac{1}{2}n} + 1\rangle \to \mathbb{Q}[\omega] \tag{1.1}$$

is een bijectieve afbeelding (in feite een lichaamsisomorfisme).

We hebben eerder de discrete Fouriertransformatie (DFT) besproken. Deze kan beschouwd worden als een afbeelding van een polynoom $f(y) \in \mathbb{Q}[y]/\langle y^n - 1\rangle$ naar een rijtje $\widehat{f}$ van n polynomen in ω, nl.

$$\widehat{f} = \{f(1), f(\omega), \ldots, f(\omega^{n-1})\}. \tag{1.2}$$

Gezien het isomorfisme (1.1) kunnen we dus evengoed schrijven:

$$\widehat{f} = \{f(1), f(x), \ldots, f(x^{n-1})\} \bmod (x^{\frac{1}{2}n} + 1). \tag{1.3}$$

We hadden als belangrijke toepassing van de DFT gezien dat dure vermenigvuldiging van polynomen ($O(n^2)$) door de DFT wordt teruggebracht tot goedkope vermenigvuldiging van getallen. Immers, de DFT van $f(y)\,g(y) \bmod (y^n - 1)$, geschreven volgens (1.2), is

$$(fg)\widehat{\ } = \{f(1)g(1), f(\omega)g(\omega), \ldots, f(\omega^{n-1})g(\omega^{n-1})\}.$$

Deze n vermenigvuldigingen zouden betrekkelijk goedkoop kunnen worden uitgevoerd als we de machten van ω benaderden met complexe getallen tot op een zeker aantal decimalen voor de reële en imaginaire delen. Echter, dit staan we onszelf niet toe omdat we exact willen rekenen. Equivalent, met de DFT geschreven volgens (1.3), hebben we:

$$(fg)\widehat{\ } = \{f(1)g(1), f(x)g(x), \ldots, f(x^{n-1})g(x^{n-1})\} \bmod (x^{\frac{1}{2}n} + 1). \tag{1.4}$$

We zijn dus nauwelijks opgeschoten, integendeel, want nu moeten we n keer twee polynomen van graad $< \frac{1}{2}n$ met elkaar vermenigvuldigen.

Laat $R := \mathbb{Q}[\omega] \simeq \mathbb{Q}[x]/\langle x^{\frac{1}{2}n} + 1\rangle$. De situatie wordt gunstiger als we de DFT willen gebruiken om het product $f(y)g(y)$ van twee polynomen van graad $< n$ met coëfficiënten in R uit te rekenen. Dus laat $f(y), g(y) \in R[y]/\langle y^n - 1\rangle$ en bereken $f(y)g(y) \in R[y]/\langle y^n - 1\rangle$. Op de rechtstreekse manier kost dit $O(n^2)$ vermenigvuldigingen in R dus $O(n^4)$ vermenigvuldigingen in $\mathbb{Q}$. Maar het resultaat van de DFT van $f(y)g(y)$ in de vorm (1.4) vereist $O(n)$ vermenigvuldigingen in R, dus slechts $O(n^3)$ vermenigvuldigingen in $\mathbb{Q}$.

Dit brengt ons op het idee om het probleem van vermenigvuldiging van twee polynomen van graad $< n$ in x terug te brengen tot vermenigvuldiging van twee polynomen in x en y, elk van graad $< \sqrt{n}$ in x en van graad $< \sqrt{n}$ in y. Verder hebben we gezien dat we na het enmen van de DFT polynomen in x met elkaar moeten vermenigvuldigen modulo $x^{\frac{1}{2}n} + 1$. Om een recurrent algoritme te krijgen, is het dan gunstig om te beginnen met vermenigvuldiging modulo $x^n + 1$ i.p.v. modulo $x^n - 1$. Voor twee polynomen in x waarvan het product graad $< n$ heeft is het natuurlijk geen bezwaar dat we modulo $x^n + 1$ i.p.v. $x^n - 1$ gaan rekenen.

2. Beschrijving van het algoritme

Laat weer $n := 2^k$. Laat $m := 2^{\frac{1}{2}k}$, $t := 2^{\frac{1}{2}k}$ als k even, en $m := 2^{\frac{1}{2}k-\frac{1}{2}}$, $t := 2^{\frac{1}{2}k+\frac{1}{2}}$ als k oneven is. Laat $f(x) \in \mathbb{Q}[x]/\langle x^n + 1\rangle$. We kunnen $f(x)$ schrijvan in de vorm

$$f(x) = \sum_{j\geq 0} f_j(x)\, x^{jm} \bmod (x^n + 1),$$

waarbij $f_j(x)$ voor elke j een polynoom van graad $< m$ in x is. Dus we kunnen schrijven

$$f(x) = f'(x, x^m) \quad \text{met} \quad f'(x, y) = \sum_{j\geq 0} f_j(x)\, y^j \bmod (y^t + 1).$$

Schrijf evenzo, voor $g(x) \in \mathbb{Q}[x]/\langle x^n + 1\rangle$,

$$g(x) = g'(x, x^m) \quad \text{met} \quad g'(x, y) = \sum_{j\geq 0} g_j(x)\, y^j \bmod (y^t + 1),$$

waarbij elk polynoom $g_j(x)$ graad $< m$ heeft. Dan kunnen we schrijven

$$h'(x, x^m) := f'(x, x^m)\, g'(x, x^m) = f(x)\, g(x)$$

met

$$f'(x, y)\, g'(x, y) = h'(x, y) = \sum_{j\geq 0} h_j(x)\, y^j \bmod (y^t + 1),$$

waarbij elk polynoom $h_j(x)$ graad $< 2m$ heeft.

Omdat alle graden in x kleiner dan $2m$ zijn kunnen we $f'(x,y)$, $g'(x,y)$ en $h'(x,y)$ injectief afbeelden naar resp.

$$f^*(y) := \sum_{j\geq 0} \big(f_j(x) \bmod (x^{2m}+1)\big)\, y^j \bmod (y^t+1),$$
$$g^*(y) := \sum_{j\geq 0} \big(g_j(x) \bmod (x^{2m}+1)\big)\, y^j \bmod (y^t+1),$$
$$h^*(y) := \sum_{j\geq 0} \big(h_j(x) \bmod (x^{2m}+1)\big)\, y^j \bmod (y^t+1).$$

Schrijf $R := \mathbb{Q}[x]/\langle x^{2m}+1\rangle$. Dan liggen $f^*(y)$, $g^*(y)$ en $h^*(y)$ in $R[y]/\langle y^t+1\rangle$ en

$$h^*(y) = f^*(y)\, g^*(y) \tag{2.1}$$

in $R[y]/\langle y^t+1\rangle$.

We zijn nu bijna in een goede situatie om het product in (2.1) via de DFT om te zetten in een goedkoper uit te rekenen product. Alleen zouden we nog modulo y^t-1 moeten kunnen werken i.p.v. modulo y^t+1. Dit laatste kan met de volgende truc. Zij eerst k even. Dan $m=t$, dus $x^{2t} = -1 \bmod (x^{2m}+1)$, dus $(x^2)^t = -1$ in R en $(x^4)^t = 1$ in R, dus met $\omega := x^4 \bmod (x^{2m}+1) \in R$ hebben we $\omega^t = 1$ en $\omega^k \neq 1$ als $k = 1,2,\ldots,t-1$. Dus $f^*(x^2y) \in R[y]/\langle y^t-1\rangle$ omdat $f^*(y) \in R[y]/\langle y^t+1\rangle$. We kunnen dan de DFT van $f^*(x^2y)$ t.o.v. ω nemen als een rij van $t=m$ elementen in R, verkregen door in $f^*(x^2y)$ achtereenvolgens $y=1$, $y = x^4 \bmod (x^2+1)$, $y = x^8 \bmod (x^2+1)$, $\ldots$ te substitueren. Dus

$$(f^*(x^2y))\hat{} = \{f^*(x^2), f^*(x^6), \ldots, f^*(x^{4m-2})\}, \tag{2.2}$$

waarbij de elementen

$$f^*(x^{4l-2}) = \sum_{j\geq 0} f_j(x) x^{(4l-2)j} \bmod (x^{2m}+1) \qquad (l = 1,2,\ldots,m)$$

alle in R liggen. Analoog aan (2.2) kunnen we schrijven

$$(g^*(x^2y))\hat{} = \{g^*(x^2), g^*(x^6), \ldots, g^*(x^{4m-2})\},$$

en er geldt

$$(h^*(x^2y))\hat{} = \{f^*(x^2)g^*(x^2), f^*(x^6)g^*(x^6), \ldots, f^*(x^{4m-2})g^*(x^{4m-2})\}.$$

Laat nu k oneven. Dan $t = 2m$, dus $x^t = -1 \bmod (x^{2m}+1)$, dus met $\omega := x^t \bmod (x^{2m}+1) \in R$ hebben we $\omega^t = 1$ en $\omega^k \neq 1$ als $k = 1,2,\ldots,t-1$. Dan liggen $f^*(xy)$, $g^*(xy)$ en $h^*(xy)$ in $R[y] \bmod (y^t-1)$. Neem de DFT t.o.v. ω, dan

$$(f^*(xy))\hat{} = \{f^*(x), f^*(x^3), \ldots, f^*(x^{4m-2})\},$$
$$(g^*(xy))\hat{} = \{g^*(x), g^*(x^3), \ldots, g^*(x^{4m-2})\},$$
$$(h^*(xy))\hat{} = \{f^*(x)g^*(x), f^*(x^3)g^*(x^3), \ldots, f^*(x^{4m-2})g^*(x^{4m-2})\},$$

waarbij alle elementen in R liggen.

Via de FFT hebben we dus 1 vermenigvuldiging in $\mathbb{Q}[x]/\langle x^{tm+1}\rangle$ gereduceerd tot t vermenigvuldigingen in $\mathbb{Q}[x]/\langle x^{2m+1}\rangle$. Dit kunnen we itereren. De complexiteit (zie boek) is $O(n \log n \log(\log n))$.