Prof. Friedrich Eisenbrand
Alfonso Cevallos and Yuri Faenza

Discussions from: April 23, 2014

Computer Algebra

Spring 2014

Assignment Sheet 5

Exercises marked with a ⋆ can be handed in for bonus points. Due date is May 06.

Exercise 1

Let $R = \mathbb{Z}_7$ and $S = \mathbb{Z}_{11}$ and consider the product ring $T = R \times S \cong \mathbb{Z}_{77}$. Consider the following example about how roots of unity in the product ring *don't* relate to roots of unity in the component rings (compare also the next exercise!).

1. Show that $\omega_1 = 2$ is a primitive 3-rd root of unity modulo 7.
2. Show that $\omega_2 = 4$ is a primitive 5-th root of unity modulo 11.
3. Let $\omega = 37$. Prove that $\omega \equiv \omega_1 \pmod 7$ and $\omega \equiv \omega_2 \pmod{11}$ and that ω is a 15-th root of unity modulo 77 (that is, $\omega^{15} \equiv 1 \pmod{77}$, and $\omega^k \not\equiv 1 \pmod{77}$ for $1 \le k < 15$).
4. Prove that ω is *not* a primitive root of unity modulo 77.

Exercise 2

Let R and S be commutative rings and consider their product ring $T = R \times S$. Let $\omega = (\omega_R, \omega_S) \in T$. Prove that ω is a primitive n-th root of unity if and only if ω_R and ω_S are primitive n-th roots of unity in R and S, respectively.

Exercise 3

Let $R = \mathbb{Z}_{21}$. For every element $x \in R$, determine (without using a computer!) whether it is in R^* (that is, whether it is invertible) and whether it is a zero divisor. Determine the order of every element $x \in R^*$. Finally, determine which elements are primitive roots of unity.

Exercise 4 (⋆)

Develop an algorithm that, given an odd-degree polynomial $f \in \mathbb{Z}[x]$ and $\varepsilon > 0$, computes an interval of length at most ε enclosing a root of f using binary search. This algorithm has to run in polynomial time in the encoding length of f and ε. Prove the correctness of your algorithm.

Exercise 5

Let $n \in \mathbb{N}$. Show that 2 is a primitive $2n$-th root of unity modulo $2^n + 1$ if and only if n is a power of 2.

Exercise 6
Let $f = x^2 + 2x - 5$ and $g = x^2 + 3x + 2$. Let $N = 17$ and $\omega = 2 \in \mathbb{Z}_N$.

1. Show that ω is an 8-th primitive root of unity in $\mathbb{Z}_N$.
2. Use the discrete Fourier transform to compute $f(\omega^i)$ and $g(\omega^i) \mod N$, $i = 0 \ldots 7$.
3. Use the inverse discrete Fourier transform on $f(\omega^i)g(\omega^i)$. Can you use the result to find $fg \in \mathbb{Z}[x]$?

Exercise 7
Let $a \in \mathbb{Z}_M$, where $M = 2^L + 1$ and let j, $1 \le j \le L$ be a natural number. Show that the product $a \cdot 2^j$ can be computed with $O(L)$ bit-operations. *Hint: This is not just shifting to the left but a little bit more*

Exercise 8
You are to multiply two n-degree polynomials $f(x)$ and $g(x)$ in $\mathbb{Z}[x]$. For this you want to use the modular DFT approach. Thus you want to translate the problem into a suitable problem of polynomial multiplication in $\mathbb{Z}_M[x]$ using the following scheme. The polynomials f and g are mapped into $\mathbb{Z}_M[x]$ via the canonical homomorphism. In there they are multiplied using the modular FFT. From this product, the original product $f \cdot g \in \mathbb{Z}[x]$ is to be reconstructed.

1. Let a be an upper bound on the absolute values of the coefficients of f and g. Determine an $M \in \mathbb{N}_+$ such that the reconstruction of the product $f \cdot g \in \mathbb{Z}_M[x]$ is unique. Derive a lower bound on M. (These bounds should not be far apart!)
2. Derive an upper bound on the bit-complexity of this modular approach in terms of n and $size(a)$.

Exercise 9 ($\star$)

- Implement the algorithm seen in class that, given an $n-$th root of unity ω (n is a power of 2) and the coefficients $a = (a_0, \ldots, a_{n-1})$, computes $DTF_\omega(a)$.
- Using the existence of $n-$th root of unity for appropriate n and the fact that $DFT_\omega^{-1} = n^{-1} DFT_{\omega^{-1}}$ (recall the arguments seen in class), implement an algorithm that computes the product of two polynomials in $\mathbb{Z}[x]$. Test it on the polynomials from Exercise 6.