Einführung in die Computeralgebra

Übungsblatt 6

Sommersemester 2016

Michael Stoll

20. Mai 2016

Abgabe:
Freitag, 27. Mai, in der Vorlesung (bis 11:00 Uhr online für Programmieraufgaben).

Übungsaufgaben bitte **handschriftlich** bearbeiten (außer Programmieraufgaben);
nur ein Name pro Blatt! —
Schnellhefter und **Deckblatt** nicht vergessen!

(1) Sei R ein (kommutativer) Ring und sei $\omega \in R$ eine primitive n-te Einheitswurzel. Zeigen Sie:

(a) Sei $\ell \in \mathbb{Z}$. ω^ℓ ist genau dann eine primitive n-te Einheitswurzel in R, wenn ℓ und n teilerfremd sind.

(b) Sei m ein Teiler von n. Dann ist $\omega^{n/m}$ eine primitive m-te Einheitswurzel in R.

(15+10)

(2) Sei R ein (kommutativer) Ring.

(a) Sei $n \geq 2$ in R invertierbar. Zeigen Sie, dass die Restklasse von y im Faktorring $R' = R[y]/\langle y^n - 1\rangle$ *keine* primitive $n$te Einheitswurzel in R' ist.

(b) Sei 2 in R invertierbar und sei $k \geq 1$. Zeigen Sie, dass die Restklasse von y in $R' = R[y]/\langle y^{2^{k-1}} + 1\rangle$ eine primitive 2^k-te Einheitswurzel in R' ist. (15+10)

(3) Betrachten Sie folgende Darstellung von Polynomen in einer Variablen über $\mathbb{F}_2$: Das Polynom $f = \sum_j a_j X^j \in \mathbb{F}_2[X]$ wird dargestellt durch die aufsteigende Folge der Exponenten j mit $a_j = 1$ (d.h., $a_j \neq 0$). Sei $\ell(f)$ die Länge von f in dieser Darstellung, also die Anzahl der von null verschiedenen Terme in f. Zeigen Sie, dass die Komplexität der Multiplikation zweier Polynome $f, g \in \mathbb{F}_2[X]$ mindestens von der Ordnung $\ell(f)\ell(g)$ ist. (25)

(4) **Programmieraufgabe:**
Bestimmen Sie die Zeit, die Magma für die Multiplikation zweier Zahlen der Länge 2^n braucht, für $n \leq 20$. Tragen Sie den Logarithmus der Zeit gegen n in einem Graphen auf und interpretieren Sie das Resultat.

Hinweis: Um sinnvolle Ergebnisse zu bekommen, sollten Sie die Multiplikation mehrmals durchführen und die Gesamtdauer bestimmen. Wenn Sie den `Cputime()`-Befehl von Magma benutzen, dann sollten Sie dem Ergebnis mittels `ChangePrecision` mehr Stellen Genauigkeit geben, bevor Sie dividieren. Eine zufällige Zahl der Länge n erhält man mit `Random(2^(n*64-1), 2^(n*64)-1)`. (25)