CS-E4500 Problem Set 8

Jaan Tollander de Balsch - 452056

March 21, 2019

Material used in this report: Gathen and Gerhard (2013), Sections 12.1-3, 14.6.

Problem 1

Factor the polynomial $f = 1 + x + x^2 + 2x^3 + x^4 \in \mathbb{Z}_3[x]$.

First, lets evaluate the polynomial f in all points in $\mathbb{Z}_3$

$$f(0) = 1, \quad f(1) = 0, \quad f(2) = 0.$$

As can be seen, the polynomial has two roots, 1 and 2. Therefore $(x - 1)$ and $(x - 2)$ are its factors. Now, diving the polynomial f by the product of these factors will give us the polynomial

$$\frac{f}{(x-1)(x-2)} = \frac{f}{(x+2)(x+1)} = x^2 + 2x + 2$$

which is irreducible (from last week's problem 1). Therefore all of the factors of f are $(x - 1)$, $(x - 2)$ and $(x^2 + 2x + 2)$.

Problem 2

Square-and-multiply modular exponentiation. Let q be a prime power. Present an algorithm that, given $m \in \mathbb{Z}_{\geq 1}, f \in \mathbb{F}_q[x]$, and $g \in \mathbb{F}_q[x] \setminus \{0\}$ with $\deg f, \deg g \leq d \in \mathbb{Z}_{\geq 1}$ as input, computes $f^m \operatorname{rem} g$ in $O(M(d) \log m)$ operations in $\mathbb{F}_q$. Carefully justify the number of operations used by your algorithm.

Decompose the polynomial $f^m \operatorname{rem} g$ into sum polynomials $f^{2^i} \operatorname{rem} g$

$$f^m \operatorname{rem} g = f^{2^{n_1}} \operatorname{rem} g + f^{2^{n_2}} \operatorname{rem} g + ... + f^{2^{n_k}} \operatorname{rem} g.$$

This is done by decomposing m into a sum of powers of two

$$m = 2^{n_1} + 2^{n_2} + ... + 2^{n_k}$$

such that $\log m \geq n_1 > n_2 > ... > n_k \geq 0$ and $k \in \mathbb{N}$. The decomposition can be computed with $\log m$ successive divisions by 2.

The powers of two of the polynomial f can be computed recursively upto power 2^n

$$\begin{aligned} f^1 &= f \\ f^2 &= (f^1 \cdot f^1) \operatorname{rem} g \\ f^4 &= (f^2 \cdot f^2) \operatorname{rem} g \\ &\vdots \\ f^{2^n} &= (f^{2^{n-1}} \cdot f^{2^{n-1}}) \operatorname{rem} g, \quad n \in \mathbb{N}. \end{aligned}$$

The full algorithm for square-and-multiply modular exponentiation using the ideas above gives:

- **Input**: Let q be a prime power, the inputs are $m \in \mathbb{Z}_{\geq 1}$, $f \in \mathbb{F}_q[x]$, and $g \in \mathbb{F}_q[x] \setminus \{0\}$ with $\deg f, \deg g \leq d \in \mathbb{Z}_{\geq 1}$.
- **Output**: The polynomial $h = f^m \operatorname{rem} g \in \mathbb{F}_q[x]$.

Square-and-Multiply-Modular-Exponentiation(f, m, g)

1) Compute $N = \{n_1, n_2, ..., n_k\}$ from the decomposition of m
2) $h = 0$
3) $\tilde{f} = f^1$
4) **for** $n = 0, 1, ..., n_k$
5) ..... **if** $n \in N$
6) .... $h = h + \tilde{f}$
7) ..... $\tilde{f} = (\tilde{f} \cdot \tilde{f}) \operatorname{rem} g$
8) **return** h

Analysis: There are $\log m + 1$ iterations in the for-loop. Inside the for-loop, there are the following operations

- Maximum of one polynomial addition. *Polynomial addition (Horner's rule)* $O(d)$.
- One polynomial multiplication. *Fast polynomial multiplication* $O(M(d))$.
- One polynomial remainder. *Fast polynomial remaindering (Euclidean algorithm)* $O(M(d) \log d)$.

The total number of operations in $\mathbb{F}_q[x]$ is

$$O((M(d) \log d) \log m).$$

NOTE: I'm not sure how to get rid of the $\log d$ term arising from the remainder.

Problem 3

Formal derivative of a factorization. Let q be a prime power. Let $f \in \mathbb{F}_q[x]$ be monic with factorization $f = f_1^{d_1} f_2^{d_2} \cdots f_r^{d_r}$ into distinct irreducible polynomials $f_1, f_2, \ldots, f_r \in \mathbb{F}_q[x]$ and $d_1, d_2, \ldots, d_r \in \mathbb{Z}_{\geq 1}$. Show that the formal derivative of f satisfies

$$f' = d_1 f_1' \frac{f}{f_1} + d_2 f_2' \frac{f}{f_2} + \ldots + d_r f_r' \frac{f}{f_r} \in \mathbb{F}_q[x].$$

Above we write d_j for a sum of d_j copies of the multiplicative identity of $\mathbb{F}_q$.

Let f and g be polynomials in $\mathbb{F}_q[x]$. The formal derivative satisfies two properties:

1) The product rule
$$(fg)' = f'g + fg'$$

2) The chain rule
$$(f(g))' = f'(g)g'$$

The product rule where $f = f_1 f_2 \cdots f_r$ can be generalized into

$$\left(\prod_{i=1}^{r} f_i \right)' = \sum_{i=1}^{r} f_i' \frac{f}{f_i}.$$

Also, the chain rule gives the derivative

$$(f^d)' = d f^{d-1} f', \quad d \in \mathbb{Z}_{\geq 1}.$$

Therefore using the rules above, the formal derivative satisfies

$$\begin{aligned}
f' &= (f_1^{d_1} f_2^{d_2} \cdots f_r^{d_r})' \\
&= \left(\prod_{i=1}^{r} f_i^{d_i} \right)' \\
&= \sum_{i=1}^{r} \left(f_i^{d_r} \right)' \frac{f}{f_i^{d_r}} \\
&= \sum_{i=1}^{r} d_r f_i^{d_r - 1} f_i' \frac{f}{f_i^{d_r}} \\
&= \sum_{i=1}^{r} d_r f_i' \frac{f}{f_i} \\
&= d_1 f_1' \frac{f}{f_1} + d_2 f_2' \frac{f}{f_2} + \ldots + d_r f_r' \frac{f}{f_r}.
\end{aligned}$$

Problem 4

References

Gathen, Joachim von zur, and Jurgen Gerhard. 2013. *Modern Computer Algebra.* 3rd ed. New York, NY, USA: Cambridge University Press.