SS 2008 Prof. Dr. K. Madlener

Übungen zur Vorlesung **Computer Algebra**

Blatt 1 Abgabe: 2008/04/14

1. Aufgabe: *Ringe und Körper* Wir wollen einige einfache Sätze zeigen:

a) Sei R ein Integritätsbereich, ferner seien $a, b \in R^*$ (die Menge der von 0 verschiedenen Elemente von R). Es sind a und b genau dann assoziiert, wenn es ein $u \in E(R)$ (die Einheitengruppe von R) mit $a = ub$ gibt.

b) Sei R ein euklidischer Ring und sei I ein Ideal von R. Ist $0 \neq a \in I$, so ist genau dann $I = aR$, wenn $v(a) \leq v(b)$ für alle $b \in I \setminus \{0\}$ ist (wobei v die Bewertungsfunktion von R sei).

Zeigen Sie, dass weiter folgt, dass R ein Hauptidealring ist (d. h., dass alle Ideale von R Hauptideale sind).

c) Ist K ein Körper, so ist $K[x]$ ein euklidischer Ring.

d) Ist K ein Körper und L eine Erweiterung von K (also ein Körper, der K als Teilkörper enthält), ist weiter $f \in K[x]$ und $\ell \in L$ eine Nullstelle von f, so ist f in $L[x]$ durch $x - \ell$ teilbar.

2. Aufgabe: *Polynomdivision* Wir untersuchen die Laufzeit der klassischen Polynomdivision mit Rest. Gegeben sei folgender Algorithmus:

function PolyQuoRem(a, b)
$a = \sum_{0 \leq i \leq n} a_i x^i$, $b = \sum_{0 \leq i \leq m} b_i x^i \in R[x]$, R ist ein kommutativer Ring mit 1, alle $a_i, b_i \in R$, b_m ist eine Einheit in R und $n \geq m \geq 0$.
Ausgabe: $q, r \in R[x]$ mit $a = qb + r$ und $\deg r < m$ oder $r = 0$.
$r \leftarrow a$
for $i \leftarrow n - m, n - m - 1, \ldots 0$ **do**
if $\deg(r) = m + i$ **then**
$q_i \leftarrow \mathrm{lc}(r)/b_m; r \leftarrow r - q_i x^i b$
else $q_i \leftarrow 0$
end if
end for
return $q = \sum_{0 \leq i \leq n-m} q_i x^i$ und r
end function

Nehmen Sie an, dass ein Polynom $p = \sum_{0 \leq i \leq k} p_i x^i$ vom Grad k durch eine dichte Darstellung gegeben sei, d. h. im Wesentlichen durch einen Koeffizientenvektor $\vec{p} = (p_0, \ldots, p_k)$. Geben Sie die Laufzeit, gemessen in der Anzahl von Ringoperationen in R, im schlechtesten Fall in Abhängigkeit von n und m an. Sind q und r im Allgemeinen eindeutig? Beweis!

3. Aufgabe: *Diophantische Gleichungen* Gibt es $s, t \in \mathbb{Z}$, so dass $24s + 14t = 1$ bzw. so dass $61s + 37t = 56$? Geben Sie jeweils alle möglichen Lösungen an.

Zeigen Sie allgemeiner: Die lineare diophantische Gleichung $ax + by = c$ mit $a, b, c \in \mathbb{Z}$ ist genau dann (in $\mathbb{Z}$) lösbar, wenn für $d = \mathrm{GGT}(a, b)$ gilt: $d|c$. Ist in diesem Fall (x_0, y_0) eine spezielle Lösung, dann ist

$$\{(x_0 + k \cdot \frac{b}{d}, y_0 - k \cdot \frac{a}{d} \mid k \in \mathbb{Z}\}$$

die Menge alle Lösungen. (Was bedeutet eigentlich die Schreibweise $\frac{a}{d}$ in diesem Zusammenhang?)

4. Aufgabe: *GGT*

a) Berechnen Sie größte gemeinsame Teiler von $f = x^5 + x^4 + x^3 - x^2 - x + 1$ und $g = x^3 + x^2 + x + 1$ ($f, g \in \mathbb{Z}_p[x]$) für $p = 3$ sowie $p = 5$. Berechnen Sie jeweils auch Polynome s und t mit $\mathrm{ggT}(f, g) = sf + tg$.

b) Wir betrachten den folgenden ggT-Algorithmus nach J. Stein (war möglicherweise aber schon im antiken China bekannt):

```
1: function BinaryGCD(u, v ∈ ℕ⁺)                    ▷ Returns the g.c.d of u and v
2:     g ← 1
3:     while (u mod 2 = 0) ∧ (v mod 2 = 0) do
4:         u ← u/2; v ← v/2; g ← 2g
5:     end while
6:     while (u ≠ 0) do
7:         if (u mod 2 = 0) then
8:             u ← u/2
9:         else if (v mod 2 = 0) then
10:            v ← v/2
11:        else
12:            t ← |u − v|/2
13:            if u ≥ v then
14:                u ← t
15:            else
16:                v ← t
17:            end if
18:        end if
19:    end while
20:    return g · v
21: end function
```

Zeigen Sie, dass dieser Algorithmus tatsächlich für Eingaben $u, v \in \mathbb{N}^+$ den Wert $\mathrm{ggT}(u, v)$ berechnet, und zwar mit $O((\lambda(uv))^2)$ Bitoperationen im schlechtesten Fall. Dafür nehmen wir an, dass Zahlen aus $\mathbb{N}^+$ in Binärdarstellung gegeben sind und $\lambda(x)$ die Länge der Binärdarstellung (ohne führende Nullen) von x bezeichnet. Zeigen Sie abschließend, dass im schlechtesten Fall $O(\lambda(u)\lambda(v))$ Bitoperationen *nicht* ausreichen.